CRIC: Kontextbasierte Empfehlung unstrukturierter Texte in Echtzeitumgebungen

Ein Verfahren zur Bestimmung der semantischen Proximität von Textobjekten auf Basis eines heuristischen asymmetrischen Distanzmaßes

Vom Fachbereich Elektrotechnik und Informatik der Universität Siegen

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften

(Dr. rer. nat.)

genehmigte Dissertation

von

Dipl. Inform. André Klahold

1. Gutachter: Prof. Dr. Wolfgang Merzenich
2. Gutachter: Prof. Dr. Madjid Fathi

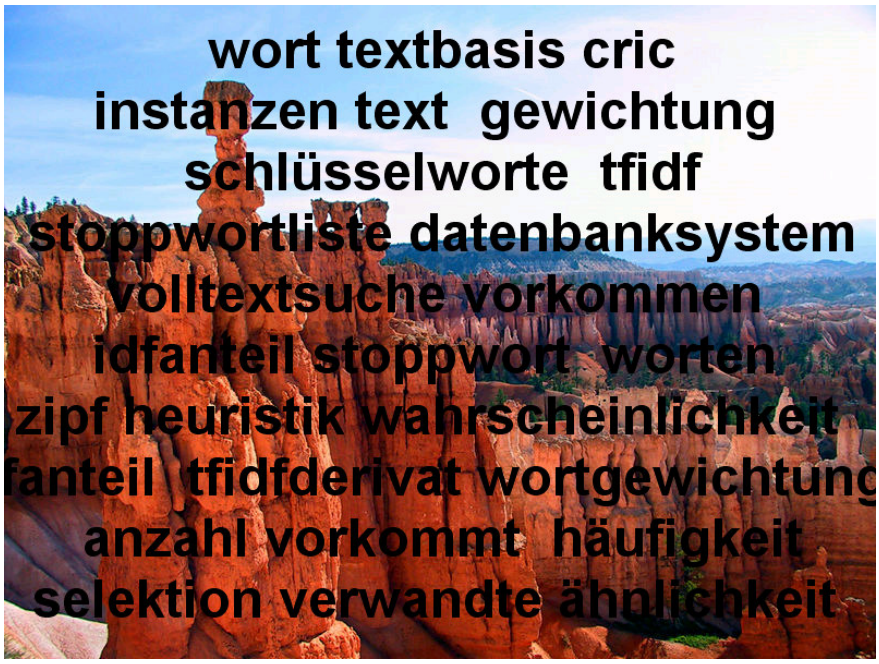
Vorsitzender: Prof. Dr. Hans Wojtkowiak

Tag der mündlichen Prüfung: 2006-08-16

urn:nbn:de:hbz:467-2410

Vorwort

Der Gegenstand dieser Arbeit ist ein neues Empfehlungsverfahren mit dem auf automatisiertem Wege Querverweise zwischen Texten erzeugt werden können. Die Grundidee des Verfahrens entstand im Herbst des Jahres 2002 auf einer Reise durch den Westen der USA. Beim Blick auf die unzähligen so genannten Hoodoos (durch Erosion entstandene Gesteinssäulen aus Kalkstein) des Amphitheaters des Bryce Canyon kam die Assoziation mit den Worten eines Textes zustande. Wie sich in einem Text bestimmte Worte aus der scheinbar homogenen Menge konkatenierter Buchstabenfolgen abheben, so fallen im Bryce Canyon auch einzelne Hoodoos besonders auf. Ein Beispiel ist der als "Thors Hammer" bezeichnete Hoodoo im folgenden Bild, das mit den durch CRIC ermittelten "wichtigsten" Worten (27 aus über 70.000 Worten) dieser Arbeit überlagert ist.



Damit drängte sich die Frage auf, wie man die "prägenden" Worte eines Textes bestimmen könnte. Sollte das gelingen, könnte man inhaltlich verwandte Texte über einen Vergleich der prägenden Worte ermitteln.

Der erste Prototyp des Verfahrens hat nur die Grundidee mit der im Rahmen dieser Arbeit vorgestellten Endfassung gemein. Insbesondere das Laufzeitverhalten war in frühen Versionen nur in unzureichender Weise für größere Textmengen oder eine hohe Nutzungsrate geeignet. Um eine solide Basis für eine Optimierung des Laufzeitverhaltens zu erhalten, wurde daher zunächst ein Verfahren zur Abschätzung der Laufzeit von SQL-Befehlen entwickelt.

Nach rund einem Jahr war dann die erste Fassung des Verfahrens einsatzbereit und konnte im ersten Quartal 2004 in der Praxis evaluiert werden. Auf der KnowTech 2004 wurde das Verfahren, sowie die Ergebnisse der Evaluation, erstmals vorgestellt.

Während der Erstellung der Arbeit waren die intensiven Diskussionen mit Professor Wolfgang Merzenich eine wertvolle Hilfe.

Zwei Menschen haben maßgeblich zur Entstehung der Arbeit beigetragen. Die Diskussionen mit meinem Freund, Mentor und Vorbild Prof. Armin John haben mein Interesse an akademischen Themen stets wach gehalten. Und ohne das Verständnis und die Unterstützung meiner Frau Daniela wären die zahllosen "geopferten" Wochenenden und durchgearbeiteten Nächte nicht möglich gewesen.

André Klahold, Haiger, im Juli 2006

Kurzfassung

Die Idee, inhaltlich verwandte Texte automatisiert in Verbindung zueinander zu setzen, ist nicht neu. Der im Rahmen dieser Arbeit vorgestellte Lösungsansatz verfolgt zwei Hauptziele: automatisiert auf Basis unstrukturierter Texte zu arbeiten und eine hohe Anzahl gleichzeitiger Zugriffe zu unterstützen. Es unterscheidet sich von anderen Verfahren im Wesentlichen durch die Ermittlung des semantischen Abstandes zwischen den Texten auf Basis einer asymmetrischen vorberechneten Distanzmatrix. Die Beziehungen zwischen unstrukturierten Textobjekten werden mittels eines, von der Landessprache unabhängigen, heuristischen Algorithmus zur Merkmalsselektion hergestellt. Die resultierende "Wortwolke" wird dann als Anfrageparameter für die Selektion passender Texte verwendet. Dem Benutzer werden zum gerade angezeigten Text inhaltlich verwandte Texte empfohlen. Die Verarbeitungsgeschwindigkeit des Verfahrens wurde in Form der Laufzeitkomplexität der Algorithmen analysiert. Über einen Zeitraum von 12 Monaten wurden außerdem umfangreiche Praxistests auf der Website eines Industriemagazins durchgeführt, um die Effizienz des Verfahrens im Hinblick auf die Qualität der Empfehlungen zu prüfen. Die Ergebnisse zeigen, dass der vorgestellte Ansatz den manuell erstellten Empfehlungen professioneller Redakteure nahezu ebenbürtig ist. Eine umfangreichere Zusammenfassung findet sich auf Seite 158.

Abstract

The idea to link texts with related content in an automated way is not new. The approach developed and presented here has two main goals: to work automatically on unstructured texts and to support a large number of parallel accesses. It is distinct from other approaches in that it determines the semantic distance between texts on the basis of an asymmetrical pre-calculated distance matrix. The relations between unstructured text objects are generated by a language independent heuristic algorithm for feature selection. The resulting bag of words is used in a query to select matching texts. The user receives recommendations to texts the content of which is related to the text that appears in front of him. Performance was analyzed on basis of the algorithms runtime complexity. Extensive real-life tests over a period of 12 months were conducted on the website of an industrial magazine in order to check the efficiency of the procedure with regard to quality of the recommendations. Results show that the presented approach nearly equals the quality of manual made recommendations by professional editors. A more extensive summary is found at page 164.

Inhaltsverzeichnis

TEIL I - EINLEITUNG

Zusammenfassung – Teil I
Die Motivation der Arbeit wird beschrieben.
Die Zielsetzung der Arbeit wird beschrieben.
Es wird eine erste grobe Einordnung in den Stand der Forschung gegeben.
Der Aufbau der Arbeit wird vorgestellt.

1 Motivation

Das Problem zu einem Thema "passende" Informationen zu finden, ist vermutlich so alt, wie die Fähigkeit des Menschen, Informationen in Schriftform zu erfassen. Eine genauere Definition von "passend" wird im Rahmen dieser Arbeit vorgenommen. Sinngemäß sei "passend" zunächst im Sinne von "inhaltlich verwandt" verstanden.

Dem Alter der Problemstellung entsprechend vielfältig sind die vorhandenen Lösungsansätze. Von den ersten thematisch gegliederten Bibliotheken über umfangreiche Bibliographien bis zu dem von Vannevar Bush [BUS1945] erdachten Hyperlink-Konzept hat sich auch die Technik des Verweisens auf Texte ständig weiter entwickelt. Der aktuellste Ansatz (Stand Juli 2006) mit Reichweite dürfte derzeit das "Semantic Web" [FEN2003] sein.

Mit einer steigenden Zahl von Texten wird die Lösung des Problems nochmals erschwert, da es nicht mehr nur gilt, "passende" Texte zu finden, sondern bei einer Vielzahl "passender" Texte diejenigen mit der höchsten Relevanz zu selektieren und nach eben dieser Relevanz zu sortieren. Wir wollen "Relevanz" im Folgenden im Sinne von "Nutzwert für den Leser" verstehen. Es ist daher offensichtlich keine objektiv messbare, sondern aufgrund der subjektiven Komponente eine nur empirisch fassbare Größe.

Bereits seit Beginn des 20. Jahrhunderts wird die Problematik der "Informationsflut" in der Wissenschaft diskutiert (exemplarisch sei auf [OPP1927] verwiesen). Der amerikanische Trendforscher John Naisbitt prägte den Satz "Wir ertrinken in Informationen, aber hungern nach Wissen" [NAI1982]. Dass die Informationsmenge mittlerweile nicht mehr exponentiell wächst, ist einzig dem bereits jetzt erreichten Maß an Informationsausstoß zu verdanken [MAR2002].

Das konzeptuelle Problem spiegelt sich auch in einem wirtschaftlichen Aspekt wider. Mitarbeiter wenden laut dem Bundesamt für Wirtschaft und Technologie durchschnittlich 35% ihrer Arbeitszeit dafür auf, im Unternehmen vorhandenes Wissen zu finden [BMI2002]. Selbst bei niedrig angesetzten Personalkosten von 35.000 Euro pro Mitarbeiter ergibt sich folglich ein potenzielles Einsparpotential von 12.250 Euro pro Mitarbeiter. Natürlich wird man die "Suchzeit" nicht komplett eliminieren können. Aber bereits eine Halbierung würde umfangreiche Ressourcen für sinnvollere Tätigkeiten freilegen.

Aus einer unüberschaubaren Informationsmenge die "passenden" Texte selektieren zu können, ist daher wichtiger denn je.

2 Zielsetzung

Das im Rahmen dieser Arbeit vorgestellte Verfahren soll es ermöglichen, aus einer großen Menge von Texten die zu einem vorliegenden Text "passenden" Texte zu selektieren und dem Benutzer automatisch anzuzeigen. Dabei sollen die im Folgenden aufgeführten Prämissen (i) – (iv) mit den angegebenen Nebenbedingungen (a,b) erfüllt werden:

(i) Akzeptanz beim Benutzer
   a. Konstant auch über längere Zeitintervalle

(ii) Garantiert schnelle Antwortzeiten
   a. Auch bei großen Textmengen
   b. Auch in Hochlastumgebungen (hohe Nutzungsrate)

(iii) Erschließung aller vorhandenen Quellen
   a. Unstrukturierte Texte ohne Metadaten
   b. Unterschiedliche Dateiformate

(iv) Automatisierte Verarbeitung
   a. Kein manueller Eingriff durch den Autor
   b. Kein manueller Eingriff durch den Benutzer

In der Praxis sind viele bereits existierende Lösungsansätze aufgrund der damit einhergehenden Kosten durch nicht erfülltes (iv) insbesondere für KMU[1] nicht wirtschaftlich einsetzbar. Wenn man vor der Nutzung einer Software das System "trainieren" oder gar die Texte kategorisieren muss, so lässt sich – auch ohne eventuelle Software-Kosten einzubeziehen – nur bei sehr großen Benutzerzahlen auf absehbare Zeit ein *Return on investment* erzielen.

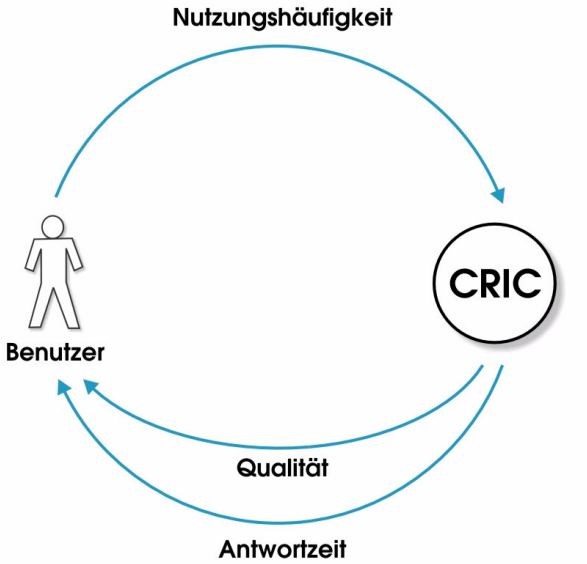


Abbildung 1: Die gesetzten Prämissen sollen potenzielles Nutzungsvolumen und die Nutzungsfrequenz maximieren

Durch die oben angeführten Prämissen (iii) und (iv) wird das betriebswirtschaftlich sinnvolle und damit real vorhandene Einsatzgebiet der Lösung erheblich erweitert.

[1] Klein- und Mittelständische Unternehmen. Per Definition sind dies Unternehmen bis 250 Mitarbeiter [EUK2003]

3 Einordnung

Die vorliegende Arbeit stellt ein neues Verfahren für ein *Empfehlungssystem* vor. Dem Benutzer werden zum gerade angezeigten Text inhaltlich verwandte Texte empfohlen. Der im Rahmen dieser Arbeit vorgestellte Ansatz unterscheidet sich von ähnlichen Verfahren[2] im Wesentlichen dadurch, dass eine Heuristik zum Einsatz kommt, die ein TF-IDF-Derivat (siehe Seite 19) mit Eigenschaften der Textstruktur verbindet und darauf sprachunabhängig[3] eine asymmetrische vorberechnete Distanzmatrix[4] aufbaut. Als Basis können beliebige unstrukturierte Texte verwendet werden. Insbesondere werden keine Metadaten, aber auch keine Thesauri oder Korpora benötigt.

Die unterschiedlichen Lösungsansätze für das Problem "passende" Texte zu finden, lassen sich in zwei große Gruppen unterteilen: die "push"- und die "pull"-Ansätze. Zu Ersteren zählt man Empfehlungssysteme (Recommender Systems) aller Art, zu Letzteren die unterschiedlichen Suchverfahren.

Die bekanntesten Vertreter der Empfehlungssysteme basieren auf dem Content-based Filtering (CBF), dem Collaborative Filtering (CF) oder hybriden Varianten. Diese Ansätze werden in der Literatur ausgiebig behandelt ([RES1997], [BAD2000] et cetera).

3.1 Content-based Filtering (CBF)

Beim CBF werden die Daten und/oder Metadaten eines Textes verwendet, um verwandte Texte zu finden. Dem Benutzer werden dann die Texte angeboten, die den von ihm implizit oder explizit bewerteten Texten oder dem gerade angezeigten Text am ähnlichsten sind. Ein Verfahren, das diesen Ansatz wählt, ist "The information lens" [MAL1986]. Es setzt allerdings beim Autor der Texte eine explizite Strukturierung derselben voraus. Dem Nutzer werden dann über strukturierte Filter passende Texte zugespielt. Ein anderes Beispiel für CBF ist INFOSCOPE [FIS1991], das keine Vorarbeit beim Autor voraussetzt und den Nutzer beim Aufbau und Anpassen der Filter unterstützt. Weitere prominente Vertreter sind Newsweeder [LAN1995], InfoFinder [KRU1997], News Dude [BIL1999] und LIBRA [MOO2000].

3.2 Collaborative Filtering (CF)

Das CF nutzt anstelle der Verwandtschaft von Texten die Ähnlichkeit von Benutzerprofilen. Empfehlungen für einen Benutzer werden aus dem Verhalten anderer Benutzer mit ähnlichem Profil gewonnen. Ausgehend von Hey [HEY1989] existieren viele Beispiele für Collaborative Filtering. Einige davon sind Tapestry [GOL1992], GroupLens [RES1994], Ringo [SHA1995], Siteseer [RUC1997] und Jester [GOL2000].

3.3 Hybrid-Systeme

In der Praxis sind auch Hybrid-Systeme zu finden, die Content-based Filtering und Collaborative Filtering verbinden. Bekannte Vertreter dieser Gruppe sind INFOS [MOC1996], Fab [BAL1997] und Tango [CLA1999].

---

[2] Eine genaue Abgrenzung auf Basis einer Systematisierung der unterschiedlichen Lösungsansätze findet sich in "5 Stand der Forschung"; hier wird nur eine erste Abgrenzung zu ähnlichen Verfahren, also solchen mit vergleichbarem Lösungsansatz, vorgenommen.

[3] Die Sprachunabhängigkeit gilt für romanische und indogermanische Sprachen.

[4] Für die Proximität oder inhaltlich Nähe zweier Informationsobjekte wird im Rahmen von Empfehlungssystemen oft der Begriff der Metrik verwendet. Da aber nicht alle Verfahren eine symmetrische Proximität liefern, wird der Begriff der Distanz verwendet.

4 Aufbau der Arbeit

Die vorliegende Arbeit kann in zwei große Bereiche sowie einen Prolog (Teil I), einen Epilog (Teil IV) und einen Anhang (Teil V) aufgeteilt werden.

In Teil II werden zunächst die wichtigsten Empfehlungssysteme vorgestellt und einer systematischen Einordnung unterzogen (Abschnitt 5 "Stand der Forschung").

Teil III besteht aus einer detaillierten Beschreibung des im Rahmen dieser Arbeit vorgestellten Verfahrens (Abschnitt 6 "Der gewählte Lösungsansatz"), einer theoretischen Bewertung des Laufzeitverhaltens (Abschnitt 7 "Laufzeitkomplexität" und Abschnitt 8 "Das theoretische Laufzeitverhalten") und einer praktischen Evaluation des Laufzeitverhaltens (Abschnitt 9 "Das reale Laufzeitverhalten") und der Benutzerakzeptanz (Abschnitt 10 "Qualitative Evaluation").

In Teil IV wird dann eine Zusammenfassung in deutscher (Abschnitt 11 "Zusammenfassung") und englischer Sprache (Abschnitt 12 "English Summary") vorgenommen. Abschließend wird ein Ausblick auf die weiteren Entwicklungen gegeben (Abschnitt 13 "Ausblick").

Abbildung 2: Struktur der vorliegenden Arbeit

TEIL II – GRUNDLAGEN

Zusammenfassung – Teil II
Die wichtigsten der verwendeten Begriffe werden definiert.
Die wesentlichen Eigenschaften von Empfehlungssystemen werden diskutiert und die Verfahren des Content-based Filtering und des Collaborative Filtering vorgestellt.
Es wird eine Klassifikation für Empfehlungssysteme auf Basis unterschiedlicher Merkmale erarbeitet.
Es werden rund 50 bedeutende Vertreter der verschiedenen Varianten von Empfehlungssystemen vorgestellt. Dabei liegt der Schwerpunkt auf Content-based Filtering Verfahren.

5 Stand der Forschung

Es finden sich bereits zahlreiche Ansätze, die Problemstellung zu einem Thema "passende" Informationen zu finden, zu lösen. Man bezeichnet Verfahren zur Empfehlung "passender" Informationen allgemein als "Recommender Systems" (Empfehlungssysteme). Eine wichtige Bedingung für den praktischen Nutzen solcher Systeme besteht offensichtlich in der durch die Qualität und Geschwindigkeit des Verfahrens bedingten Akzeptanz der Benutzer.

Abgrenzung zu Suchverfahren

Der wesentliche Unterschied zwischen Empfehlungssystemen und Suchverfahren besteht darin, dass Empfehlungssysteme aktiv "passende" Informationen anbieten, Suchverfahren hingegen eine Eingabe seitens des Benutzers verlangen. Der Übergang zwischen beiden ist allerdings fließend. Verwendet man beispielsweise einen vom Nutzer vorgegeben Text – zum Beispiel in Form eine Frage – im Verein mit seinem Profil zur Selektion verwandter Texte, so liegt ein Mischverfahren vor.

5.1 Begriffsdefinitionen

Im Folgenden sollen wichtige Begriffe definiert werden, um deren eindeutige Verwendung zu gewährleisten.

Textbasis

Eine *Textbasis* stellt eine Menge von Texten dar: $C=\{T_1,\ldots,T_m\}$.

Korpus

Ein *Korpus* besteht aus sprachlichen Daten, die einer sprachwissenschaftlichen Analyse als Grundlage dienen [GLU2000].

RDF

Das Akronym *RDF* steht für *Resource Description Framework*. Es ist eine Metasprache zur Beschreibung einfacher semantischer Zusammenhänge, die neben dem für den Menschen sichtbaren Text zur Verfügung gestellt wird. Dadurch wird eine maschinelle Verarbeitung ermöglicht.

OWL

Die Abkürzung *OWL* steht für *Web Ontology Language* und ist eine semantische Auszeichungssprache um Ontologien im Internet zu veröffentlichen. Die OWL ist ein W3C-Standard und basiert auf dem RDF.

URL

Das Akronym *URL* steht für *Uniform Resource Locator*. Ein URL adressiert eine Ressource durch das primäre Zugriffsprotokoll (beispielsweise `http`) und eine eindeutige Ortsangabe. Dabei muss letztere einer speziellen Syntax gerecht werden [BER1994].

Webseite

Eine *Webseite* ist ein in HTML (Hypertext Markup Language) codiertes Informationsobjekt.

Informationsobjekt

Ein *Informationsobjekt* ist im Rahmen dieser Arbeit eine aus einem Volltext (im Folgenden auch *Objekt* oder *Dokument* genannt) und gegebenenfalls aus strukturierten Daten bestehende Informationseinheit. Die in dieser Arbeit relevanten Informationsobjekte sind:

- Webseiten
- Bücher (der Volltext ist hier der Beschreibungstext)
- Filme (der Volltext ist hier der Beschreibungstext)
- Newsgroup-Texte
- Musik (der Volltext ist hier der Beschreibungstext)

Multimediale Daten wie Bilder, bewegtes Bild oder Audio spielen im Rahmen dieser Arbeit keine Rolle, wenn sie nicht zusätzliche Informationen in Form von Metadaten oder unstrukturiertem Text besitzen.

Session

Eine *Session* ist definiert als die Sequenz von aufeinander folgenden Anfragen, die ein Benutzer während des Besuches einer Website ausführt [MEN1999].

Bookmark

Ein *Bookmark* ist die dauerhafte Speicherung einer Webseite in Form des URL und eines Namens.

Hyperlink

Ein *Hyperlink* ist ein Verweis von einer bestimmten Stelle einer Webseite auf eine andere Webseite oder eine bestimmte Stelle in einer anderen Webseite. Die Hyperlinks sind das wesentliche Merkmal des Hypertextes. Im Vergleich zu klassischen Querverweisen, kann man durch Anklicken des Hyperlinks das verlinkte Ziel aufrufen.

Suchmaschine

Eine *Suchmaschine* ist eine Software zur Recherche von unstrukturierten Texten, die auf einem einzelnen Computer oder einem Computernetz wie dem Internet in unterschiedlichen Formaten und an unterschiedlichen Orten vorliegen. Nach Eingabe einer Suchanfrage liefert eine Suchmaschine eine Liste mit Verweisen auf potenziell relevante Texte. Diese werden dabei in der Regel durch eine automatisch aus dem Text gewonnene Kurzbeschreibung ergänzt.

Suchanfrage

Eine *Suchanfrage* (kurz *Anfrage;* engl. *Query*) ist ein aus einem oder mehreren Worten bestehender Text, der von einer Suchmaschine als Ausgangswert für eine Recherche verwendet wird. Potenziell kann eine Suchanfrage auch logische Operatoren enthalten (`UND`, `NICHT` et cetera), mit denen komplexere Anfragen erzeugt werden können.

Synonym

Ein *Synonym* ist ein Wort, das zumindest teilweise die gleiche Bedeutung wie ein Wort mit unterschiedlicher Schreibweise hat. Beispiele für Synonyme sind:

- Bildschirm - Monitor - TFT
- Bank - Geldinstitut
- Computer - PC- Laptop

Benutzerbewertung

Eine *Benutzerbewertung* für ein Informationsobjekt ist im einfachsten Fall ein binärer Wert ("interessant", "nicht interessant"). Alternativ erfolgt die Bewertung auf einer Skala (beispielsweise "nicht interessant", "etwas interessant", "interessant", "sehr interessant").

Benutzerprofil

Ein *Benutzerprofil* (kurz *Profil*) besteht aus einer Menge von Informationsobjekten, die gegebenenfalls mit einer Benutzerbewertung und dem Zeitpunkt der Aufnahme ins Profil versehen sind. Alternativ kann ein Benutzerprofil auch aus Worten bestehen, die das Interessensgebiet des Benutzers widerspiegeln.

Flüchtiges und persistentes Profil

Ein *flüchtiges Profil* ist nur während einer begrenzten Zeitspanne existent. In der Regel handelt es sich dabei um die Dauer einer Session.

Ein *persistentes Profil* bleibt auf Dauer und über verschiedene Sessions hinweg erhalten. Dazu muss der Benutzer erkannt werden, was in der Regel eine Anmeldung erforderlich macht.

Kontext

Unter dem *Kontext* eines Benutzers sollen das oder die gerade angezeigten Informationsobjekte zu verstehen sein.

Wortvektor

Der Volltext eines Informationsobjektes oder ein aus Worten bestehendes Benutzerprofil kann durch einen Wortvektor dargestellt werden. Dabei stellt jede Komponente des Vektors ein bestimmtes Wort mit dessen Gewichtung dar. Durch diese Vereinheitlichung werden Texte in Form der Wortvektoren im Vektorraum

mathematisch vergleichbar. Als Sonderfall geben binäre Wortvektoren nur an, ob ein Wort in einem Text vorhanden ist.

Wortvektor als Profil

Analog zu den Informationsobjekten können auch Benutzerprofile als Wortvektoren dargestellt werden. Dazu werden die einem Profil zugehörigen Texte (beziehungsweise deren Wortvektoren) summiert (beispielsweise bei PRES [MET2000]) oder deren Mittel gebildet (beispielsweise bei Syskill & Webert [PAZ1996]).

Distanz und Ähnlichkeit

Es gibt formal betrachtet einen Unterschied zwischen einem Distanzmaß und einem Ähnlichkeitsmaß. Ein Distanzmaß auf einer Menge C von Texten ist eine reelle Funktion $dist(T_1,T_2)$ mit $dist(T_1,T_2)=0$ falls $T_1=T_2$. Ein Ähnlichkeitsmaß ist eine reelle Funktion $sim(T_1,T_2)$ mit $sim(T_1,T_2)=1$ falls $T_1=T_2$. Neben der Reflexivität gilt auch die starke Reflexivität: $dist(T_1,T_2)=0 \Rightarrow T_1=T_2$ und $sim(T_1,T_2)=1 \Rightarrow T_1=T_2$. Im Rahmen dieser Arbeit wird keine metrische Distanz gefordert. Damit sind zusätzliche Bedingungen metrischer Distanzen (Symmetrie et cetera) nicht zwingend gegeben.

Trainingsdaten

Im Kontext der Empfehlungssysteme stellen Trainingsdaten (auch *Lernmenge*) eine möglichst repräsentative Sammlung von Texten des betrachteten Korpus beziehungsweise der Textbasis dar. Auf Basis der Traningsdaten und manuell vorgenommener Klassifikation leiten verschiedene Klassifikationsverfahren automatisch (maschinelles Lernen) Klassifikationsregeln ab.

Typographie

Die Typographie ist die Gestaltung von und mittels Drucklettern [GLU2000]. Dabei werden Mikrotypographie und Makrotypographie unterschieden. Die Mikro- oder auch Teiltypographie umfasst die Gestaltung von Buchstaben, Worten und Zeilen. Dabei spielen Schriftgröße, Zeichen-, Wort- und Zeilenabstand eine wesentliche Rolle. Die Makrotypographie ist hingegen die Gesamtkonzeption in Form der Schriftarten, des Satzspiegels (Nutzfläche auf dem Papier), dem Aufbau der Seiten und der Bildgestaltung.

5.2 Empfehlungssysteme (Recommender Systems)

Ein Empfehlungssystem liefert einem Benutzer aufgrund seines Profils oder des aktuellen Kontextes "ähnliche" Informationsobjekte. Die bekanntesten Verfahren dafür sind das "Collaborative Filtering" (CF) und das "Content-based-Filtering" (CBF), sowie deren Mischform. Das CF nutzt die "Ähnlichkeit" (in der Regel auf Basis eines Ähnlichkeits- oder Distanzmaßes zwischen Profil-Vektoren ermittelt) zwischen dem Benutzerprofil und den Profilen anderer Benutzer. Das CBF nutzt primär den Kontext, um Informationsobjekte mit ähnlichen Eigenschaften zu finden; also beispielsweise "Geländewagen", falls der Kontext ein "Jeep" ist oder "Rechtsvorschriften für den Datenschutz", falls der Kontext ein Text über "Datenschutz im Internet" ist. Hybrid-Varianten verbinden CF und CBF und nutzen in der Regel sowohl Benutzerprofile als auch den Kontext.

Precision und Recall

Die qualitative Güte der Empfehlungen eines Empfehlungssystems wird in der Regel mit den Werten *Precision* und *Recall* angegeben.

Werden aus einer Menge M von Dokumenten n Dokumente empfohlen, so gibt die *Precision* an, welchen Prozentsatz die relevanten Dokumente in der Empfehlung ausmachen. Als *Recall* bezeichnet man hingegen den Prozentsatz der relevanten Dokumente, die empfohlen wurden. Daher ist *Precision* ein Maß für die Güte von Empfehlungen und *Recall* eines für die Berücksichtigung potenziell relevanter Texte.

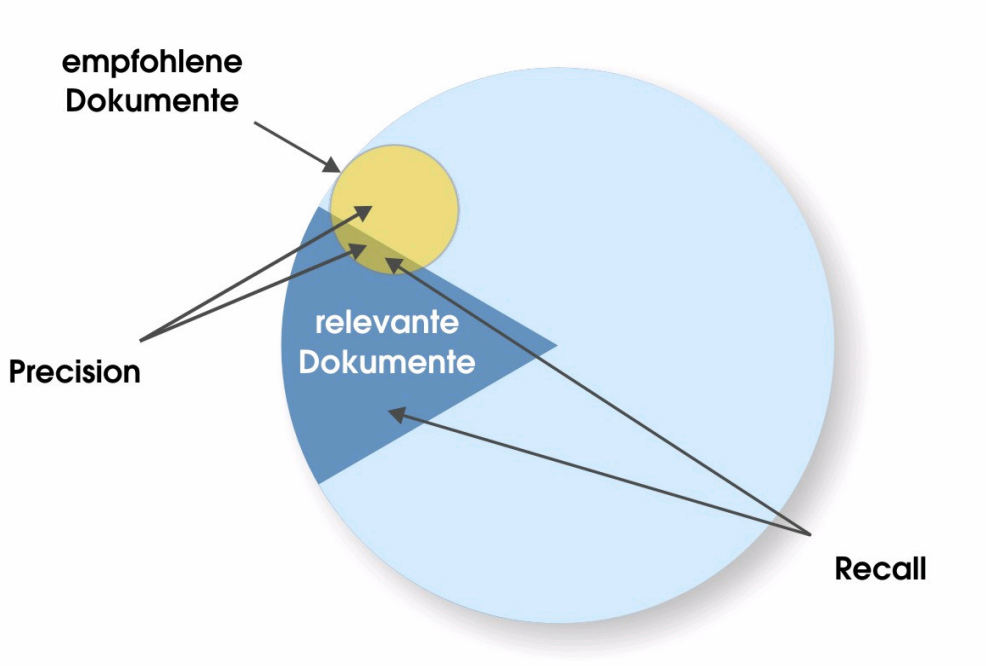


Abbildung 3: "Precision" ist das Verhältnis von relevanten zu insgesamt empfohlenen Texten und "Recall" das Verhältnis von empfohlenen relevanten Texten zu insgesamt relevanten Texten.

Gibt es beispielsweise 10 relevante Dokumente und sind unter 15 empfohlenen Dokumenten 5 relevante, so hat *Precision* einen Wert von 33% und *Recall* einen Wert von 50%.

Eine konkrete Ermittlung von *Precision* und *Recall* ist allerdings problematisch, da das Kriterium "relevant" in der Regel subjektiv ist. Man versucht dieses Manko dadurch zu eliminieren, dass man benutzerseitig manuell als relevant ausgezeichnete Dokumente verwendet und diese mit den maschinell gewonnenen Empfehlungen vergleicht.

Wesentliche Konzepte

Für eine systematische Betrachtung sollen die Konzepte für Empfehlungssysteme nun zunächst in Klassen aufgeteilt werden.

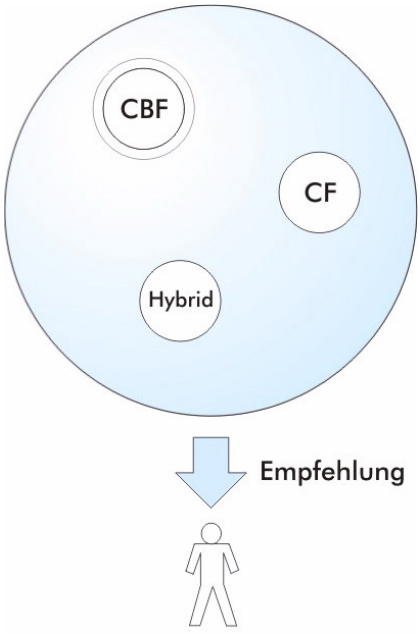


Abbildung 4: Klassifikation von Empfehlungsverfahren

Die wesentliche Differenzierung erfolgt durch eine Aufteilung der Verfahren in solche, auf Basis des "Content-based Filtering" (CBF) und des "Collaborative Filtering" (CF) sowie in hybride Ansätze, die beide Verfahren mischen. Diese Varianten werden in der Literatur ausgiebig behandelt ([MOO2000], [BAS1998], [ALS1998], [FER2002], [HER2000], [SAR2001], [SCH2002], [MON2003]). Das CBF fokussiert, wie oben beschrieben, die Inhalte, das CF hingegen baut auf einer Auswertung des Benutzerverhaltens auf.

5.2.1 Content-based Filtering (CBF)

Beim Content-based Filtering (CBF) werden die Daten und/oder Metadaten eines oder mehrerer Informationsobjekte verwendet, um verwandte Informationsobjekte zu finden. Dem Benutzer werden dann die Informationsobjekte angeboten, die dem von ihm gerade gesehen Informationsobjekt und/oder den in der Vergangenheit von ihm implizit oder explizit bewerteten Informationsobjekten am ähnlichsten sind.

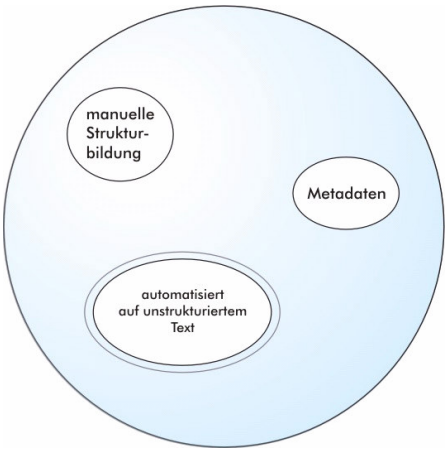


Abbildung 5: Content-based Filtering (CBF)

Einige Beispiele für Content-based Filtering sind Newsweeder [LAN1995], InfoFinder [KRU1997] oder NewsDude [BIL1999].

Einige CBF Verfahren benötigen Metadaten um Empfehlungen für unstrukturierte Texte aussprechen zu können. Andere erfordern eine manuelle Strukturbildung. Im Folgenden sollen daher die wichtigsten Klassen von Metadaten und manueller Strukturbildung vorgestellt werden.

5.2.1.1 Relationale und autonome Metadaten

Bei den Metadaten muss zwischen relationalen und autonomen Metadaten unterschieden werden. Erstere setzen Informationsobjekte in eine bestimmte Relation zu anderen Informationsobjekten in verschiedenen Ordnungstrukturen, wohingegen Letztere autonom am Informationsobjekt mitgeführt und genutzt werden können. Nachfolgend werden einige Arten von Metadaten aufgeführt:

- Schlagwort [autonom]
- Taxonomie [relational]
- Thesaurus [relational]
- Topic Map [relational]
- Ontologie [relational]

Schlagwort

Die Zuordnung von Schlagworten erfolgt in der Regel, um eine eindeutige Suche nach Informationsobjekten zu ermöglichen. Unter einem Schlagwort wird ein natürlichsprachlicher Ausdruck verstanden, der den Inhalt eines Textes möglichst kurz und präzise wieder gibt. Komplexe Inhalte werden in der Regel durch mehrere Schlagworte beschrieben.

Grundlage für eine systematische Schlagwortvergabe ist in der Praxis oft ein Verzeichnis der zu verwendenden Schlagwörter. Neben individuellen gibt es auch genormte Verzeichnisse wie beispielsweise die Schlagwortnormdatei (SWD) [DEU2005].

Taxonomie

Eine Taxonomie ist ein Klassifikationssystem das Gruppen von Informationsobjekten durch eine Einordnung in eine Hierarchie bildet:

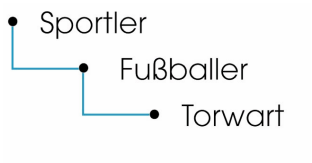


Abbildung 6: Beispielhafte Struktur einer Taxonomie

Thesaurus

Ein Thesaurus besteht aus einer Sammlung von Begriffen, die in Beziehung zueinander stehen. Neben der Hierarchie der Taxonomie in Form von Ober- und Unterbegriffen verwaltet ein Thesaurus vorrangig Synonyme:

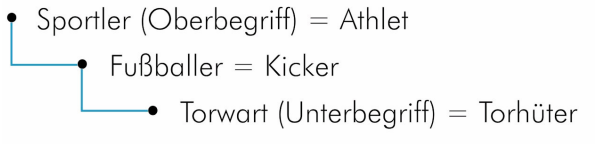


Abbildung 7: Beispielhafte Struktur eines Thesaurus

Topic Map

Der in ISO/IEC 13250 definierte *Topic Map Standard* bietet eine standardisierte und implementationsunabhängige Notation, um "Topics" (Themen) und deren Beziehung zueinander und zu Texten zu definieren. Topic Maps fassen Texte durch Adressierung ("occurence") zu einer Gruppe zusammen, die einem bestimmten Thema ("topic") zugeordnet ist. Zwischen den Themen ("topics") selbst bestehen wiederum Beziehungen ("associations"). Durch Typisierung werden Themen ("topics") zu gleichartigen Gruppen zusammengefasst (beispielsweise "Städte", "Menschen" et cetera). Man kann eine Topic Map als einen mehrdimensionalen Themenraum betrachten, in dem die Orte durch Themen repräsentiert werden und die semantischen Abstände zwischen zwei Themen (auf dem "association" Pfad) durch die Zahl der dazwischen liegenden anderen Themen bestimmt werden.

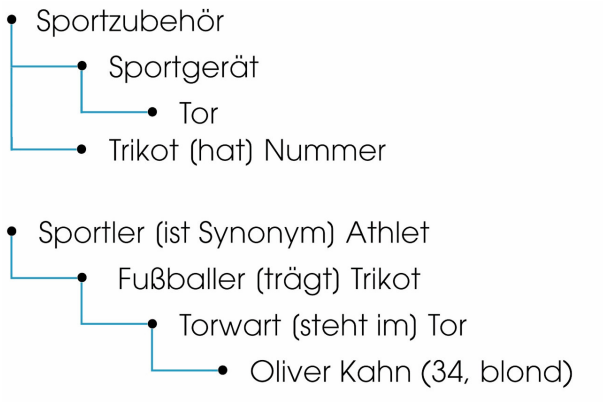


Abbildung 8: Beispielhafte Struktur einer Topic Map

Ontologie

Eine Ontologie definiert Objekte und deren Relationen zueinander (zum Beispiel mittels RDF). Sie definiert ein Datenmodell mit Entitätstypen und Entitäten (zum Beispiel mittels XML; unterstützt Mehrsprachigkeit) und kann Semantik und einfache Abhängigkeiten mit Hilfe des Ontologie Vokabulars beschreiben (zum Beispiel mittels OWL). Logische Zusammenhänge können mittels Regeln ausgedrückt werden (zum Beispiel mittels F-Logic):

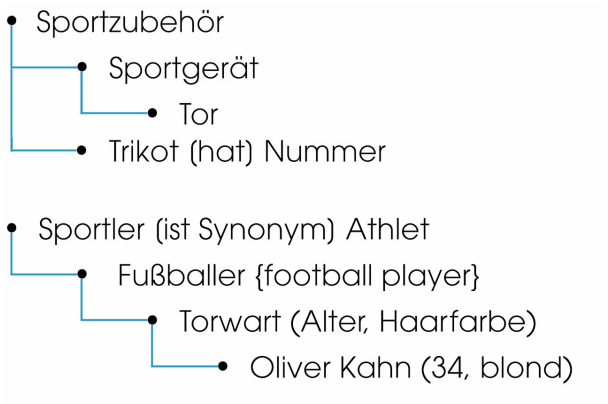


Abbildung 9: Beispielhafte Struktur einer Ontologie. Hinzu kommen Regeln wie beispielsweise " Fußballer trägt Trikot => Fußballer hat Nummer".

5.2.1.2 Textanalyse

Verfahren die auf Metadaten oder manueller Strukturbildung basieren, setzen den manuellen Eingriff seitens der Textautoren und/oder der Benutzer voraus und erfüllen nicht die Zielsetzung "iv" auf unstrukturierten Daten zu arbeiten (siehe Seite 4). Ein Verfahren, das diese Vorgabe erfüllt, ist beispielsweise der Remembrance Agent [RHO1999], der verwandte Texte auf Basis des Vektorabstandes als ad hoc Anfrage ermittelt. Auch Watson [BUD1999] und Margin Notes [RHO2000] erfüllen die Zielsetzung "iv", indem sie die wichtigsten Begriffe eines Textes ermitteln und diese dann zur ad hoc Suche nach verwandten Texten nutzen; es findet aber keine Vorberechnung statt, was in Hochlastszenarien problematisch ist.

Bei der Textanalyse spielen einige Konzepte und Verfahren eine wichtige Rolle, die nun kurz vorgestellt werden sollen.

Stemming

Worte wie "schreiben" kommen in der Sprachanwendung in zahlreichen *Wortformen* vor ("geschrieben", "schrieb", "schreibt", "Schreiber" et cetera). Die abstrakte Gruppe der begrifflich zusammengehörigen Wortformen bezeichnet man als *Lexem* und verwendet zur Benennung in der Regel (sprachabhängig) den Wortstamm ("*schreib*").

Zwei bekannte Stemming-Verfahren sind das von Lovins [LOV1968] und Porter [POR1980]. Einen guten Überblick und einen Vergleich von drei Verfahren gibt [PAI1994].

Durch Stemming wird versucht, Wortformen automatisiert auf Lexeme zurückzuführen, um begrifflich zusammengehörige Wortformen durch dieselbe Zeichenkette (das Lexem) darzustellen. In der Regel wird dies durch Heuristiken wie der Folgenden erreicht:

- Vereinheitlichung von Umlauten und "ß/ss"
- Regelbasiertes Abtrennen von Suffixen (beispielsweise "schreib-en")

Damit werden "Pseudo-Lexeme" erzeugt, die aber gerade in der deutschen Sprache mit ihren stark unterschiedlichen Wortformen ("schrieb", "Schreiber") problematisch sind. Zusätzlich gehen beim Stemming Informationen verloren, die durch die Wortform codiert werden (beispielsweise die zeitliche Information (Vergangenheit) in "geschrieben"). Es ist empirisch belegt, dass einfaches Stemming im Mittel keinen positiven Einfluss auf Empfehlungen hat [HUL1996].

Inverser Index

Ein inverser Index (*reverse index*) ist eine Datenstruktur, die für jedes Wort, das in mindestens einem analysierten Text vorkommt alle Texte aufführt, in denen das Wort vertreten ist

Stoppwortliste

Eine Stoppwortliste `SWL` definiert die Worte, die für eine Textanalyse keine Bedeutung haben. Das sind in der Regel die am häufigsten vorkommenden Worte einer Sprache (feste Stoppwortliste) oder einer Menge von Dokumenten (berechnete Stoppwortliste). Eine Stoppwortliste reduziert den Umfang eines *inversen Index* signifikant. Nach Zipf´s Gesetz [ZIP1949], das besagt, dass Rang (Position in der Reihenfolge, der nach ihren Häufigkeiten des Vorkommens in Texten absteigend sortierten Worte) und Anzahl (über alle Dokumente) eines

Wortes antiproportional sind, können schon weniger als 50 Stoppworte mit dem höchsten Rang das Volumen des inversen Index halbieren.

Token-Bildung

Das Zerlegen eines Textes in zusammengehörige Zeichenketten (Worte, Zahlen et cetera) bezeichnet man als Token-Bildung. Dies ist meist die Basis zur weitergehenden Textanalyse.

Kollokation & Kookurrenz

Kookurrenz ist die Wahrscheinlichkeit (korpus-abhängig, empirisch ermittelt in Form der Häufigkeiten) mit der zwei Worte w_1 und w_2 in einem Textfenster F_n von n Worten gemeinsam auftreten:

$$K_n(w_1, w_2) = P(w_1, w_2, F_n)$$

Eine Kollokation liegt vor, wenn die Kookurrenz zweier Worte signifikant über dem Durchschnittswert aller $K_n(w_i, w_j)$ liegt.

5.2.2 Collaborative Filtering (CF)

Das CF nutzt anstelle der Verwandtschaft von Texten die Ähnlichkeit von Benutzerprofilen. Die Profile repräsentieren das Benutzerverhalten in Form der Nutzung von Informationsobjekten (Texte, Produkte, Bilder et cetera). Daraus resultiert eine Benutzer-Objekt-Matrix:

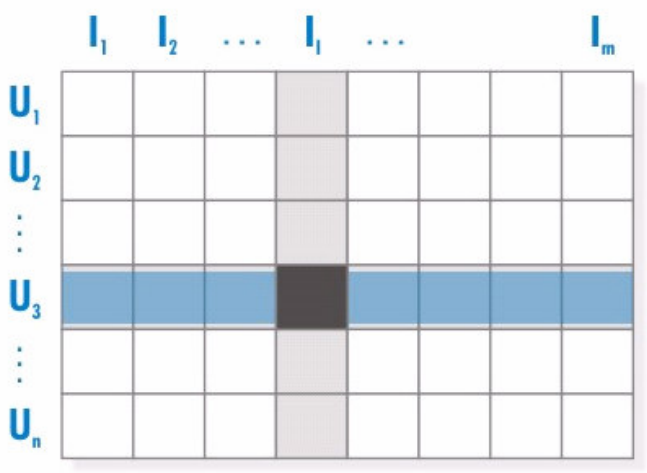


Abbildung 10: Die CF-Matrix der Benutzer-Objekt-Beziehungen. Mit $U_1,\ldots,U_n$ als Benutzer und $I_1,\ldots,I_m$ als Informationsobjekte. Eine Zelle $[U_x,I_y]$ der Matrix stellt die implizite oder explizite Bewertung des Informationsobjektes I_y durch den Benutzer U_x dar. Die Bewertung kann boolesch (angesehen/nicht angesehen beziehungsweise. gut/schlecht) oder auf einer diskreten Skala erfolgen.

Beispiele für CF Verfahren sind Ringo [SHA1995], Siteseer [RUC1997] und GroupLens [KON1997]. Empfehlungen für einen Benutzer werden in der Regel aus dem Verhalten anderer Benutzer mit ähnlichem Profil gewonnen. Das zugrunde liegende Konzept des CF (siehe auch [SAR2001]) soll nun kurz vorgestellt werden.

5.2.2.1 Der benutzerbezogene Basis-Algorithmus

Um für einen Benutzer "vorherzusagen", welche Objekte ihn interessieren könnten und ihm diese anzubieten, arbeiten benutzerbezogene CF-Verfahren im Wesentlichen wie folgt:

Auf der Basis von n Benutzern und m Objekten wird die Matrix $R=(r_{ij})$ mit $i=1..n$ und $j=1..m$ erzeugt. Der Wert r_{ij} repräsentiert die "Bewertung" des Objektes j durch den Benutzer i. Diese Bewertung kann explizit durch den Benutzer erfolgen (beispielsweise "sehr gut, "gut" et cetera) oder aber implizit aus seinem Verhalten (beispielsweise dem Kauf eines Produktes oder dem Lesen eines Textes) abgeleitet werden.

Die Aufgabe besteht darin, für ein Informationsobjekt I und den Benutzer U (der das Objekt noch nicht gesehen hat) zu bewerten, wie hoch die Relevanz R für ihn ist: $R(I,U)$.

Dazu wird beim benutzerbezogenen CF für alle Informationsobjekte I_n mit $n=1\ldots n$ auf Basis des Verhaltens von Benutzern, die dem Benutzer U_x am ähnlichsten sind, berechnet wie groß die Relevanz (das erwartete Interesse) $R(U_x,I_y)$ dieser Objekte für U_x sein wird. Die Ähnlichkeit wird auf Basis von Distanz- oder Ähnlichkeits-

maßen ermittelt (siehe 5.3.4). Im zweiten Schritt werden dann die Objekte mit dem höchsten Relevanz-Wert zur Auswahl gestellt.

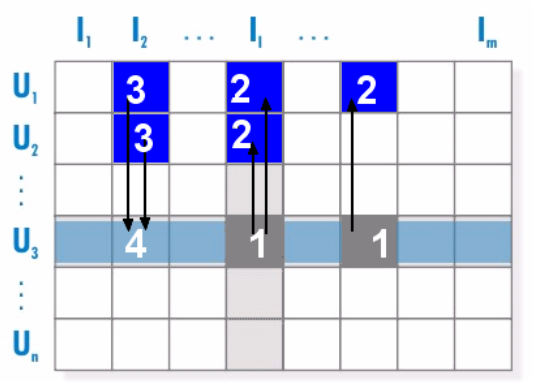


Abbildung 11: Benutzerbezogenes CF Konzept; aufgrund der Bewertungen des Benutzers (1=U_3) werden ähnliche Benutzer (2=U_1, U_2) gesucht. Die von den meisten ähnlichen Benutzern als "gut" bewerteten Informationsobjekte (3) werden dem Benutzer empfohlen (4).

5.2.2.2 Der objektbezogene Basis-Algorithmus

Um für einen Benutzer "vorherzusagen" ob Objekt I_y ihn interessieren könnte und ihm dieses anzubieten, arbeiten objektbezogene CF-Verfahren im Wesentlichen wie folgt:

Auf der Basis von n Benutzern und m Objekten wird wiederum die Matrix $R=(r_{ij})$ mit $i=1..n$ und $j=1..m$ erzeugt.

Auch hier besteht die Aufgabe darin, für ein Informationsobjekt I_y und den Benutzer U_x (der das Objekt noch nicht gesehen hat) zu bewerten, wie hoch die Relevanz R für ihn ist: $R(I_y, U_x)$.

Dazu werden zunächst auf Basis der vom Benutzer bereits bewerteten Objekte $I_1, \ldots, I_n$ die Ähnlichkeiten S zu I_y berechnet. Die Ähnlichkeit wird dadurch bestimmt, dass aus allen Objektpaaren (I_y, I_1), …, (I_y, I_n) (kurz $(I_y, I_{1..n})$) aufgrund der Bewertungen von anderen Benutzern, die jeweils beide Objekte bewertet haben, ein Vektor gebildet wird, der dann für die Ähnlichkeitsbestimmung verwendet wird. Haben alle Benutzer I_y und I_j jeweils identisch bewertet, ist die Ähnlichkeit am größten.

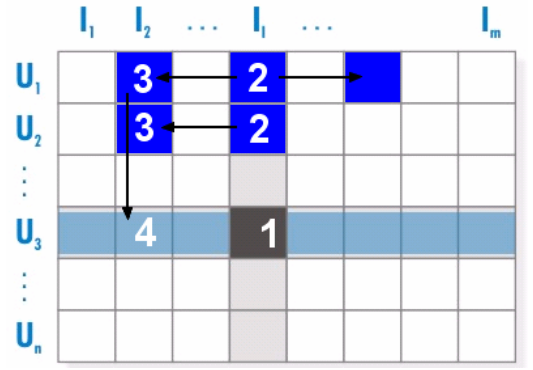


Abbildung 12: Objektbezogenes CF Konzept; aufgrund der gut bewerteten Informationsobjekte des Benutzers (1=U_3) werden in der Bewertungsmatrix R alle Objektpaare in denen eines dieser Informationsobjekte vorkommt selektiert (2). Die am besten bewerteten anderen Objekte in diesen Paaren (3=I_2) werden dem Benutzer dann empfohlen (4).

Die Relevanz $R(I_y, U_x)$ wird dann beispielsweise in Form des gewichteten Durchschnitts der Bewertungen des Benutzers für die I_y ähnlichsten Objekte ermittelt. Je ähnlicher ein Objekt I_j dem Objekt I_y ist (je größer also $S(I_y, I_j)$) , desto stärker fließt die Bewertung durch den Benutzer für dieses Objekt ein:

$$R(I_y, U_x) = \frac{\sum_{j=1\ldots n} S(I_y, I_j) * R(I_j, U_x)}{\sum_{j=1\ldots n} S(I_y, I_j)}$$

Dabei sind $I_{j=1..n}$ die durch den Benutzer bewerteten Informationsobjekte.

5.2.2.3 Modell- und speicherbasiertes Verfahren

Beim speicherbasierten Ansatz wird die Relevanz `R(I,U)` auf Basis der kompletten CF-Matrix berechnet. Beim modellbasierten Verfahren tritt anstelle der CF-Matrix ein vereinfachtes Modell wie zum Beispiel eine "Benutzer-Cluster" Matrix. Dabei werden "ähnliche" Benutzer" gruppiert und damit eine deutlich einfachere Matrix geschaffen. Der auf den ersten Blick überlegene modellbasierte Ansatz hat allerdings den Nachteil, dass neue Daten aufwändiger hinzugefügt werden müssen und dass durch die Vereinfachung auch Informationen für die Entscheidungsfindung verloren gehen können.

5.2.2.4 Nachteile des CF

Die wesentlichen Nachteile des CF sind das *Kaltstart-Problem*, das Problem der *"Spärlichkeit" (sparsity)* und der *Lemming-Effekt*. Diese sollen im Folgenden kurz beschrieben werden.

Kaltstart-Problem

Da Empfehlungen offensichtlich nur auf dem Verhalten anderer Benutzer ausgesprochen werden können, muss ein CF Verfahren zunächst eine "kritische Menge" an Benutzeraktionen erfasst haben, bevor es Empfehlungen aussprechen kann. Das gilt nicht nur für die CF-Matrix an sich (und damit für neue Objekte), sondern insbesondere für jeden neuen Benutzer. Ohne dass dieser ausreichend viele eigene Aktionen (implizite oder explizite Bewertung von Informationsobjekten) durchgeführt hat, kann ein CF System ihm keine Empfehlungen geben.

Spärlichkeit (sparsity)

Aufgrund der in CF Umgebungen häufig sehr großen Anzahl von Objekten sind oft 98% bis 99% der Objekte nicht mit einer Bewertung durch den Benutzer versehen [CLA1999]. Betrifft dies im Mittel über alle Benutzer mehr als 99,5% der Objekte, so fällt die Empfehlungsgüte eines CF Verfahrens drastisch ab [MEL2002]. Anschaulich sinkt mit zunehmender "Spärlichkeit" der Bewertungen die Wahrscheinlichkeit Benutzer zu finden, die gleiche Objekte gleich oder ähnlich bewertet haben.

Lemming-Effekt

Eines der wohl bekanntesten Portale mit CF Verfahren dürfte der Internet-Shop "Amazon" sein. Hier funktioniert das CF offenbar – vorausgesetzt man hat bereits ein paar Bücher angesehen oder gekauft und daher das Kaltstart-Problem beseitigt – recht gut. Wie bei allen Anwendungsfällen, die eine Bewertung eines Informationsobjektes mit einem pekuniären Aspekt verbinden, besteht auch hier eine natürliche Hürde, die empfohlenen Objekte anzunehmen (zu erwerben) und damit selbst positiv zu bewerten.

Abbildung 13: Auch Empfehlungen bei Shops wie "Amazon" sind nicht komplett vor dem Lemming-Effekt gefeit.

Allerdings kann man bei sehr populären Produkten, wie beispielsweise im Jahr 2006 bei den Büchern von Dan Brown, selbst hier den Lemming-Effekt beobachten. Die Bücher von Dan Brown werden zu sehr vielen anderen Büchern empfohlen, obwohl thematisch kaum Zusammenhänge erkennbar sind. Der Grund dafür sind extrem viele Benutzer, die Dan Browns Bücher gekauft haben. Entfällt die "Kaufhürde", rufen Benutzer die Ihnen empfohlenen Informationsobjekte ungleich öfter auf, so dass diese - besonders bei impliziter Bewertung durch den Aufruf - wiederum anderen Benutzern empfohlen werden und so fort. Dadurch werden empfohlene Objekte

immer weiter gestärkt. Neue Objekte haben kaum eine Chance in die Liste der empfohlenen Informationsobjekte aufgenommen zu werden.

5.2.3 Hybrid-Systeme

Hybrid-Systeme die CBF und CF verbinden, versuchen die Vorteile beider Ansätze zu kombinieren. Die Varianz der Mischformen ist sehr breit und wird im Folgenden einer systematischen Betrachtung unterzogen werden.

5.3 Differenzierungsmerkmale von Empfehlungssystemen

Empfehlungssysteme nur durch die Verwendung von CBF, CF oder einem Hybrid-Ansatz zu klassifizieren, erscheint aufgrund der großen Bandbreite der verwendeten Techniken innerhalb der jeweiligen Klasse, als zu grob. Die im Folgenden beschriebenen Merkmale werden in logisch zusammenhängenden Gruppen aufgeführt. Die Merkmale sollen einer sinnvollen Einordnung verschiedener Empfehlungssysteme im Rahmen diese Arbeit dienen.

5.3.1 Informationsobjekt

5.3.1.1 Merkmal: Art

Die *Art* beschreibt die Zugehörigkeit der behandelten Informationsobjekte zu Objektgruppen der realen Welt (Text, Filme, Bücher et cetera).

5.3.2 Content-Charakteristika Ermittlung

Bei der Verarbeitung von Informationsobjekten spielt die Ermittlung der bestimmenden Charakteristika ("Kerninformation") eine wichtige Rolle. Da die betrachteten Verfahren alle auf Textobjekten arbeiten, sollen im Folgenden die wichtigsten Verfahren zur Ermittlung der Content-Charakteristika für Texte vorgestellt werden. Konkret werden die Charakteristika von Texten meist durch einen Wortvektor, Wortmengen oder Textfragmente repräsentiert.

5.3.2.1 Merkmal: Manuelle Metadaten

Dieses Merkmal ist gegeben, wenn manuell erfasste *Metadaten* des Informationsobjektes, wie in 5.2.1.1 (Relationale und autonome Metadaten) beschrieben, verwendet werden.

5.3.2.2 Merkmal: Textstruktur

Wenn ein Verfahren für die Ermittlung der Charakteristika eines Informationsobjektes die *Textstruktur* (Textanfang, Textende, Absätze et cetera) verwendet, ist dieses Merkmal gegeben.

5.3.2.3 Merkmal: NLP

Das *NLP* (natural language processing) nutzt zur Extraktion der Charakteristika eines Textes formale Eigenschaften einer Sprache (Morphologie, Grammatik et cetera). Dadurch lassen sich beispielsweise Funktionsgruppen von Worten (Verben et cetera) bilden. Ein Verfahren, das NLP Techniken verwendet, besitzt dieses Merkmal.

5.3.2.4 Merkmal: Basiskorpus

Ein weiteres potenzielles Hilfsmittel bei der Textanalyse zur Ermittlung der Charakteristika ist die Verwendung eines Referenzkorpus oder *Basiskorpus*, der in der Regel aus einer großen Zahl repräsentativer Texte einer Sprache oder einer Sprachdomäne (Fachsprache et cetera) besteht. Ein Basiskorpus liefert verwertbare statistische Daten über die betreffende Sprache oder Sprachdomäne.

5.3.2.5 Merkmal: TF-IDF-Derivat

Das bekannteste Verfahren zur Ermittlung "bestimmender Worte" eines Textes ist das TF-IDF-Verfahren [LUH1958],[CLE1967], [SPA1972], [WHU1981]. Es verwendet statistische Daten im eigenen Korpus (IDF) und im Text (TF), um die bedeutsamen Worte eines Textes zu ermitteln. Das Konzept beruht auf zwei

Annahmen zu Texten. Zum einen sind Worte für einen Text vermeintlich umso bedeutsamer, je häufiger sie in diesem vorkommen (term frequency *TF*). Zum anderen sind Worte, die in sehr vielen Texten vorkommen, für den einzelnen Text und insbesondere dessen Charakterisierung weniger bedeutsam (inverse document frequency *IDF*). Formal gilt für ein Wort W:

$W = TF * IDF$ mit $TF = N_T(W)$ und $IDF = \log \frac{|C|}{I_C(W)}$

mit

$N_T(W)$ = Anzahl der Instanzen ("Frequenz") von Wort W im Text T

$I_C(W)$ =Anzahl der Texte in der Textbasis $C = \{T_1, \ldots, T_n\}$ mit mindestens einer Instanz von W

$|C|$ =Anzahl der Texte der Textbasis C

Das TF-IDF-Verfahren kommt in unterschiedlichen Derivaten zum Einsatz, die alle unter diesem Merkmal zusammengefasst werden sollen.

5.3.2.6 Merkmal: Heuristik

Verfahren, welche die bestimmenden Worte eines Textes durch eine Heuristik ermitteln, bezeichnet man als heuristisch. Eine Heuristik ist eine Annäherung oder eine "Daumenregel" für die Lösung eines Problems [GIG1999].

5.3.3 Profilbildung

5.3.3.1 Merkmal: Manuelle Pflege

Wenn der Benutzer sein Profil explizit und für ihn erkennbar definiert (beispielsweise durch die Aufnahme von Informationsobjekten oder die Auswahl gewünschter Informationskategorien) besitzt ein Verfahren dieses Merkmal.

5.3.3.2 Merkmal: Bewertung von Objekten

Wenn der Benutzer sein Profil implizit durch die Bewertung von Informationsobjekten bildet, besitzt das Verfahren dieses Merkmal.

5.3.3.3 Merkmal: Benutzerverhalten

Wenn ein Verfahren das Benutzerverhalten (abgerufene Informationsobjekte, Lesedauer et cetera) protokolliert und daraus ein implizites Profil bildet, trägt es dieses Merkmal..

5.3.3.4 Merkmal: Flüchtiges Profil

Ist das Profil nur für die Dauer einer Session existent, handelt es sich um ein flüchtiges Profil. Dies ist in der Regel bei Personalisierung ohne Anmeldung durch den Benutzer der Fall.

5.3.3.5 Merkmal: Persistentes Profil

Ein persistentes Profil bleibt auf Dauer verfügbar. Es erfordert daher zwingend die Identifikation des Benutzers. Dazu kann beispielsweise eine explizite Authentifikation oder ein Browser-Cookie verwendet werden.

5.3.4 Berechnung der Distanz

Die Berechnung der Distanz zwischen zwei Informationsobjekten oder einem Informationsobjekt und einem Benutzerprofil kann auf vielfältige Weise erfolgen. Die wichtigsten Konzepte sollen im Folgenden aufgeführt werden. Diese können potenziell auch für die Distanzberechnung zwischen Benutzern (CF) eingesetzt werden. Im Folgenden ist daher abstrakt von *Objekten* die Rede.

5.3.4.1 Merkmal: Asymmetrisch

Sei d ein Distanzmaß und a,b Objekte und sei $d(a,b)$ die Distanz von a zu b. Gibt es ein (a,b) mit $d(a,b) \neq d(b,a)$, so ist d *asymmetrisch*. Diese Eigenschaft ist für die Qualität von CBF-basierten

Empfehlungen sehr bedeutsam, da die semantische Beziehung von Informationsobjekten selten symmetrisch ist. Daher können nur asymmetrische Verfahren die "realen" Beziehungen zumindest potenziell adäquat abbilden.

5.3.4.2 Merkmal: Symmetrisch

Sei `d` ein Distanzmaß und `a,b` beliebige Informationsobjekte und gelte `d(a,b)=` Distanz von a zu b. Gilt `d(a,b)= d(b,a)` für alle `(a,b)`, so ist `d` symmetrisch.

5.3.4.3 Merkmal: Vektorbasiert

Wird die Distanz auf Basis von Vektoren gewonnen, welche die Objekte repräsentieren, handelt es sich um ein vektorbasiertes Verfahren. Bei *vektorbasierten* Verfahren werden die Objekte in der Regel durch Wortvektoren repräsentiert.

5.3.4.4 Merkmal: Standardkonzepte

Bei den *Standardkonzepten* sind solche mit direkter (beispielsweise Kosinus Ähnlichkeitsmaß) und mit indirekter Bestimmung der Distanz durch Klassifikation zu unterscheiden. Bei der Klassifikation liegt ein symmetrisches Distanzmaß vor, wenn ein Objekt genau einer Klasse zugeordnet wird. Kann ein Objekt mehreren Klassen zugeordnet werden, liegt ein asymmetrisches Distanzmaß vor. Im Folgenden werden die im Rahmen der Einordnung der Empfehlungsverfahren verwendeten Konzepte zur Distanzermittlung vorgestellt.

COS – Kosinus Ähnlichkeitsmaß

Das Kosinus Ähnlichkeitsmaß (cosine similarity) bestimmt die Ähnlichkeit `Sim` zweier n-dimensionaler Vektoren $v=(v_1,\ldots,v_n)$ und $w=(w_1,\ldots,W_n)$ wie folgt durch deren Kosinus:

$$Sim_{cos\theta}(v,w) = \frac{v \bullet w}{|v| * |w|} = \frac{v_1 * w_1 + \ldots + v_n * w_n}{\sqrt{v_1^2 + \ldots + v_n^2} * \sqrt{w_1^2 + \ldots + w_n^2}}$$

wobei v•w das Skalarprodukt und |v|*|w| das Produkt der Beträge (Längen) der Vektoren ist. Je ähnlicher Vektoren sind, desto geringer ist der Winkel zwischen Ihnen und desto mehr tendiert das Kosinus Ähnlichkeitsmaß $Sim_{cos\theta}(v,w)$ gegen den Wert 1.

MDL – Minimum Description Length

Beim MDL Verfahren (Minimum Description Length [RIS1978]) wird ein Dokument `D` der Klasse K_1 eher zugeordnet als der Klasse K_2, wenn die binärcodierte Darstellung der Klasse K_1 erweitert um `D` weniger Speicherplatz benötigt als die binärcodierte Darstellung der Klasse K_2 erweitert um `D`.

MDL basiert auf der Annahme, dass das kompakteste Modell auch das optimale ist. Es folgt somit dem Prinzip *Pluralitas non est ponenda sine necessitate* [ENC2005] von "Ockhams Razor" (William of Ockham, 1285-1349), das besagt, dass "Große Mengen ("Vielheiten) nicht ohne Grund angenommen werden sollten".

Formal kann der MDL Ansatz über das Bayes´sche Theorem hergeleitet werden. Gesucht ist die Klasse K_i, welche die Wahrscheinlichkeit $p(K_i|D)$ für K_i bei gegebenem D maximiert. Mit dem Bayes´schen Theorem kann das in

$$\max_{K_i} \frac{p(D|K_i) * p(K_i)}{p(D)}$$

umgeformt werden. Da `P(D)` unabhängig von K_i ist, ist es für die Maximierung des Terms über K_i irrelevant. Gesucht ist also

$$\max_{K_i} p(D|K_i) * p(K_i)$$

Das ist wegen `p(x)<=1` äquivalent zu

$$\min_{K_i}\left(-\log\left(p(D|K_i) * p(K_i)\right)\right)$$

$$\Leftrightarrow \min_{K_i}\left(\left(-\log p(D|K_i)\right) + \left(-\log p(K_i)\right)\right) \Leftrightarrow \min_{K_i}\left(-\log p(D|K_i) - \log p(K_i)\right)$$

Nun ist aber $-\log_2(p(X))$ nach Shannon [SHA1948] gleich der Bitzahl eines optimalen Binärcodes für X. Daher sucht man bei MDL die wahrscheinlichste Klasse K_i bei gegebenem Dokument D, indem man die Klasse K_i mit der kürzesten Bitcodierung sucht. Die Bitcodierung umfasst dabei zum einen die Klasse selbst, zum anderen die Klasse zuzüglich des Dokumentes.

Um MDL konkret anwenden zu können, muss die Bitcodierung konkretisiert werden. Exemplarisch sei hier die Verfahrensweise von NewsWeeder vorgestellt:

D = das zu klassifizierende Dokument

W_D = binärer Wortvektor des Dokumentes D

L_D= die Anzahl der Nicht-Null-Komponenten in W_D =Anzahl unterschiedlicher Worte in D

T= die Trainingsdaten mit denen die Klassen aufgebaut wurden; dies ist ein Wortvektor der für jedes Wort die Anzahl der Dokumente speichert, die dieses Wort enthalten

K_i= die Klasse besteht aus "gelernten" Dokumenten der Trainingsdaten; dies ist analog zu T ein Wortvektor – allerdings beschränkt auf die Dokumente aus K_i.

Gesucht ist dann die Klasse K_i, welche die Binärcodierung der Klasse und des Dokumentes minimiert [LAN1995]:

$$\min_{K_i}\left\{-\log\left(p\left(W_D\middle|K_i,L_D,T\right)\right)-\log\left(p\left(K_i\middle|L_D,T\right)\right)\right\}$$

NBK – Naiver Bayes-Klassifikator

Der Naive Bayes-Klassifikator (NBK) ermöglicht es ein Dokument D zu einer Klasse zuzuordnen. Es wird für den Wortvektor $W_D=(w_1,\ldots,w_n)$ des Dokumentes die Klasse K_i mit der größten bedingten Wahrscheinlichkeit für W_D gesucht:

$\max_i\ P(K_i|W_D)$

Mit dem Bayes´sche Theorem wird dies zu:

$$\max_{K_i}\frac{p\left(W_D\middle|K_i\right)*p\left(K_i\right)}{p\left(W_D\right)}$$

Da $p(W_D)$ für alle Klassen K_i identisch ist und damit für die Maximierung über die Klassen irrelevant, folgt:

$$\max_{K_i}p\left(W_D\middle|K_i\right)*p\left(K_i\right)$$

Die Naive Bayes Klassifikation trifft die Annahme, dass die Attribute von W_D (Worte) unabhängig voneinander auftreten. Dies ist in der Praxis natürlich nicht der Fall, da beispielsweise die Worte "Doktor" und "Krankenschwester" signifikant häufiger gemeinsam auftreten als die Worte "Pinguin" und "Wüste". Dennoch wird diese Annahme getroffen, was mit w_j als der j-ten Komponente (Wort) des Wortvektors W_D zu folgender Umformung führt:

$$\max_{K_i}p\left(w_1\middle|K_i\right)*\ldots*p\left(w_n\middle|K_i\right)*p\left(K_i\right)$$

Die Wahrscheinlichkeiten $p(w_j|K_i)$ und $p(K_i)$ werden auf Basis statistischer Daten der Lernmengen wie folgt abgeleitet (geschätzt). Die Wahrscheinlichkeit $p(K_i)$ wird aus der Anzahl der Dokumente einer Klasse in Relation zur Summe der Dokumente aller Klassen berechnet:

$$p(K_i)=\frac{\left|\left\{D\middle|D\in K_i\right\}\right|}{\sum_k\left|\left\{D\middle|D\in K_k\right\}\right|}$$

Neben dieser a-priori Wahrscheinlichkeit für die Klassen gilt für die Wahrscheinlichkeit der Zugehörigkeit des Wortes w_j zur Klasse K_i mit $W_D[w_j]$ als j-ter Komponente des Wortvektors W_D des Dokumentes D:

$$p\left(w_j\middle|K_i\right)=\frac{\left|\left\{D\middle|D\in K_i\wedge W_D\left[w_j\right]>0\right\}\right|}{\left|\left\{D\middle|D\in K_i\right\}\right|}$$

Die bedingte Wahrscheinlichkeit wird also aus der Anzahl der Dokumente einer Klasse, die das Wort enthalten, in Relation zu der Anzahl der Dokumente der Klasse insgesamt berechnet.

Bayes´sches Netz

Ein Bayes´sches Netz wird von keinem der untersuchten Empfehlungsverfahren verwendet, soll aber als Weiterentwicklung von NBK dennoch kurz vorgestellt werden.

Ein Bayes´sches Netz $B=<V,E,p>$ ist ein gerichteter azyklischer Graph (DAG) mit V als Menge der Knoten und E als Menge der Kanten sowie einer bedingten Wahrscheinlichkeit für jeden Knoten N. Jeder Knoten N stellt eine Zufallsvariable dar. Die Verbindung von Knoten N_1 zu Knoten N_2 steht für die bedingte Wahrscheinlichkeit $P(N_2|N_1)$. Bei einer Klassifikation mittels eines Bayes´schen Netzes stellen die inneren Knoten *Worte* und die Blätter die *Klassen* dar.

Der Aufbau des Bayes´schen Netzes muss aus den Trainingsdaten abgeleitet werden. Er erfolgt in zwei Schritten. Zunächst wird die Struktur aufgebaut, dann werden die Parameter gesetzt [CHE1999][CHE1997].

Der Strukturaufbau besteht aus drei Teilschritten. In der Draft-Phase wird die Mutual Information (siehe unten) aller inneren Knoten (Worte) berechnet und aufgrund der so ermittelten Ähnlichkeiten der Worte ein Graph erzeugt.

In der zweiten Phase (Thickening) werden Kanten zwischen nicht *d-separierbaren* Knoten eingezogen. Zwei Knoten N_1 und N_2 sind d-separierbar, wenn auf allen Pfaden zwischen N_1 und N_2 ein Knoten Z existiert, so dass die Wahrscheinlichkeit von Z bekannt ist und die Verbindung von Z zu N_1 und N_2 seriell oder divergierend, oder so dass Z und dessen Nachfolger unbekannt sind und Z von N_1 und N_2 zu Z konvergierend ist.

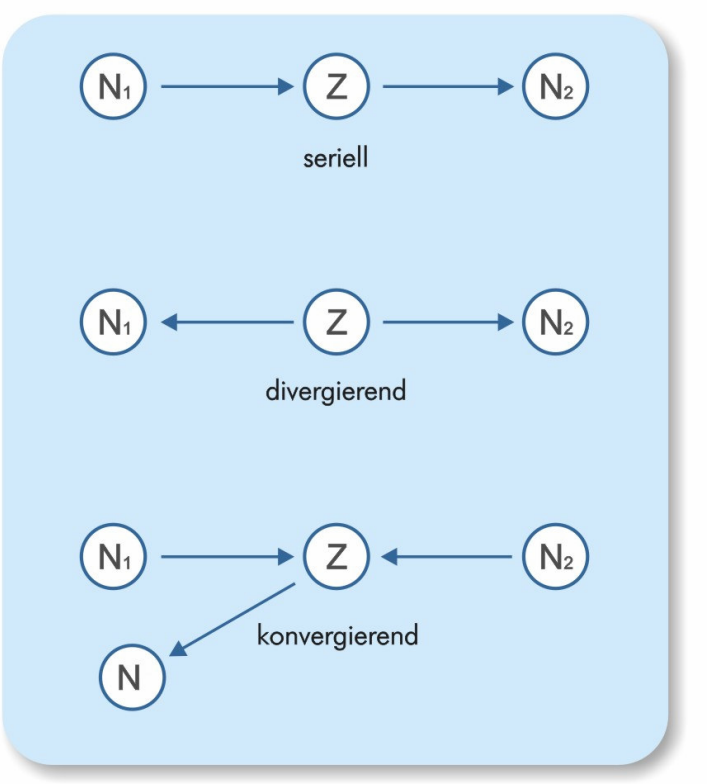


Abbildung 14: Kriterien für d-separierte Knoten

Im folgenden Graphen sind beispielsweise die Knotenpaare (1,2) und (5,6) d-separiert, wenn die Wahrscheinlichkeit für die Knoten "2" und "5" gegeben ist.

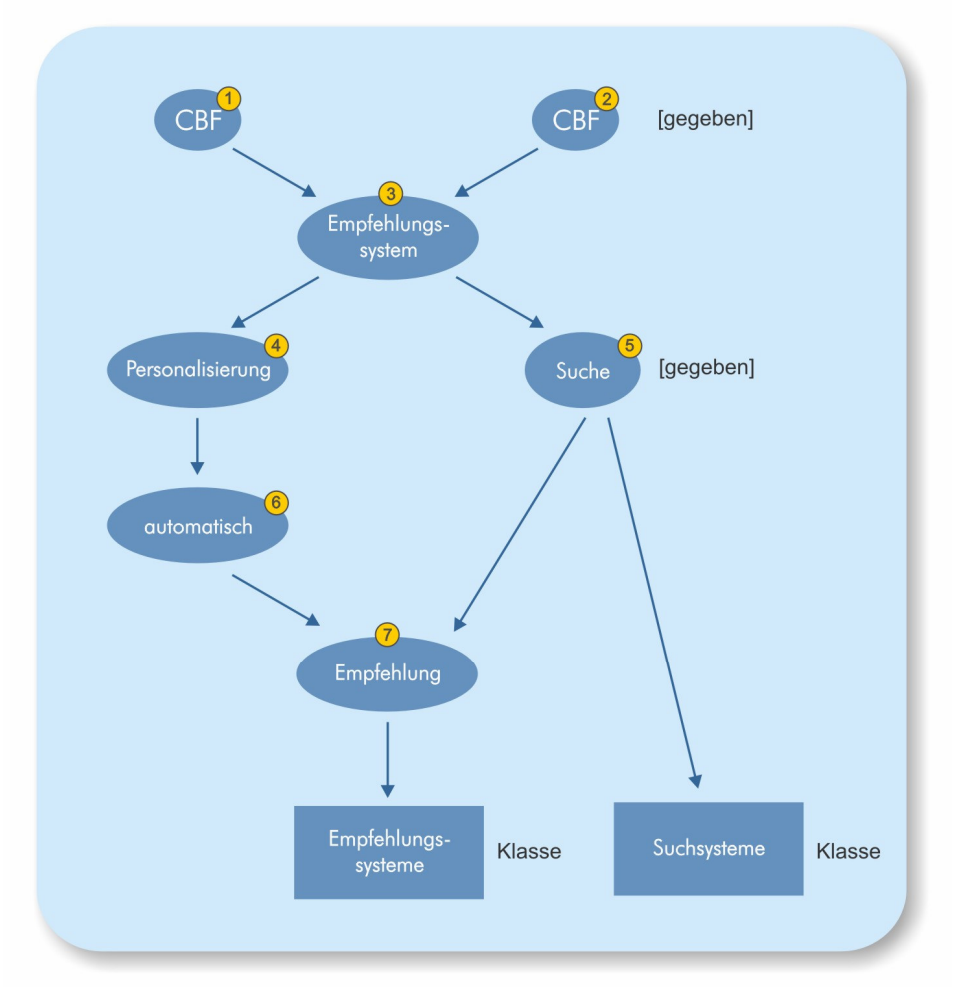


Abbildung 15: Ein Bayes'sches Netz im Aufbau

In der dritten Phase (Thinning) werden schließlich alle Kanten entfernt, deren Knoten d-separierbar sind. Die Parameter ergeben sich bei der Textklassifikation aus den Testdaten. Hier sind sowohl die a-priori als auch die bedingten Wahrscheinlichkeiten der Worte in Form der relativen Häufigkeiten beziehungsweise der Kookurrenzen vorhanden. Deren Berechnung ist aber ein recht aufwändiges Unterfangen.

MI – Mutual Information

Beim *Mutual Information* Ansatz wird die Beziehung zwischen zwei Wortvektoren aus dem Grad des gemeinsamen Vorkommens (wechselseitige Information) der einzelnen Worte abgeleitet [CHU1989].

Wenn zwei Worte w_i und w_j die Wahrscheinlichkeiten $p(w_i)$ und $p(w_j)$ haben und $p_F(w_i, w_j)$ die Wahrscheinlichkeit für deren gemeinsames Auftreten in einem Textfenster F ist, ist $MI(w_i, w_j)$ wie folgt definiert:

$$MI(w_i, w_j) = \begin{cases} \log_2 \dfrac{p_F(w_i, w_j)}{p(w_i) * p(w_j)} \; für (w_i \neq w_j) \\ 1, sonst \end{cases}$$

Es wird also die Wahrscheinlichkeit des gemeinsamen Auftretens mit der des unabhängigen Auftretens verglichen. Bei einer signifikanten Beziehung von w_i und w_j wird $MI(w_i, w_j)$ deutlich größer als Null. Umgekehrt haben bei $MI(w_i, w_j) \approx 0$ die Worte wenig gemein.

Zu Berechnung von $p_F(w_i, w_j)$ wird F konkret festgelegt. In [CHU1989] wird es beispielsweise auf fünf Worte (Token) gesetzt. Sei außerdem W die Menge aller Worte aller Dokumente in der Textbasis C und $|W|$ die Anzahl der Worte in C sowie $|w_i|$ die Anzahl des Vorkommens (Instanzen) von w_i in C.

Dann gilt:

$$p(w_i) = \frac{|w_i|}{|W|}$$

und mit `F(wi,wj)` als der Anzahl des gemeinsamen Vorkommens (Kookurrenz) von `wi` und `wj` in einem Textfenster von `F` Worten gilt:

$$p(w_i, w_j) = F(w_i, w_j)$$

Dieses `F` für alle Worte zu bestimmen ist aufgrund der Permutationsmöglichkeiten offensichtlich ein aufwändiges Unterfangen. Daher wird oft auf Werte zurückgegriffen, die auf Basis verhältnismäßig großer Korpora ermittelt wurden [MEI1978], [LEI2006].

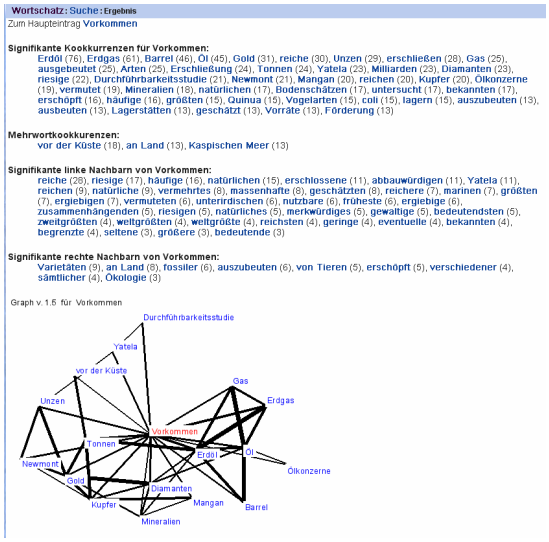


Abbildung 16: Abfrage der Kookurrenzen zum Wort "Vorkommen" auf http://wortschatz.uni-leipzig.de/index_js.html (Wortschatz Lexikon der Universität Leipzig; 2006-01-04)

Für zwei Dokumente `D1` und `D2` mit deren Wortvektoren W_{D1} und W_{D2} kann die Ähnlichkeit MI_{DOC} dann wie folgt mit der `MI` über alle Permutationen der Worte aus `D1` und `D2` berechnet werden:

$$MI_{doc}(W_{D1}, W_{D2}) = \sum_{(w_i, w_j), w_i \in W_{D1}, w_j \in W_{D2}} MI(w_i, w_j)$$

Je größer der resultierende Wert, desto ähnlicher sind sich vermeintlich die beiden Texte.

CBR – Case based reasoning

Das case based reasoning bestimmt in der Vergangenheit aufgetretene Fälle, durch Ähnlichkeit zum aktuellen Fall. Es kommt nur bei INFOS [MOC1996] zum Einsatz und wird dort (siehe 5.4.2.29) genauer beschrieben.

PC - Korrelationskoeffizient

Der Korrelationskoeffizient (*Pearson Correlation* [BAS2004]) ist eine Ähnlichkeitsfunktion. Für zwei Vektoren `v` und `w` mit `v=(v1,v2,…,vn)` und `w=(w1,w2,…,wm)` gilt:

$$\frac{\sum_{i=1\ldots n}\left((v_i - \overline{v})^2 * (w_i - \overline{w})^2\right)}{\sqrt{\sum_{i=1\ldots n}(v_i - \overline{v})^2 * \sum_{i=1\ldots n}(w_i - \overline{w})^2}}$$

Dabei steht $\overline{v}$ beziehungsweise $\overline{w}$ für den Durchschnitt der Komponenten von `v` beziehungsweise `w`. Ein Korrelationskoeffizient nahe 1 steht für eine starke und ein Wert nahe 0 für eine schwache Ähnlichkeit.

GHC

Das global hill climbing (GHC) Modell beruht auf einem vereinfachten TF-IDF-Derivat und kommt nur bei INFOS [MOC1996] zum Einsatz, wo (siehe 5.4.2.29) es auch genauer beschrieben wird.

OWS

Das Okapi weighting scheme (OWS) stellt eine Heuristik dar und gestaltet sich wie folgt:

$$\sum_{T \in Q} w_i * \frac{(k_1+1)*tf}{K+tf} * \frac{(k_3+1)*qtf}{k_3+qtf}$$

mit `Q` als Anfrage, welche die Worte `T` enthält und

$$K = k_1 * \left((1-b) + b * \frac{dl}{avgdl} \right)$$

k_1=1,2

k_3=zwischen 0 und 1000 wählbar

b=0,75

tf=Anzahl der Instanzen von T in einem Dokument

qtf=Anzahl der Instanzen von T in allen Texten eines *topics*

dl=Dokumentlänge

avgdl=Durchschnittliche Dokumentlänge

w_i=Robertson-Sparck-Jones (RSJ) Gewicht mit

$$w_i = \log \frac{\left(\frac{r_i + 0{,}5}{R - r_i + 0{,}5} \right)}{\left(\frac{n_i - r_i + 0{,}5}{N - n_i - R + r_i + 0{,}5} \right)}$$

mit

N = Größe der Textbasis C

n_i = Anzahl der Texte in C die das Wort T_i enthalten

R = bekannte Dokumente die zu einem *topic* relevant sind

r_i = Anzahl der als relevant ausgezeichneten Texte die das Wort T_i enthalten

OK - Overlap Koeffizient

Der Overlap Koeffizient ist ein Ähnlichkeitsmaß. Für zwei Vektoren v und w mit $v=(v_1, v_2, \ldots, v_n)$ und $w=(w_1, w_2, \ldots, w_m)$ sowie die Mengen $V_M=\{v_1, v_2, \ldots, v_n\}$ und $W_M=\{w_1, w_2, \ldots, w_m\}$ der Vektorkomponenten (in der Regel Worte) gilt:

$$OK(v,w) = \frac{2*|V_M \cap W_M|}{\min(|V_M|, |W_M|)}$$

Es wird die Größe der gemeinsamen Wortmenge beider Vektoren durch die kleinere von beiden normiert, so dass große Wortmengen nur normalen Einfluss haben.

ID3

Das ID3 [QUI1986] Verfahren dient der Klassifikation von Informationsobjekten. Dazu wird ein Entscheidungsbaum anhand einer Trainingsmenge aufgebaut. Im Entscheidungsbaum stehen die Blätter für die Klassen, die inneren Knoten stellen Attribute dar und die Kanten tragen einen Attributwert ihres Ausgangsknotens:

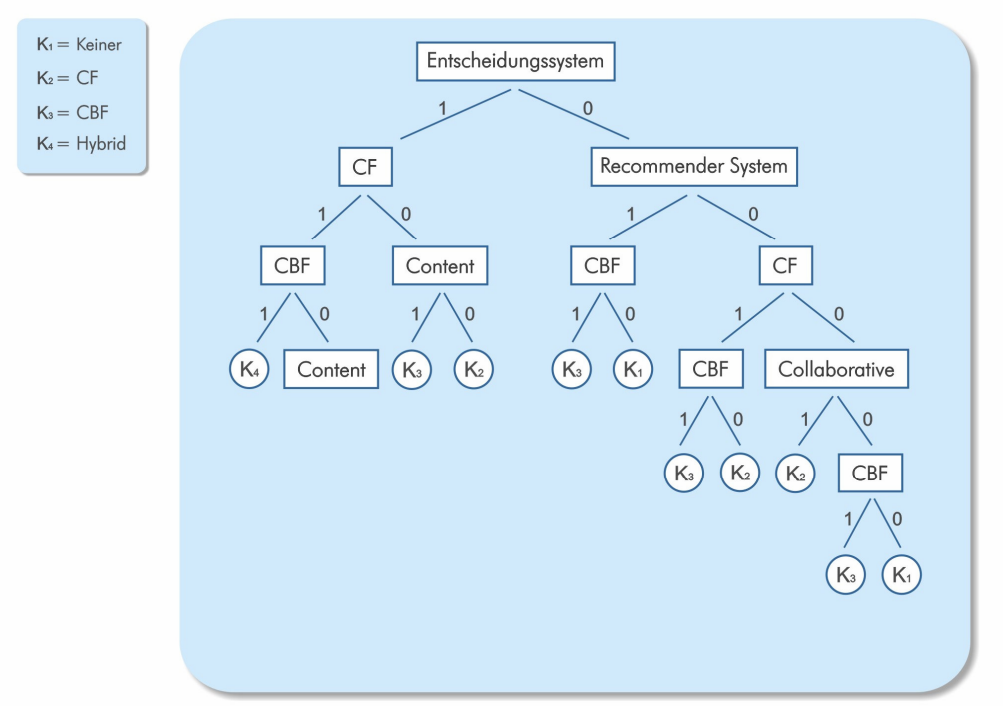


Abbildung 17: Ein ID3 Entscheidungsbaum besteht aus Klassen (Blättern), Attributen (inneren Knoten) und Attributwerten (Kanten). Aus dem Entscheidungsbaum lassen sich Regeln für die Klassenzugehörigkeit ableiten.

Die Attribute sind im Kontext von Entscheidungssystemen in der Regel Worte, die entweder (wie in der oben stehenden Abbildung) binär oder gewichtet sind. Wichtig ist allerdings, dass die Anzahl der Werte für ein Attribut klein bleiben muss, wenn ID3 sinnvoll eingesetzt werden soll, da die maximale Ausprägung eines Attributes mit dem Grad des Baumes übereinstimmt. Aufgrund des Entscheidungsbaumes werden neue Informationsobjekte (Texte) dann einer Klasse zugeordnet.

Der Aufbau eines Entscheidungsbaumes erfolgt mit folgendem Algorithmus auf Basis von Traningsdaten T (Texte) die manuell den Klassen K_i zugeordnet wurden:

```
ID3 (T,Ki)
      wenn T=Ki für eine Klasse Ki,
            dann erzeuge ein Blatt "Ki"
            sonst wenn T=∅
                  dann erzeuge ein Blatt "∅"
                  sonst
                        wähle attribut(T,Aj)
                        erzeuge Knoten "Aj"
                        für jeden Wert WAjk von Aj mit k=1…q
                              erzeuge Kante "WAjk"
                              setze T:=T∩{t|t∈T ∧ t hat Wert WAJk}
                              rufe ID3(T,Ki) auf
                              füge B an Kante "WAJk" ein
      gebe den erzeugten Baum "B" zurück
```

Algorithmus 1: ID3

Die Funktion "`wähle attribut(T,A_j)`" wählt das Attribut mit dem maximalen Informationsgewinn und ist wie folgt definiert; sind keine Attribute mehr vorhanden, gilt: "`wähle die häufigste Klasse`".

Der Informationsgewinn `G(A)` eines Attributes `A` wird aus der Differenz des Informationsgehaltes (oder der *Entropie*) `I(T)` ohne das Attribut `A` und dem Informationsgehalt `I(T|A)` mit dem Attribut `A` (*bedingte Entropie*) bestimmt. Der Informationsgewinn steht also dafür, wie gut man eine Klassifikation eines Textes mit und ohne dem Attribut vornehmen kann:

`G(A)=I(T)-I(T|A)`

Der Informationsgehalt einer Menge `T` von Texten wird durch die Entropie

$$I(T) = -\sum_{i=1\ldots n}\left(p(K_i) * \log_2\, p(K_i)\right)$$

bestimmt [MAC2005],[SHA1948]. Konkret steht `p(K_i)` hier für die Wahrscheinlichkeit von Klasse `K_i` der Trainingsdaten T beziehungsweise einer Teilmenge davon.

Der Informationsgehalt der Menge `T` mit Attribut `A` als Wurzel ist durch die bedingte Entropie [MAC2005; S. 138]

$$I(T|A) = \sum_{i=1\ldots s} p(A = WAJ_i) * I(\{t|t \in T \wedge A = WAJ_i\})$$

definiert. Mit `P(A=WAJ_i)` als Wahrscheinlichkeit, dass A den Wert `WAJ_i` annimmt und `{t|t∈T∧A=WAJ_i}` ist die Teilmenge der Trainingsdaten `T`, die Attribut `A` mit dem Wert `WAJ_i` (dem Wort) besitzt.

Insgesamt ergibt sich

$$\max_{A_j}\left(I(T) - I(T|A_j)\right)$$

$$= \max_{A_j}\left(-\sum_{i=1\ldots n}\left(p(K_i) * \log_2\, p(K_i)\right) - \sum_{i=1\ldots s} p(A = WAJ_i) * I(\{t|t \in T \wedge A = WAJ_i\})\right)$$

Die Wahrscheinlichkeiten werden algorithmisch durch Häufigkeiten ersetzt:

$$p(K_i) = \frac{|\{t|t \in K_i\}|}{|\{t|t \in T\}|}$$

und

$$p(A = WAJ_i) = \frac{|\{t|t \in T \wedge WAJ_i \in t\}|}{|\{t|t \in T\}|}$$

mit `|M|` als der Anzahl der Elemente der Menge `M` und "`WAJ_i∈t`" als Bedingung, dass ein Text `t` den Attributwert `WAJ_i` (das Wort) enthält.

KMC – K-Means-Clustering

Das Clustering teilt Elemente in Gruppen auf, unterscheidet sich von der Klassifikation aber dadurch, dass die Gruppen beim Clustering zunächst unbekannt sind und automatisch gebildet werden. Man unterscheidet hierarchisches und nicht-hierarchisches Clustering. Ersteres startet mit einem großen Cluster und spaltet dann in jedem Lauf neue Cluster ab. Letzteres startet mit meist willkürlich gewählten Clustern und bildet diese in jedem Lauf mehr und mehr aus. Weitere Spielarten sind das harte und das weiche Clustering. Während Ersteres jedes Element genau einer Gruppe zuordnet, erlaubt Letzteres die Zuordnung eines Elementes zu mehreren Gruppen.

Übertragen auf Vektoren stellt das Clustering anschaulich ein geometrisches Verfahren im n-dimensionalen Raum dar. Ein bekannter (harter) Algorithmus zur Clusterbildung ist "K-Means" der nach willkürlicher Wahl repräsentativer "Punkte" im Vektorraum die Vektoren den nächsten Punkten zuordnet und dann eine iterative Verbesserung der Punkte durchführt:

Sei X eine endliche Teilmenge eines metrischen Raumes mit der Distanzfunktion $d(x,y)$. Sei f eine Funktion zur Ermittlung des "Schwerpunktes" von n Vektoren $v_1,\ldots,v_n$. Im einfachsten Fall ist das der Mittelwertvektor der Vektoren $v_1,\ldots,v_n$.

Wähle k initiale Punkte $p_1,\ldots,p_k$

setze $p_{korrektur}$=Toleranz (Abbruch bei Erreichen der Toleranz in Summe für alle Cluster)

solange pkorrektur>=Toleranz

  setze $p_{korrektur}=0$ (Initialisierung)

  für Cluster c_i mit $i=1,\ldots,k$

    $c_i=\{v_j \mid \forall p_{l=1\ldots k}: d(v_j,p_i)<=d(v_j,p_l)\}$

    $p_{alt}=p_i$

    $p_i=f(c_i)$

    $p_{korrektur}=p_{korrektur}+(p_i-p_{alt})$ (Korrekturfaktor akkumulieren)

Algorithmus 2: K-Means Clustering

NN – Nearest Neighbours

Beim Nearest Neighbours Verfahren handelt es sich um ein Verfahren zur Ermittlung der k nächsten Nachbarn eines Elementes. Meist beruht es auf der Optimierung eines anderen Distanzmaßes. Statt in $O(n)$ Laufzeit aus n Elementen die k nächsten Nachbarn zu selektieren, wird beispielsweise durch Klassifikation ein optimiertes iteratives Verfahren angewendet. Formal sind die k-nächsten Nachbarn eines Elementes a wie folgt definiert:

Sei X eine endliche Teilmenge eines metrischen Raumes mit der Distanzfunktion $d(x,y)$. Die Punkte $b_i \in X$ mit $i=1,\ldots,k$ sind die k-nächsten-Nachbarn von a, falls gilt:

$0<=d(a,b_i)<=d(a,b_{i+1})$ für $i=1,\ldots,k$

und $d(a,b_k)<=d(a,c)$

für alle $c \in X\setminus\{b_1,\ldots,b_k\}$.

Bei Empfehlungssystemen sind die k-nächsten-Nachbarn die Informationsobjekte oder Benutzer mit dem geringsten Abstand zu einem konkreten Informationsobjekt beziehungsweise Benutzer.

5.3.4.5 Merkmal: Regelbasiert

Ein regelbasiertes Verfahren leitet aus einer gegebenen Variable X eine Variable Y ab. Dabei sind X und Y meist Tupel. Wenn X beispielsweise die bisher durch einen Benutzer aufgerufenen Informationsobjekte darstellt, könnte Y ein zu empfehlendes Informationsobjekt sein, dass durch die Regel $X \rightarrow Y$ aus X abgeleitet wird.

5.3.4.6 Merkmal: Heuristik

Alle Verfahren die eine eigene Heuristik anwenden tragen dieses Merkmal. Eine Heuristik ist eine Annäherung oder eine "Daumenregel" für die Lösung eines Problems [GIG1999].

5.3.4.7 Merkmal: Vorberechnet

Die Berechnung der Distanz kann bei Bedarf oder im Vorfeld erfolgen. Beide Ansätze haben Vor- und Nachteile. Eine Berechnung bei Bedarf kann in Hochlastszenarien schnell zum Flaschenhals werden und Wartezeiten zur Folge haben. Eine Vorberechnung ist bei asymmetrischen Verfahren (siehe oben) kritisch, da hier die Abstände aller Objekte zu allen Objekten erneut berechnet werden müssen, sobald ein neues Informationsobjekt zu berücksichtigen ist. Dieses polynomiale Laufzeitverhalten kann bei einer großen Anzahl von Informationsobjekten zu sehr langen Wartezeiten bei der Neuaufnahme von Informationsobjekten führen.

5.3.5 CF-Technik

Das Collaborative Filtering ist ein interpersonelles Konzept, bei dem einem Benutzer Informationsobjekte auf Basis seines Benutzerprofils in Bezug auf andere Benutzerprofile angeboten werden.

Dabei sind das *benutzerbezogene CF*, das zunächst "Nachbarn" des Benutzers sucht und dann Empfehlungen auf Basis deren Verhaltens macht und das *objektbezogene CF*, das zu den vom Benutzer bewerteten Objekten ähnliche Objekte mittels des Verhaltens anderer Benutzer sucht und dann Empfehlungen ausgibt, zu unterscheiden.

Ferner kann die Ermittlung der Empfehlungen auf Basis des Verhaltens aller Benutzer (*speicherbasiert*) oder durch ein abgeleitetes Modell (*modellbasiert*) erfolgen.

Im Folgenden werden diese Begriffe näher erläutert.

5.3.5.1 Merkmal: Benutzerbezogenes CF

Beim benutzerbezogenen CF sind die "anderen" Benutzerprofile bezüglich des Verhaltens "Nachbarn" des Benutzers. Sie haben also in der Vergangenheit ein ähnliches Benutzerverhalten gezeigt. Die Empfehlung wird aus den Informationsobjekten gewonnen, welche die Nachbarn gewählt haben. Verfahren mit interpersonellem Ansatz arbeiten im Wesentlichen in zwei Schritten. Zunächst wird für den Benutzer U für alle Informationsobjekte I_n mit $n=1…n$ auf Basis des Verhaltens seiner Nachbarn berechnet, wie groß die Relevanz (das erwartete Interesse) $R(U,I_n)$ dieser Objekte für U sein wird. Im zweiten Schritt werden dann die Objekte mit dem höchsten Relevanz-Wert zur Auswahl gestellt.

5.3.5.2 Merkmal: Objektbezogenes CF

Das objektbezogene CF berechnet auf Basis der vom Benutzer U bewerteten Objekte "ähnliche" Objekte. Die Ähnlichkeit wird dabei auf Basis der Daten anderer Benutzer ermittelt, die neben den Objekten des Benutzers U auch die anderen Objekte bewertet haben.

5.3.5.3 Merkmal: Speicherbasiertes CF

Bei speicherbasierten Verfahren wird die komplette Basis der Benutzerprofile mit deren Verhalten zur Berechnung der $R(U,I_n)$ verwendet. Es werden in der Regel statistische Verfahren zur Bestimmung der Nachbarn verwendet [COV1967].

5.3.5.4 Merkmal: Modellbasiertes CF

In diesem Falle wird auf Basis der Benutzerprofile ein Modell berechnet, das dann für die Ermittlung der $R(U,I_n)$ verwendet wird. Die bekanntesten modellbasierten Ansätze sind Bayes´sche Netze und Clusterbildung aber auch Regelsysteme und neuronale Netze gehören dazu.

Bei der Clusterbildung werden Benutzergruppen mit ähnlichen Präferenzen gebildet. Auf Basis der Cluster werden dann Empfehlungen für den aktiven Benutzer erstellt. Dabei werden die Präferenzwerte der Benutzer aus dem Cluster des aktiven Benutzers in einem Durchschnittswert verdichtet. Dieser Durchschnittswert kann potenziell auch übergreifend auf Basis von Präferenzwerten der Benutzer anderer Cluster, in denen sich der aktive Benutzer ebenfalls befindet, ermittelt werden. Wobei die Präferenzwerte in diesem Falle mit dem Grad der Cluster-Zugehörigkeit gewichtet werden.

5.4 Klassifikation der Empfehlungssysteme

Die folgende Übersicht der wichtigsten Empfehlungssysteme soll in die drei Kategorien CF, CBF und Hybride aufgeteilt werden. Ferner wird in 5.4.4 (Seite 70) eine Übersicht der Klassifikation nach den oben aufgeführten Merkmalen gegeben.

Um einen schnellen Überblick über das jeweiligen Verfahren geben zu können, wurde das Zusammenspiel der einzelnen Merkmale in Form einer schematisierten Grafik zusammen gefasst.

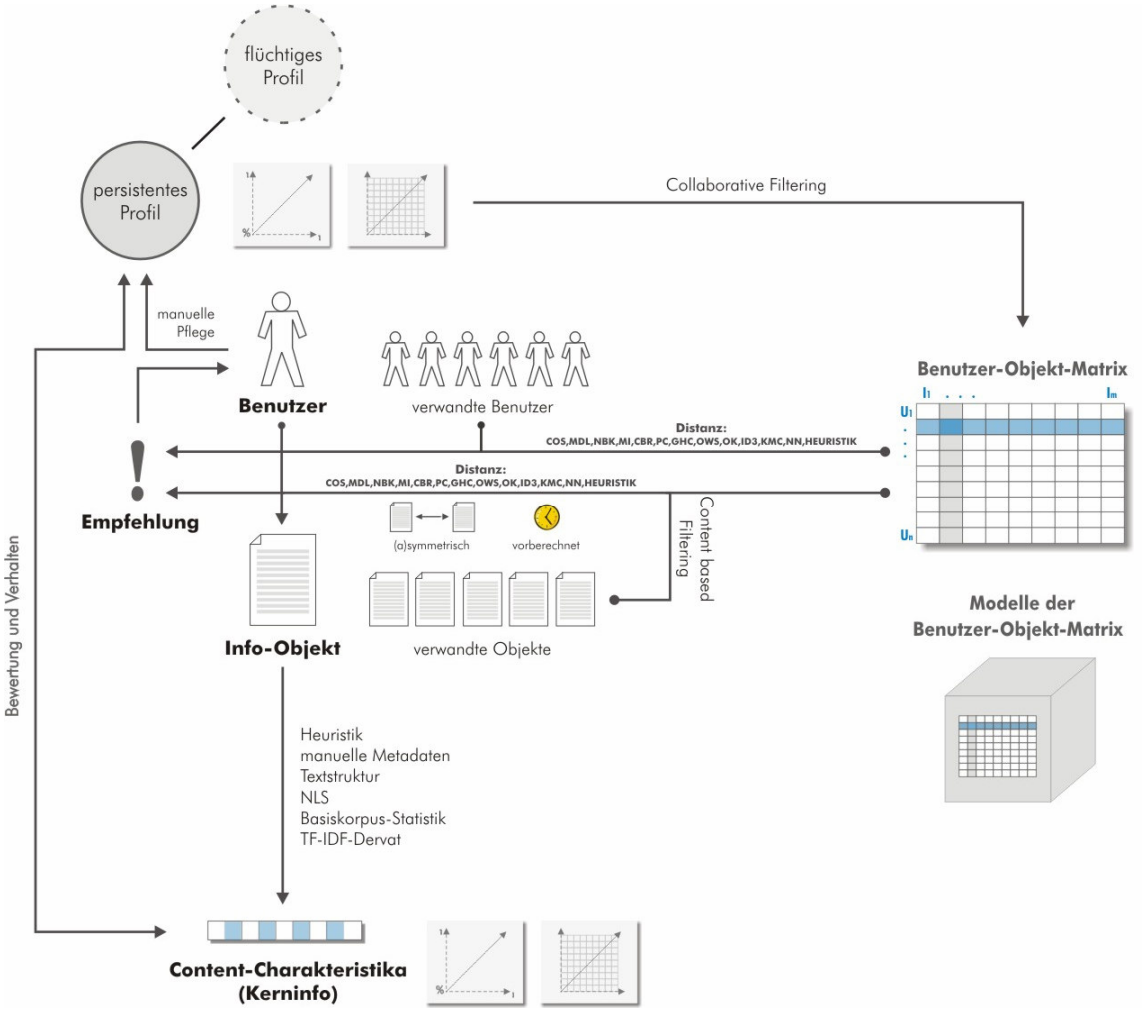


Abbildung 18: Schematische Darstellung des Zusammenspiels der Merkmale eines Empfehlungssystems

5.4.1 CF-Systeme

5.4.1.1 Tapestry [GOL1992]

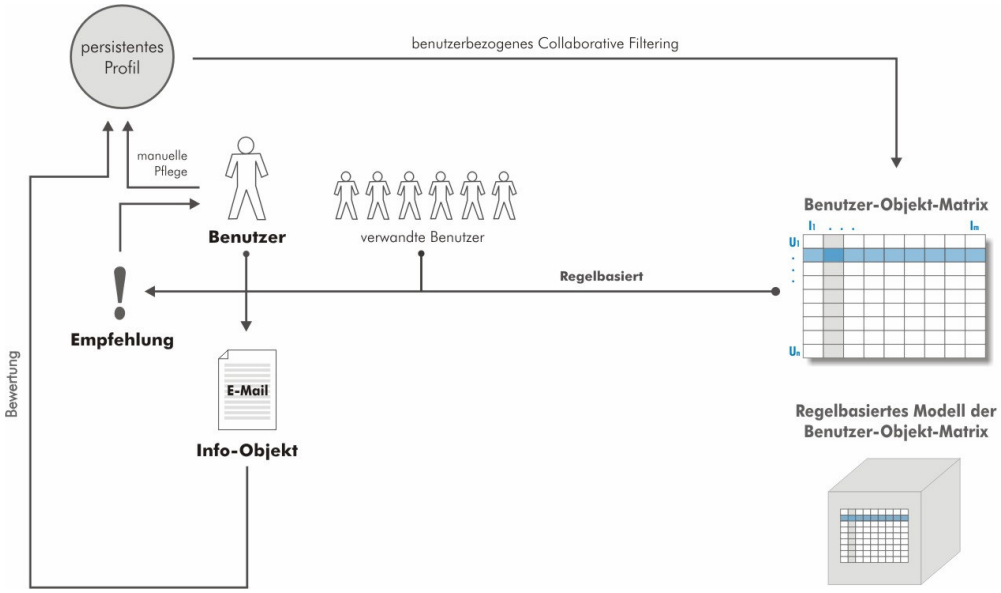


Abbildung 19: Tapestry

Das Verfahren arbeitet auf unstrukturierten Daten (E-Mail Texte). Diese werden von Benutzern explizit bewertet. Benutzer können Filter definieren (explizites Strukturprofil; modellbasiertes CF). Es kann beispielsweise alle E-Mails anzeigen, die von den Benutzern *B1* und *B2* als relevant eingestuft werden (regelbasierte Distanzermittlung, siehe Seite 29). Oder beispielsweise keine E-Mails, die von Benutzer *B3* als irrelevant eingestuft wurden. Wichtig ist also, dass die Benutzer sich untereinander kennen, um eine Auswahl von Benutzern mit ähnlichen Interessen treffen zu können.

5.4.1.2 Ringo [SHA1995]

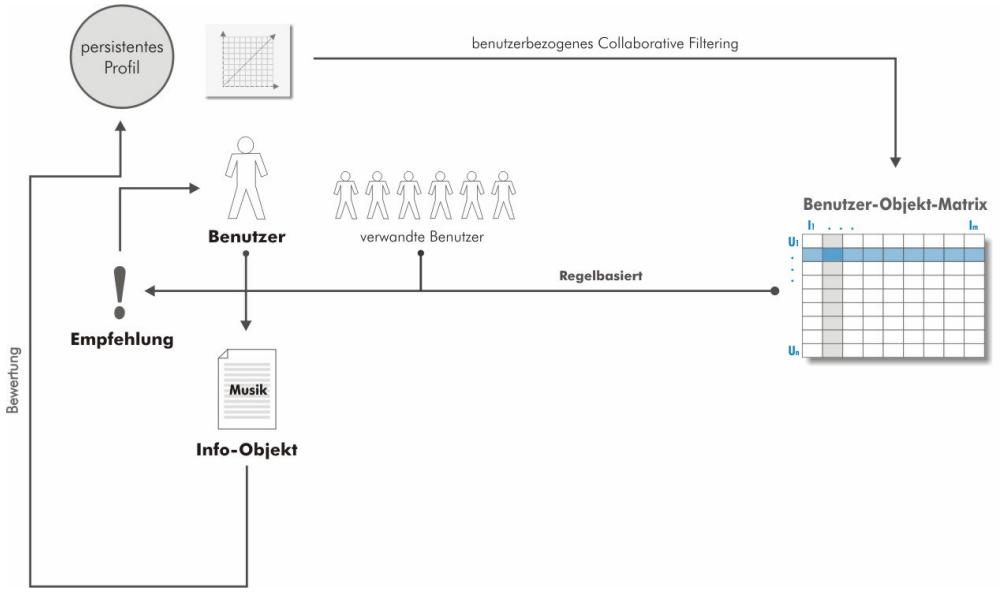


Abbildung 20: Ringo

Das Ringo Verfahren empfiehlt dem Benutzer Musik in Form von Interpreten und Alben. Dazu muss der Benutzer zunächst eine Vorschlagsliste von Interpreten auf einer Skala von 1 bis 7 bewerten. Diese Vorschlagsliste wird aus den am meisten bewerteten (erhöht Wahrscheinlichkeit verwandter Benutzer) und aus zufällig gewählten (garantiert Berücksichtigung aller Interpreten) Interpreten zusammengestellt.

Nach der Bewertung kann der Benutzer folgende Funktionen nutzen:

-Anzeige von Interpreten/Alben die dem Benutzer empfohlen werden

- Anzeige von Interpreten/Alben die dem Benutzer nicht empfohlen werden

- Anzeige der Empfehlung (Bewertung von 1 bis 7) zu einem Interpreten oder Album

Hinter den Empfehlungen steht ein CF Ansatz. Dabei wurden vier Verfahren zur Ermittlung verwandter Benutzer getestet, indem aus den Benutzerprofilen jeweils 20% der Empfehlungen entfernt und dann mit den Empfehlungen auf Basis dieser Verfahren verglichen wurden. Die Verfahren waren:

- Durchschnitt der quadrierten Differenzen
- Benutzerbezogener Korrelationskoeffizient (PC; siehe Seite 25)
- Angepasster benutzerbezogener Korrelationskoeffizient
- Objektbezogener Korrelationskoeffizient

Der angepasste benutzerbezogene Korrelationskoeffizient erzielte die besten Ergebnisse und wird daher in Ringo verwendet. Dabei werden die skalaren Wertungen der Benutzer in positiv und negativ transformiert, indem unabhängig von den Benutzerbewertungen ein fester Mittelwert in der Skalenmitte (4) zum Einsatz kommt. Formal wird für zwei Benutzerprofile v und w statt

$$\texttt{Distanz(v,w)} = \frac{\sum_{i=1\ldots n}\left((v_i-\bar{v})^2 * (w_i-\bar{w})^2\right)}{\sqrt{\sum_{i=1\ldots n}(v_i-\bar{v})^2 * \sum_{i=1\ldots n}(w_i-\bar{w})^2}}$$

die folgende Variante verwendet:

$$\texttt{Distanz(v,w)} = \frac{\sum_{i=1\ldots n}\left((v_i-4)^2 * (w_i-4)^2\right)}{\sqrt{\sum_{i=1\ldots n}(v_i-4)^2 * \sum_{i=1\ldots n}(w_i-4)^2}}$$

5.4.1.3 GroupLens [KON1997],[RES1994]

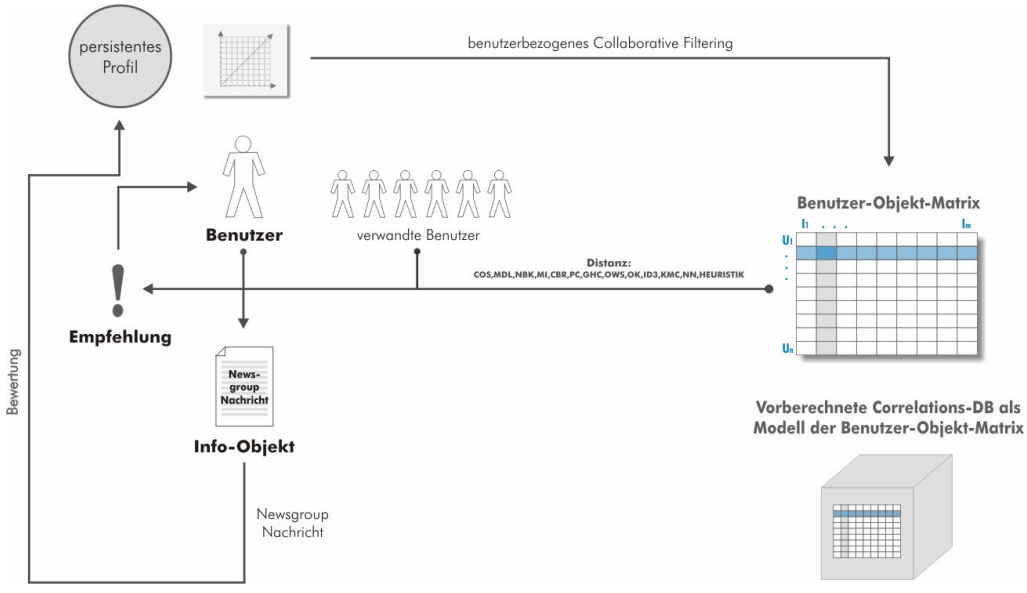


Abbildung 21: GroupLens

Das Verfahren gibt über einen CF-Algorithmus Empfehlungen für Usenet-Nachrichten aus. Die Benutzer geben anonym in einem proprietären Client explizite Bewertungen von Newsgroup-Artikeln auf der Skala von 1 bis 5 ab. Diese Bewertungen werden an einen zentralen GroupLens-Server geschickt. Dieser speichert die Bewertungen innerhalb von 60 Sekunden in der *Ratings* Datenbank und berechnet im 24 Stunden Rhythmus auf

Basis dieser Ratings die Korrelationen zwischen allen Benutzern und speichert diese in der *Correlations* Datenbank. Auf Basis dieser Datenbanken gibt der Server dann individuelle Empfehlungen für einzelne Benutzer aus. Die empfohlenen Nachrichten werden durch die betreffenden Newsgroup-Reader angezeigt.

5.4.1.4 Siteseer [RUC1997]

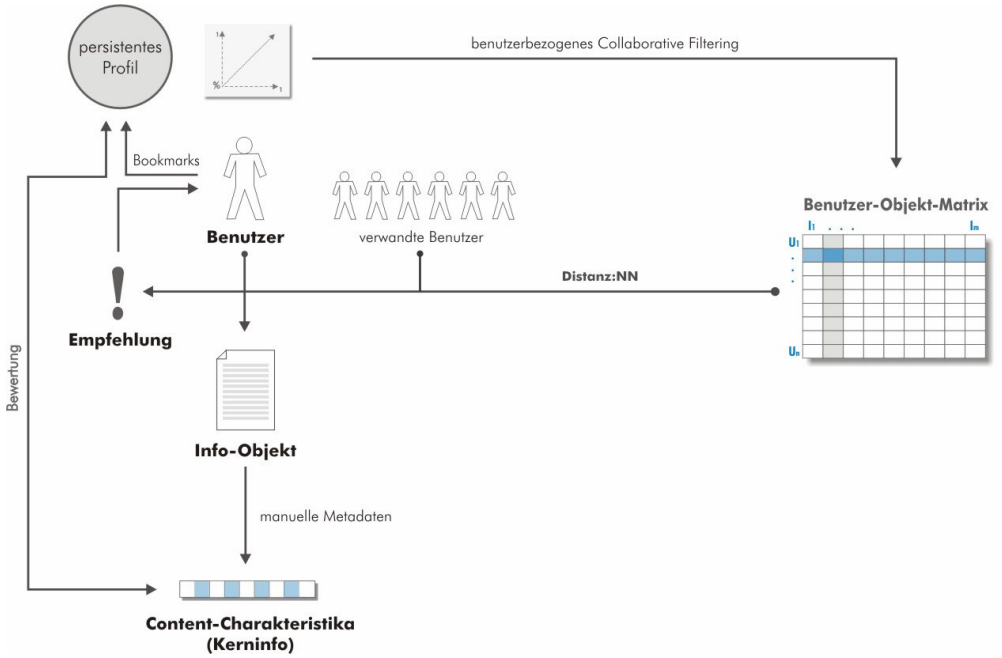


Abbildung 22: Siteseer

Siteseer basiert auf Browser-Bookmarks. Die Bookmarks eines Benutzers werden zentral gespeichert (boolesche Bewertung von Objekten). Auf diese Bookmarks, die insbesondere in Kategorien aufgeteilt werden können (manuelle Metadaten), hat der Benutzer jederzeit Zugriff, um Webseiten aufzurufen. Die Bookmarks eines Benutzers werden mit den Bookmarks anderer Benutzer auf gleiche Einträge hin verglichen. Je mehr identische Einträge bei zwei Benutzern vorliegen, umso ähnlicher werden diese Benutzer eingestuft. Das System liefert dem Benutzer dann die Bookmarks seiner nächsten Nachbarn (NN; siehe Seite 29).

5.4.1.5 Jester (Eigentaste) [GOL2000]

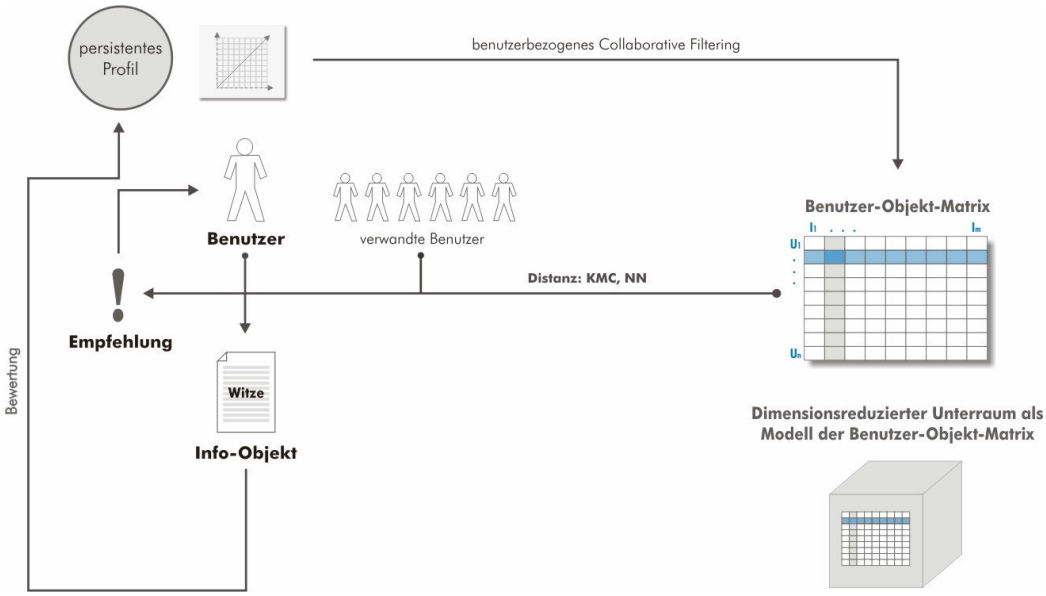


Abbildung 23: Jester

Das Jester Verfahren empfiehlt dem Benutzer Texte (Witze) auf Basis eines CF Ansatzes. Das Besondere ist, dass die Benutzer im Rahmen der Profilbildung dieselben Texte bewerten müssen. Dadurch wird das Problem der "Spärlichkeit" (siehe Seite 18) eliminiert. Dass der Benutzer auch ihm potenziell unbekannte Texte bewerten muss, versucht das Verfahren durch einen Gewichtungsfaktor in Form des Bekanntheitsgrades zu korrigieren.

Die Bewertung selbst erfolgt auf einer fein granularen Skala in Form eines Schiebereglers mit 200 nicht einzeln markierten Bewertungsstufen. Jede Bewertung wird normalisiert, indem der Durchschnitt aller Bewertungen subtrahiert und dann durch die Standardabweichung dividiert wird.

Durch Hauptkomponentenanalyse (*Principal Component Analysis (PCA)*, [PEA1901]) wird die Dimension des zu betrachtenden Vektorraums auf einen weniger dimensionalen Unterraum reduziert, in dem der Hauptteil der Datenvarianz liegt. Mit PCA wird vereinfacht ausgedrückt das Maximum an Datenvarianz in einem Minimum an Dimensionen abgebildet.

Abschließend werden mit rekursiver geometrischer Klassifikation (*GK;* siehe Seite 26) die Benutzergruppen aufgeteilt, indem Cluster (konkret 40) gebildet werden. Ein neuer Benutzer erhält dann wie folgt seine Empfehlungen:

- Bewertungen für die ausgewählten Texte einholen

- Wertungsvektor (=Profil) mittels PCA in den Unterraum abbilden

- zugehörigen Cluster wählen

- Empfehlungen aus diesem Cluster anbieten und bewerten lassen

Ein wesentlicher Vorteil des *Eigentaste* Verfahrens besteht in der konstanten Laufzeitkomplexität von `O(1)`, die Skalierungsprobleme eliminiert.

5.4.1.6 Amazon.com [LIN2003],[LIN2001]

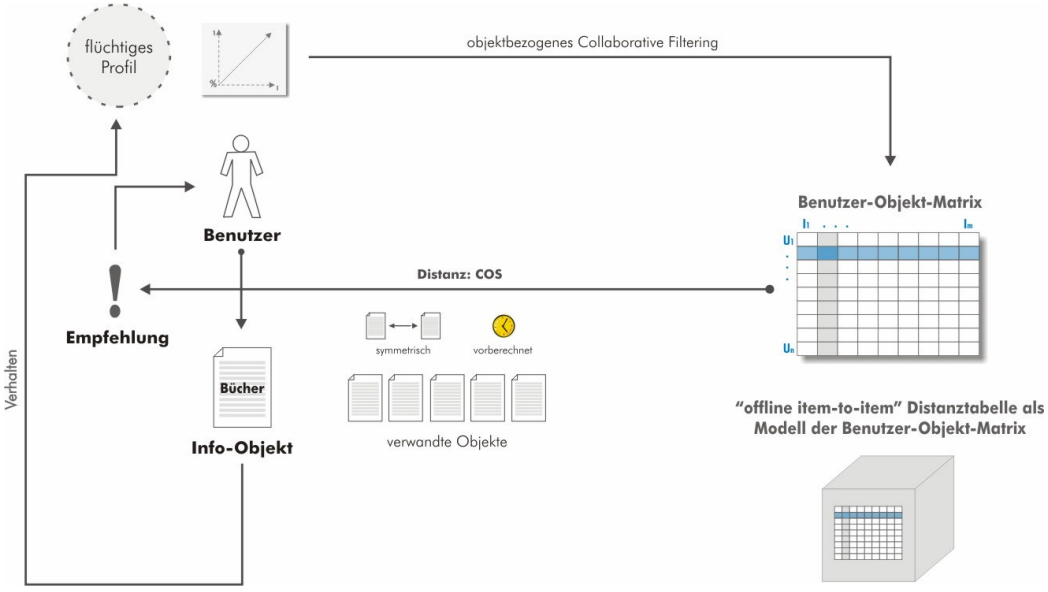


Abbildung 24: Amazon.com

Das Amazon CF-Verfahren empfiehlt dem Benutzer Bücher aufgrund der aktuell oder vor kurzem (in der gleichen Session; flüchtiges Profil[5]) in seinem Warenkorb befindlichen Bücher. Basis der Empfehlungen ist eine "offline item-to-item" Distanz-Tabelle, die aufgrund des Benutzerverhaltens (Kauf) vorberechnet wird.

Die Vorberechnung findet zu Büchern "ähnliche" Bücher, indem sie analysiert, welche Bücher von gleichen Benutzern gekauft werden. Dazu wird zu jedem Buch ein Vektor aufgebaut. Es handelt sich um ein objekt-

[5] Amazon.com bietet auch eine "Personalisierten Shop" für den Benutzer. Hier kommen Daten des ebenfalls vorhandenen persistenten Benutzerprofils zum Einsatz.

bezogenes CF Verfahren bei dem das Kosinus Ähnlichkeitsmaß für die Distanzermittlung zwischen zwei Buch-Vektoren verwendet wird. Eine Besonderheit ist die Vorberechnung dieser Distanzen. Nur dadurch kann gewährleistet werden, dass die Performanz auch bei den großen Benutzer- (29 Millionen) und Buchzahlen (mehrere Millionen) ausreichend ist. Ist die Distanz-Tabelle einmal berechnet, so ist die Laufzeit lediglich von der Anzahl der Bücher, die der Benutzer in seinem Warenkorb hat (oder kürzlich hatte), abhängig. Für jedes dieser Bücher werden dann durch einfache Punktselektion auf der Distanz-Tabelle die ähnlichsten Bücher ermittelt und anschließend zu einer Empfehlungsliste zusammengeführt.

5.4.1.7 SurfLen [FU2000]

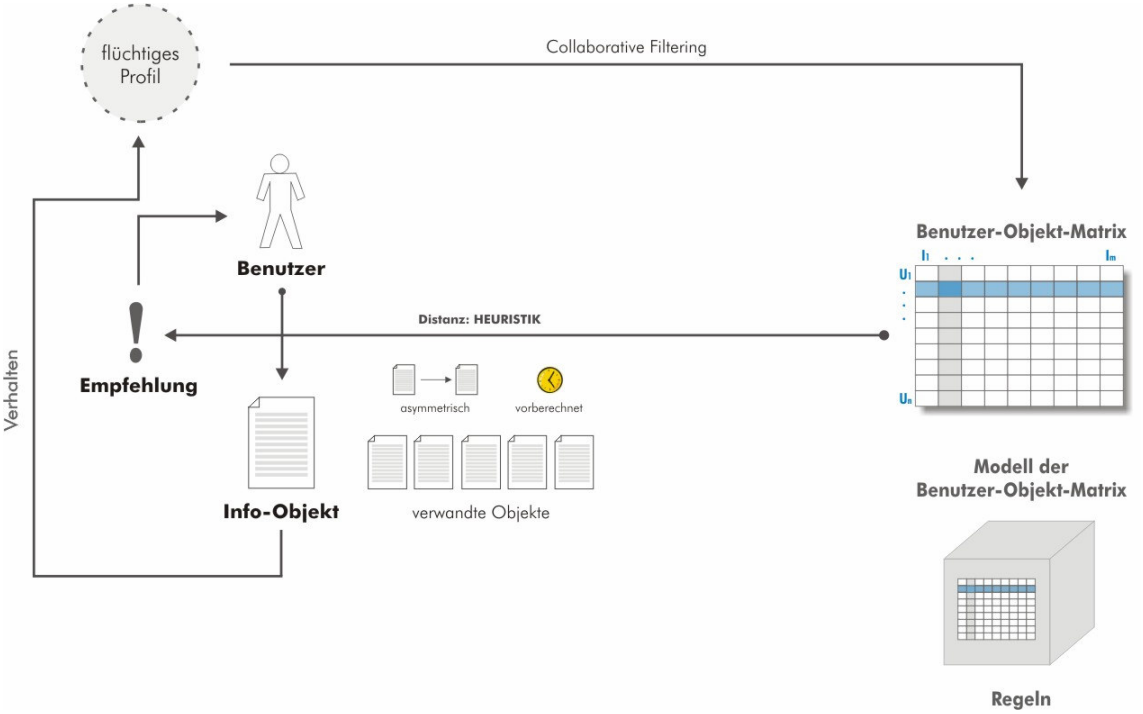


Abbildung 25: SurfLen

Das SurfLen Verfahren basiert auf objektbezogenem Collaborative Filtering. Auf Basis des Benutzerverhaltens, ermittelt durch die in einer Session aufgerufener Webseiten, werden automatisiert Regeln (so genannte Assoziationsregeln) abgeleitet, die ein Modell für die Benutzer-Objekt-Matrix darstellen. Dem Benutzer werden zur aktuell betrachteten Webseite Empfehlungen eingeblendet. Technisch erfolgt dies durch ein Browser-Plugin.

Eine neue Regel wird immer iterativ erzeugt. Dabei kommt die im Folgenden beschriebene Heuristik zum Einsatz. Zunächst werden die Informationsobjekte selektiert, deren Vorkommen über alle Sessions einen bestimmten Schwellwert (Prozentwert *minsup*) übertrifft. Dann werden die daraus bildbaren 2-Tupel (i_1, i_2) von Informationsobjekten gebildet, deren gemeinsames Vorkommen in allen Sessions wiederum den Schwellwert *minsup* überschreitet. Gleiches wird dann für 3-Tupel, 4-Tupel et cetera durchgeführt. Durch *minsup* wird die Zahl der zu betrachtenden Tupel klein gehalten.

Die so gefundenen Tupel werden zum Erzeugen der Regeln verwendet. Dazu wird aus einem Tupel $(i_1, i_2, \ldots, i_k)$ die Regel $X \rightarrow Y$ mit $X = (i_1, i_2, \ldots, i_{k-1})$ und $Y = (i_k)$ erzeugt. Daraus folgt, dass die Empfehlungen potenziell asymmetrisch sind.

Die Tupelerzeugung ist der im Rahmen der Regelerzeugung aufwändige Vorgang. Die Berechnung durch die SRE (SurfLen Recommendation Engine) ist ein dauerhaft laufender Prozess, da die Regeln ständig an die Daten neuer Sessions (Benutzerverhalten) angepasst werden müssen.

Damit dem Benutzer auch dann eine Empfehlung gegeben werden kann, wenn keine Regel auf sein aktuelles Verhalten zutrifft, wird in diesem Falle die Schnittmenge der Informationsobjekte aus der Benutzersession und der Informationsobjekte aus den Sessions in der SurfLen Datenbasis gebildet (in diesem Falle erfolgt das CF speicherbasiert und objektbezogen).

Die Empfehlungen bestehen in diesem Fall aus den Informationsobjekten der Session S_j mit der größten Schnittmenge zu B, wobei die in der Benutzersession B vorkommenden Elemente ausgeblendet werden:

$B=(i_1,i_2,\ldots,i_k),\ S=(s_1,s_2,\ldots,s_m)$

Empfehlung:=S_j-B mit $\max_i(B \cap S_j)$

5.4.1.8 PocketLens [MIL2004]

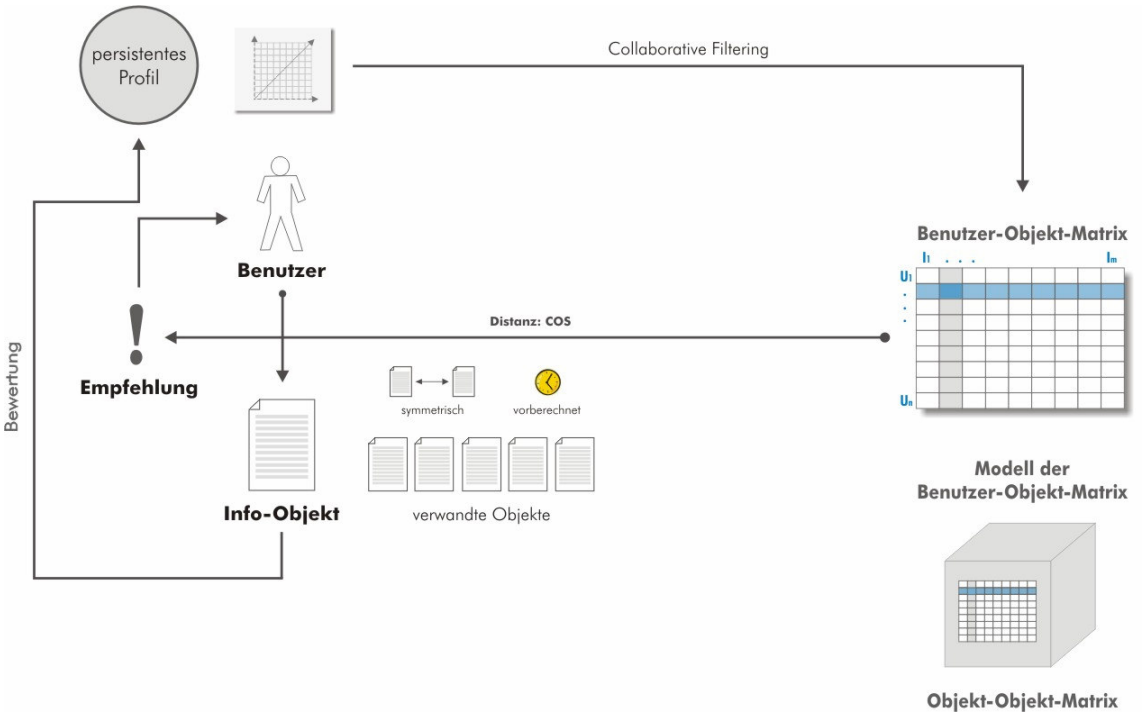


Abbildung 26: PocketLens

Dem PocketLens Verfahren liegen als einem Vertreter des Collaborative Filtering vier Zielsetzungen zugrunde: *portability*, *palmtop compatability*, *user control* und *accuracy*. Die *portability* meint die Nutzungsmöglichkeit an beliebigem Standort und insbesondere auch im Offline-Betrieb. Dies und die *palmtop compatability* werden durch einen modellbasierten CF-Ansatz erreicht. Das Modell wird erstellt und lokal gespeichert während der Benutzer online ist. Die CF-Matrix wird in eine Objekt-Objekt-Matrix transformiert und auf die Objekte reduziert, die durch den Benutzer bewertet wurden. Das vektorbasierte Benutzerprofil besteht aus Bewertungen von Informationsobjekten. Der Benutzer kann selbst bestimmen (*user control*) welche Teile seines Profils für andere Benutzer verfügbar sind. Die *accurracy* wird durch ein modifiziertes objektbasiertes CF-Verfahren erreicht.

Dazu wird zunächst aus der CF-Matrix eine Objekt-Objekt-Matrix gebildet. Letztere beinhaltet die Relevanz mit der auf Basis von Informationsobjekt I_1 das Informationsobjekt I_2 empfohlen wird. Ein trivialer Weg diese Matrix zu konstruieren ist die Summation einer identischen Bewertung zweier Objekte durch einen Benutzer. Das PocketLens Verfahren verwendet ein weiterentwickeltes inkrementelles Verfahren:

Seien $O_1,\ldots,O_n$ die Zeilen und $N_1,\ldots N_n$ die Spalten der nicht-reduzierten Objekt-Objekt-Matrix, die bei n Objekten n Zeilen und n Spalten umfasst. Wenn die Objekte O_i (durch den Benutzer selbst bewertete Objekte) und N_j die Bewertungen w_k und u_k durch einen anderen Benutzer B_k erhalten, wird die Zelle (O_i,N_j) der Matrix wie folgt angepasst:

$$Cooccur(O_i,N_i) = Cooccur(O_i,N_i)+1$$
$$PtLenW(O_i,N_j) = PtLenW(O_i,N_j)+w_k*w_k$$
$$PtLenU(O_i,N_j) = PtLenW(O_i,N_j)+u_k*u_k$$
$$PDot(O_i,N_j) = PDot(O_i,N_j)+ w_k*u_k$$

Die Distanz zweier Objekte O_i und N_j wird durch eine Kosinus Ähnlichkeitsfunktion auf Basis der Zellwerte berechnet:

$$sim(O_i, N_j) = F_k * \frac{PDot(O_i, N_j)}{\sqrt{PtLenU(O_i, N_j)} * \sqrt{PtLenW(O_i, N_j)}}$$

mit $F_k=1$ falls $Cooccur(O_i, N_j) >= 50$ und $F_k=Cooccur(O_i, N_j)$ sonst. Sie wertet Benutzer B_k ab, die hohe Übereinstimmung aber nur wenige gemeinsame Objekte besitzen.

5.4.1.9 PACT [MOB2000]

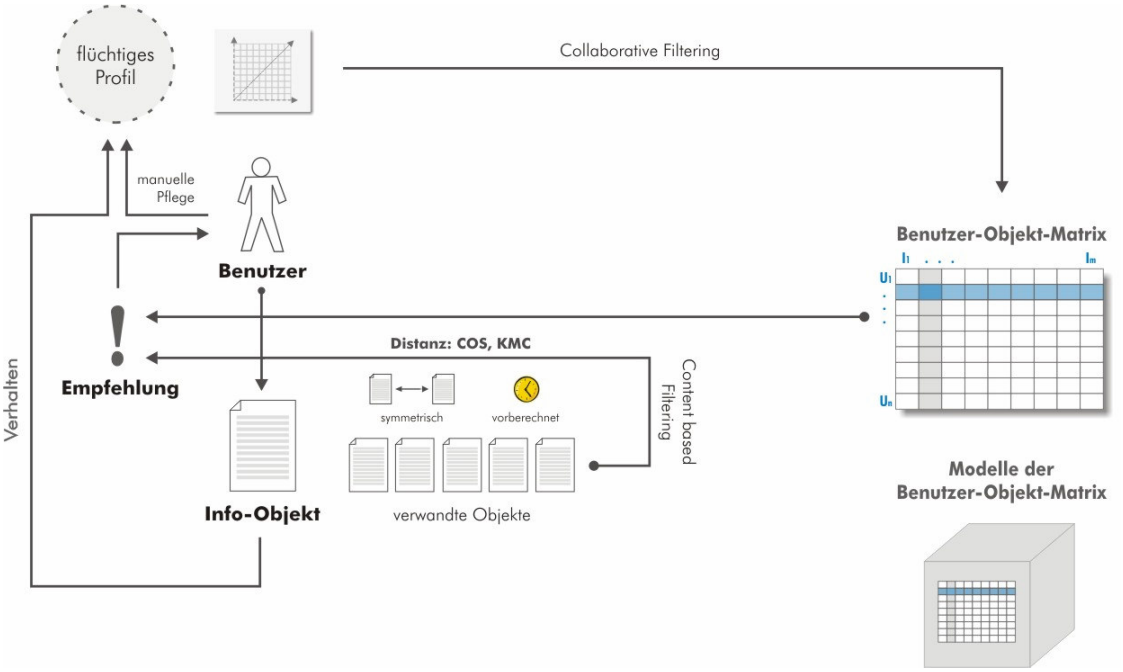


Abbildung 27: PACT

Beim PACT Verfahren wird auf Basis des Benutzerverhaltens auf einer Website ein flüchtiges Profil in Form eines gewichteten Transaktionsvektors gebildet. Die Gewichtung einer Seite resultiert dabei aus der gemessenen Lesedauer. Außerdem erzeugt PACT auf Basis der Transaktionen aller Benutzer Kategorien, indem das Mittel aller zu einer Kategorie gehörigen Transaktionsvektoren durch Addition der Komponenten und Division durch die Anzahl der Vektoren ermittelt wird (modellbasiertes CF). Welcher Klasse ein Transaktionsvektor zugeordnet wird, entscheidet das Verfahren mit dem Klassifikationsverfahren *k-means clustering* (*KLS*, siehe Seite 26).

Empfehlungen werden dem Benutzer in der aktuell angezeigten Webseite gegeben (diese wird dazu vor der Auslieferung an den Benutzer modifiziert). Dabei wird eine Webseite umso mehr empfohlen, je höher die Summe ihrer Gewichtungen über alle gemittelten Klassenvektoren ist. Zusätzlich fließt die Übereinstimmung von Transaktionsvektor (der auf "n" Schritte limitiert wird, um die Wahrscheinlichkeit inhaltlich zusammenhängender Webseiten zu erhöhen) und den einzelnen Klassenvektoren als Korrekturfaktor ein, wobei als Distanzfunktion das Kosinus Ähnlichkeitsmaß (*COS*) zum Einsatz kommt. Webseiten, die in der aktuellen Transaktion enthalten sind, werden ausgefiltert:

Empfehlung = P_i mit $p_i \in (p_1, \ldots, p_n)$ = Seiten in einem Klassenvektor C

und

$\surd$ $[(Gewichtung\ (p_i, C_j) * COS(Transaktion, C_j)] >= Schwellwert$

mit C_j = Klassenvektoren $(j=1 \ldots m)$, $i=1 \ldots n$

5.4.2 CBF-Systeme

5.4.2.1 The information lens [MAL1986]

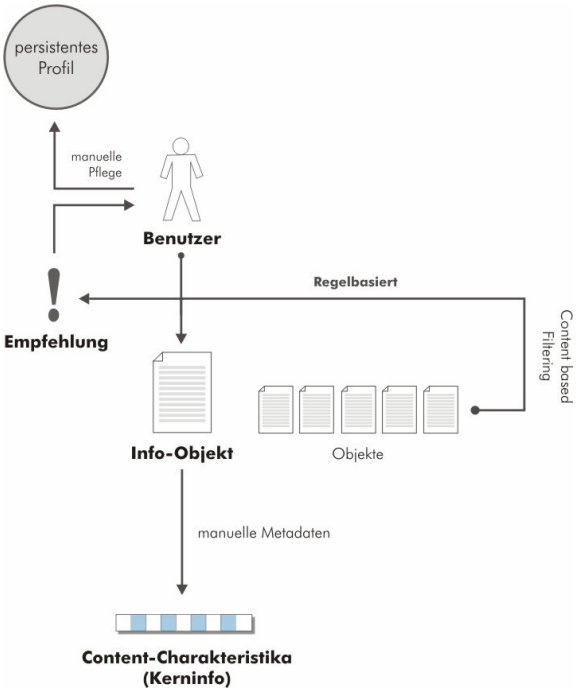


Abbildung 28: The information lens

Das System setzt auf einem E-Mail Dienst auf und arbeitet auf semi-strukturierten Daten, indem es semi-strukturierte E-Mail-Templates anbietet. Das heißt, neben Volltext werden noch Meta-Daten (beispielsweise "Ort", "Produkt" et cetera) mitgeführt, die vom Sender einer E-Mail ausgefüllt werden. Der Sender kann ferner durch einen Adressaten "lens" (spezielles E-Mail-Konto) die Verteilung der E-Mail an alle potenziell interessierten Empfänger einleiten. Ein Empfänger wiederum erhält solche E-Mails, wenn diese durch eines seiner E-Mail-Templates "erkannt" wird (regelbasierte Distanzermittlung). Die Erkennung beschränkt sich hier auf eine einfache Übereinstimmung der strukturierten Daten einer E-Mail mit einem E-Mail-Template (beispielsweise der passende "Ort").

5.4.2.2 Infoscope [FIS1991]

Infoscope verwendet Filterregeln (Distanzermittlung auf Basis einer Regel; siehe Seite 29), um Usenet-Nachrichten zu empfehlen. Dabei protokollieren Agenten das Benutzerverhalten und erzeugen daraus automatisch Filter-Regeln auf Basis der Präferenzen des Benutzers. Diese Filter-Regeln werden dem Benutzer dann als "virtuelle Newsgroups" angeboten. Die Benutzer können Regeln annehmen, verändern oder ablehnen. Diese Aktionen werden wiederum von den Agenten ausgewertet, um zukünftige Empfehlungen zu verbessern. Filter-Regeln müssen also nicht aktiv erstellt, sondern nur bewertet werden. Durch das aktive Angebot wird der Benutzer auf eine potenziell veränderte Interessenslage aufmerksam gemacht.

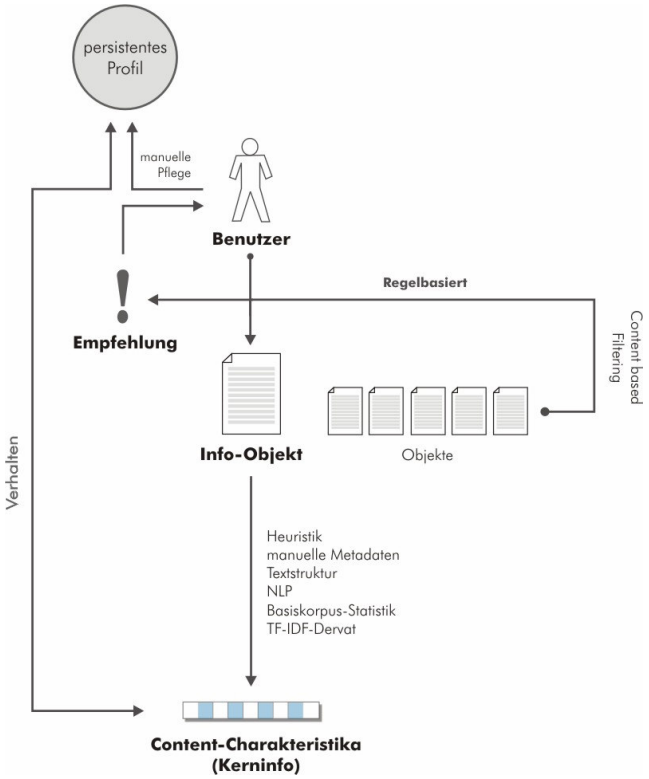


Abbildung 29: Infoscope

5.4.2.3 Newsweeder [LAN1995]

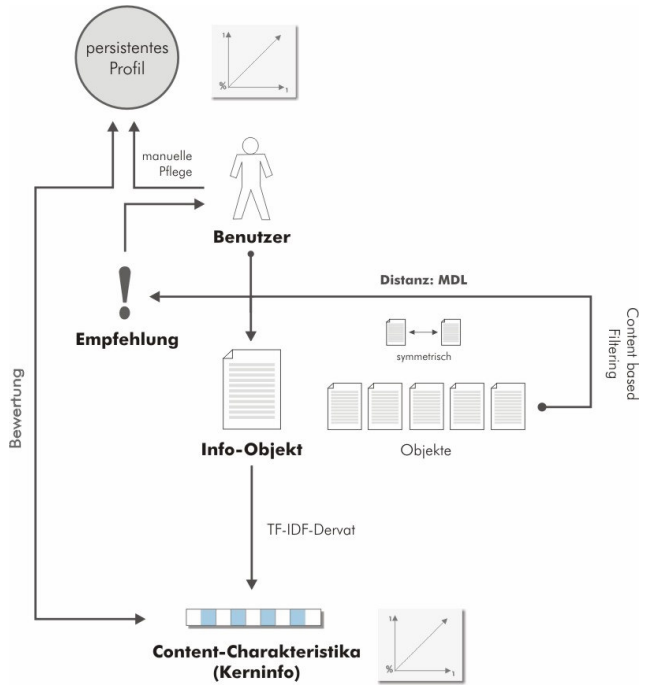


Abbildung 30: Newsweeder

Bei Newsweeder wird ein Benutzerprofil auf Basis expliziter Bewertungen von Newsgroup-Beiträgen auf einer Skala von 1 bis 5 gebildet. Texte werden dabei in der ersten Variante von Newsweeder durch Wortvektoren mit binären Werten (0,1 für Wort nicht vorhanden oder vorhanden) repräsentiert.

Um die Komplexität des Verfahrens zu reduzieren, werden in einer weiteren Variante die Worte durch ein TF-IDF-Derivat gewichtet, die Abstände der Vektoren mit dem Kosinus Ähnlichkeitsmaß ermittelt und dann mit einem NN-Verfahren (nearest neighbor) zu Kategorien zusammengefasst.

Die finale Newsweeder Variante arbeitet mit einem MDL Verfahren (minimum description length). Ein Wortvektor `v` wird bei Newsweeder der Kategorie `K` zugeordnet, deren Codierung zusammen mit `v` am wenigsten Speicherplatz benötigt.

5.4.2.4 Letizia [LIE1995]

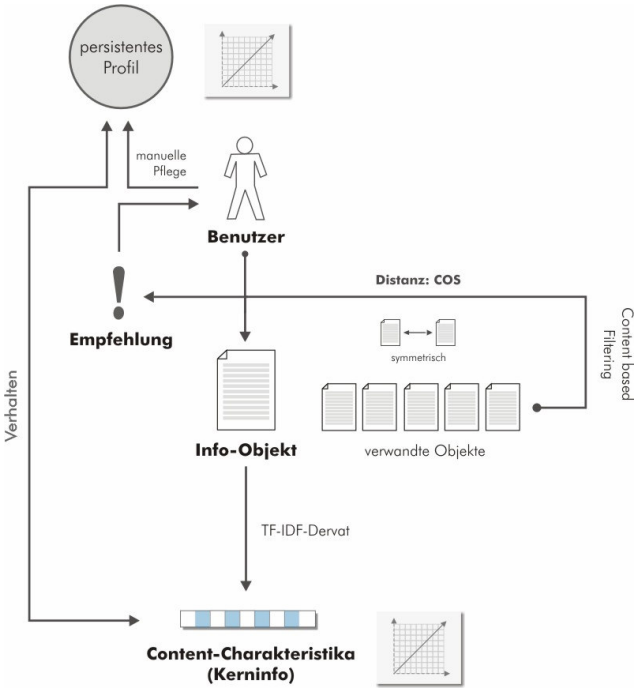


Abbildung 31: Letizia

Beim Letizia Verfahren wird das Benutzerverhalten im Webbrowser in folgender Form ausgewertet: Zur aktuell angezeigten Seite werden mit einem TF-IDF-Derivat die wichtigsten Schlüsselworte ermittelt und dem Benutzerprofil, das ein gewichteter Wortvektor ist, hinzugefügt. Außerdem werden die von der aktuell angezeigten Seite verlinkten Webseiten analysiert. Und zwar wie bei einer Patrouille (daher der Begriff "*reconnaissance agent*") in immer tieferen Link-Hierarchie-Stufen, solange der Benutzer die aktuelle Seite betrachtet. Die analysierten Seiten werden mit einem TF-IDF-Derivat in Wortvektoren transformiert und mittels des Kosinus Ähnlichkeitsmaß mit dem Benutzerprofil verglichen. Von allen möglichen Links der verschiedenen Stufen, die von der aktuell betrachteten Seite ausgehen, werden dem Benutzer die am besten zu seinem Profil passenden angeboten. Dabei wird das bestimmende Schlüsselwort, aufgrund dessen die Empfehlung ausgesprochen wurde, mit angegeben.

5.4.2.5 WebWatcher [ARM1995],[JOA1995],[JOA1997]

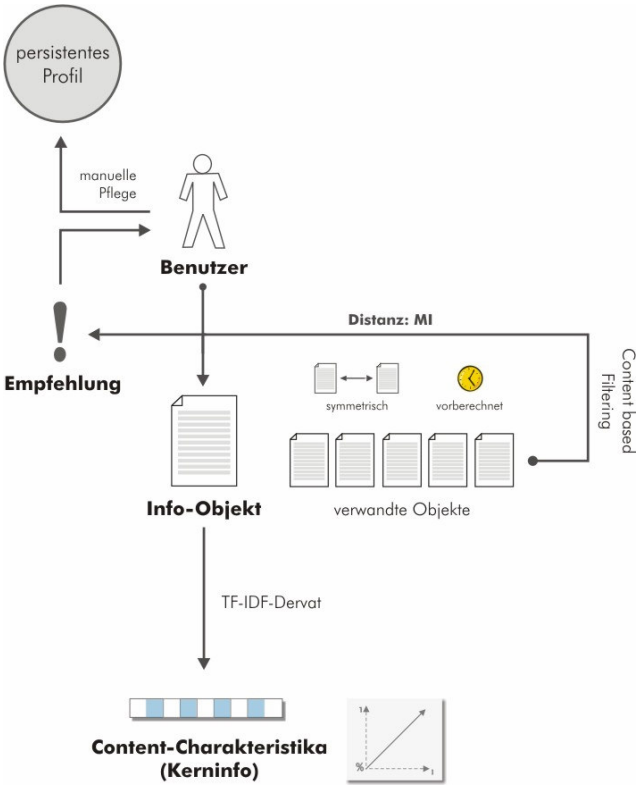


Abbildung 32: WebWatcher

Das WebWatcher Verfahren unterstützt den Benutzer bezüglich Webseiten auf Basis seines Benutzerprofils durch zwei wesentliche Arten von Empfehlungen:

- Bestehende Hyperlinks: werden in einer Webseite hervorgehoben, wenn als empfehlenswert betrachtet
- Neue Webseiten: die zur aktuellen Webseite Passenden werden empfohlen

Das Benutzerprofil besteht aus expliziten strukturierten Angaben (Interessensspezifikation). Wesentlich sind dabei die Suchworte, die der Benutzer angibt. Diese werden verwendet, um Hyperlinks im angezeigten Text hervorzuheben. Zur Bewertung der Hyperlinks wird das Kosinus Ähnlichkeitsmaß zwischen Benutzerprofil in Form eines Wortvektors der Suchworte und den Hyperlinks in Form der binären Wortvektoren $h_{1\ldots n}$ berechnet, wobei h_i aus den Worten

- im Titel des Textes, in dem der Link vorkommt
- im Satz, in dem der Link vorkommt
- im Linktext

gebildet wird.

Bei der Empfehlung "ähnlicher" Webseiten zur gerade angezeigten Webseite kommt CBF zum Einsatz, bei dem die Charakteristik von Webseiten durch die eingehenden Hyperlinks beschrieben wird. Zwei Webseiten, die durch Hyperlink-Vektoren (Matrixspalten) repräsentiert sind, werden mit dem Ähnlichkeitsmaß der Transinformation (*mutual information, MI*) verglichen.

5.4.2.6 Syskill & Webert [PAZ1996]

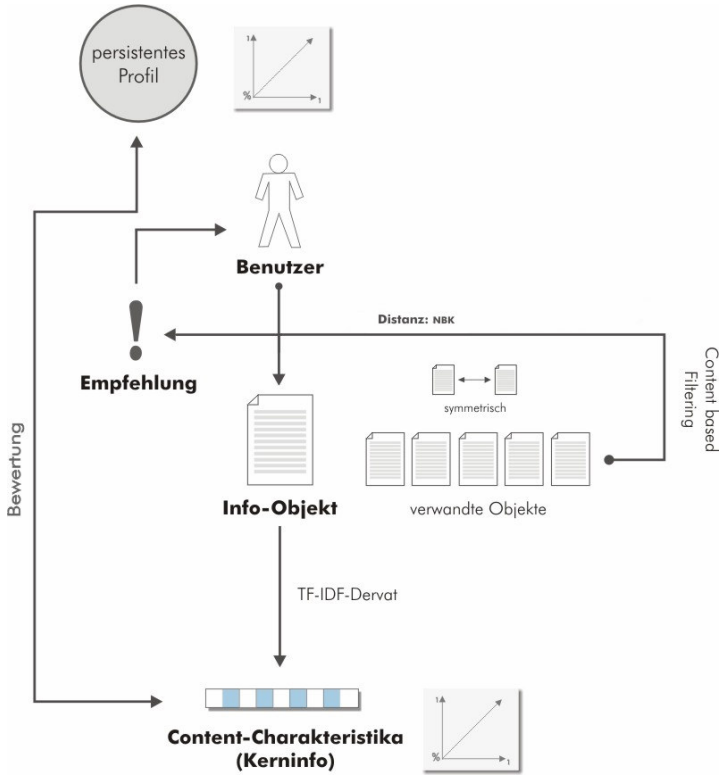


Abbildung 33: Syskill & Webert

Bei Syskill & Webert werden auf Basis der Bewertung von Webseiten Benutzerprofile generiert und dann neue Webseiten, die diesen ähnlich sind, empfohlen. Die Texte der Webseiten werden mit einem TF-IDF-Derivat in einen booleschen Wortvektor transformiert. In das Benutzerprofil, das ebenfalls einen booleschen Wortvektor darstellt, werden vereinfacht ausgedrückt die Worte aufgenommen, die in gut bewerteten Texten häufig vorkommen, also vermeintlich "bedeutend" für den Benutzer sind. Nachdem sechs verschiedene Verfahren Distanz-Ermittlung vorgestellt und evaluiert werden, wird der Naive Bayes-Klassifikator als Mittel der Wahl definiert.

5.4.2.7 Remembrance Agent [RHO1996], [RHO1999]

Der Remembrance Agent ermittelt aufgrund eines vom Benutzer gerade bearbeiteten Textes ähnliche Texte. Dabei kann der Benutzer einstellen, in welchem Radius (konkret wie viele Zeichen) um die aktuelle Cursorposition der Text für Empfehlungen berücksichtigt werden soll. Dabei können für die "n" Empfehlungen, die im proprietären Client zeilenweise angezeigt werden, unterschiedliche Text-Umfänge eingestellt werden. Die empfohlenen Texte werden dem Benutzer in einem speziellen Frontend, in dem er auch den aktuellen Text bearbeitet, angeboten. Dabei wird die SMART Suchmaschine ("*Salton's Magical Automatic Retriever of Text*", später auch als "*System for the Manipulation and Retrieval of Text*" bezeichnet) [SAL1965], [SAL1971] eingesetzt. Diese nutzt in der eingesetzten Variante den von Rocchio und Salton entwickelten Rocchio-Ansatz [SAL1971] . Dabei werden die Texte zunächst mit einem TF-IDF-Derivat in gewichtete Wortvektoren transformiert. Dies wird beim Remembrance Agent einmal pro Nacht für neue Texte durchgeführt (vorberechnet). Die Ähnlichkeit von Texten wird mit dem Kosinus Ähnlichkeitsmaß der Wortvektoren ermittelt.

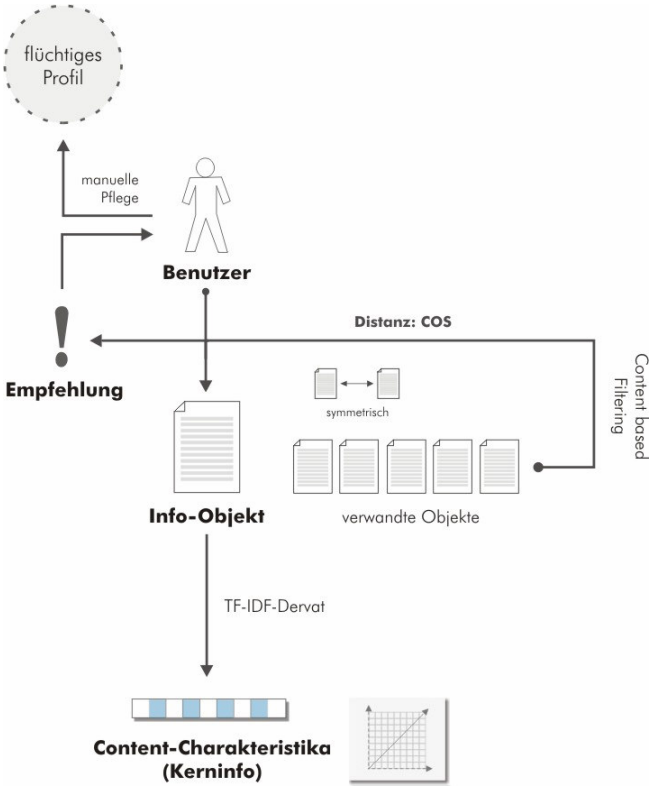


Abbildung 34: Remenbrance Agent

5.4.2.8 InfoFinder [KRU1997]

Das Infofinder Verfahren beobachtet den Benutzer beim Aufrufen von Webseiten. Es extrahiert aus den durch Anklicken eines "Smiley" explizit positiv bewerteten Webseiten mit einer Heuristik die bestimmenden Worte. Diese Worte werden in einen Entscheidungsbaum integriert, der das Benutzerprofil repräsentiert. Welchem Profil eine Webseite beziehungsweise deren bestimmende Worte hinzugefügt werden, bestimmt der Benutzer explizit durch Zuordnung zu einem bestehenden oder Anlegen eines neuen Profils. Der Baum eines Profils wird dann in eine boolesche Anfrage gewandelt und an eine Standard-Suchmaschine gesendet. Diese Anfragen werden offline (nachts) durchgeführt und dem Benutzer werden interessante neue Webseiten am nächsten Tag zur Verfügung gestellt. Die bestimmenden Worte einer Webseite werden mit folgender Heuristik ermittelt:

- Worte, die in einer Stoppwortliste stehen, sind grundsätzlich unwichtig
- komplett großgeschriebene Worte sind wichtig (Vermutung: es handelt sich um ein Akronym)
- Worte vor einem in Klammern oder Anführungszeichen stehenden komplett großgeschriebenen Wort sind wichtig (Vermutung: Definition eines Akronyms)
- In Klammern oder Anführungszeichen stehende Worte nach einem komplett großgeschriebenen Wort sind wichtig (Vermutung: Definition eines Akronyms)
- Anders formatierte Wortfolgen von zwei bis drei Worten, die kein eigenständiger Satz sind, sind wichtig (Vermutung: erstmalige Verwendung eines wichtigen Wortes)
- Worte in Aufzählungen sind wichtig
- Worte in Überschriften sind wichtig
- Worte in Bildunterschriften sind wichtig
- Worte in Tabellenspalten und -zeilen sind wichtig
- Oftmals wiederholte Wortfolgen sind wichtig
- Substantive in direkter Folge sind wichtig (Vermutung: Fachbegriff)
- Worte, die Sonderzeichen (beispielsweise Bindestrich) oder Ziffern enthalten, sind wichtig
- Worte mit Großbuchstaben im Wort sind wichtig

Es kommt ein Thesaurus zum Einsatz, um Synonyme zu erkennen und als "identisch" bewerten zu können.

Der Entscheidungsbaum eines Profils wird - solange noch nicht 10 Webseiten zugeordnet sind - mit jeder neu zugeordneten Webseite nach dem *ID3* (siehe Seite 26) Algorithmus [QUI1986] komplett neu initialisiert. Ab der 11. Webseite wird eine neue Webseite nur noch eingefügt, ohne den Baum neu aufzubauen.

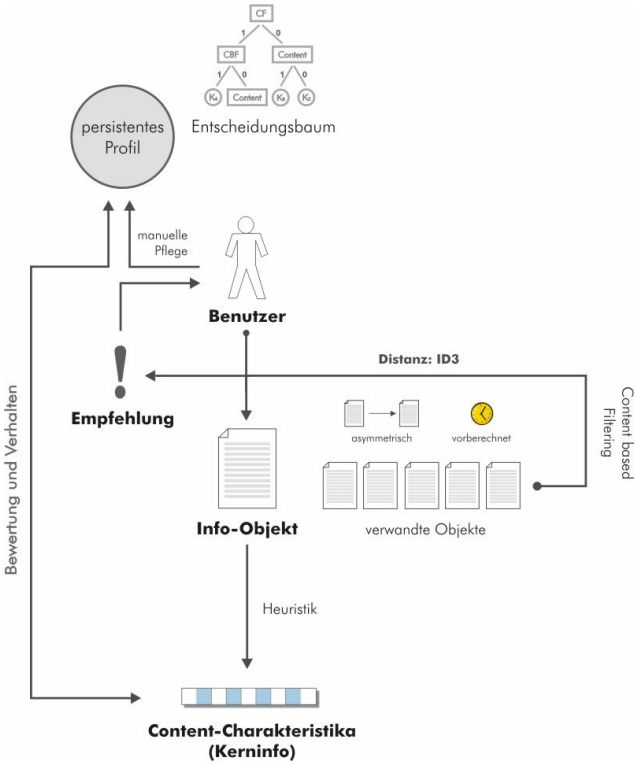


Abbildung 35: InfoFinder

5.4.2.9 Amalthaea [MOU1997]

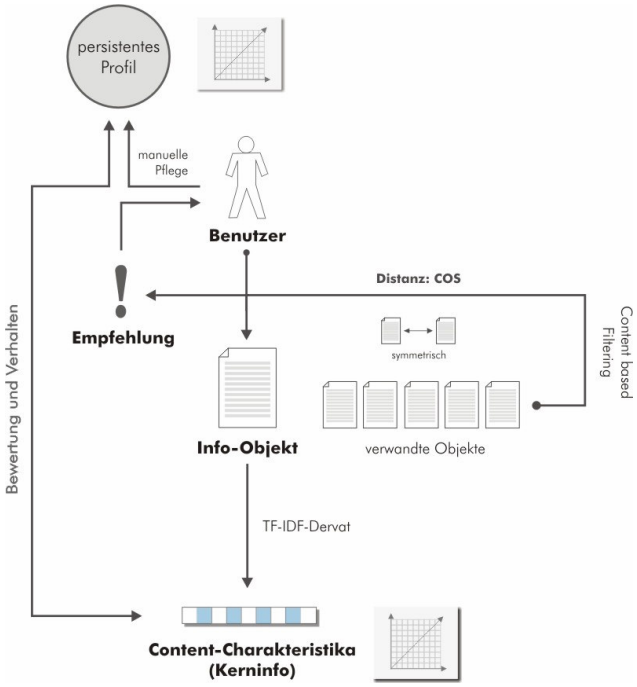


Abbildung 36: Amalthaea

In Amalthaea werden die Profile der Benutzer durch gewichtete Wortvektoren repräsentiert. Jeder Benutzer besitzt automatische und explizite Profile. Ein neues automatisches Profil wird durch die Beobachtung des Benutzerverhaltens in einer Session erzeugt (siehe *Merkmal: Flüchtiges* Profil*, Seite 20*). Die am stärksten frequentierten Webseiten werden dann mit einem TF-IDF-Derivat in Wortvektoren transformiert und zu einem Benutzerprofil zusammengefügt. Der Benutzer kann bestehende Profile (auch automatische) anpassen, indem er Webseiten entfernt, neue hinzufügt oder Worte des Wortvektors als besonders wichtig einstuft. Neue, auf Basis eines Profils empfohlene, Webseiten kann der Benutzer auf einer Skala von 1-5 bewerten und das Profil so beeinflussen. Die Empfehlungen werden mit dem Kosinus Ähnlichkeitsmaß ermittelt.

5.4.2.10 AgentDLS [CAR1998]

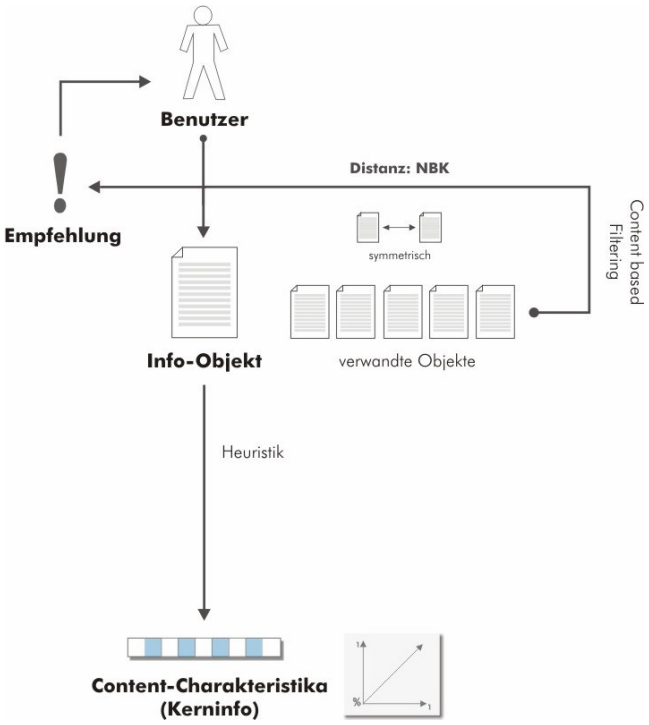


Abbildung 37: AgentDLS

Das CBF-Verfahren AgentDLS ermittelt Empfehlungen für Webseiten auf Basis folgender Heuristik:

(i) Schlüsselworte

Enthält ein Text explizite Schlüsselworte im Fließtext (erkannt aufgrund von Wort-Indikatoren wie "keyword" et cetera), so werden diese dem Text zugeordnet und für die Distanzermittlung mit dem gerade angezeigten Text verwendet. Eine Stoppwortliste sorgt dafür, dass zu allgemeine Schlüsselworte nicht verwendet werden.

(ii) Personen

Durch pattern-matching werden Namensnennungen ermittelt, die mit einem Personenverzeichnis (manuell aufgebaut) übereinstimmen. Als Empfehlung wird die Webseite der erkannten Person empfohlen.

(iii) Zitate

Analog zu "Personen" – allerdings für Veröffentlichungen.

(iv) Concept

Die Distanz zwischen dem angezeigten Text und anderen Texten - beziehungsweise deren Wortvektoren - wird mittels NBK ermittelt. Dazu kommt das Rainbow Package von McCullam [MCC1998] zum Einsatz, das mit binären Wortvektoren arbeitet.

Die Empfehlungen werden dem Benutzer bei der Anzeige des aktuellen Dokumentes angezeigt. Die Ermittlung und Anzeige der Empfehlungen erfolgt allerdings asynchron, um die Anzeige des Dokumentes nicht zu verlangsamen.

5.4.2.11 Webmate [CHE1998]

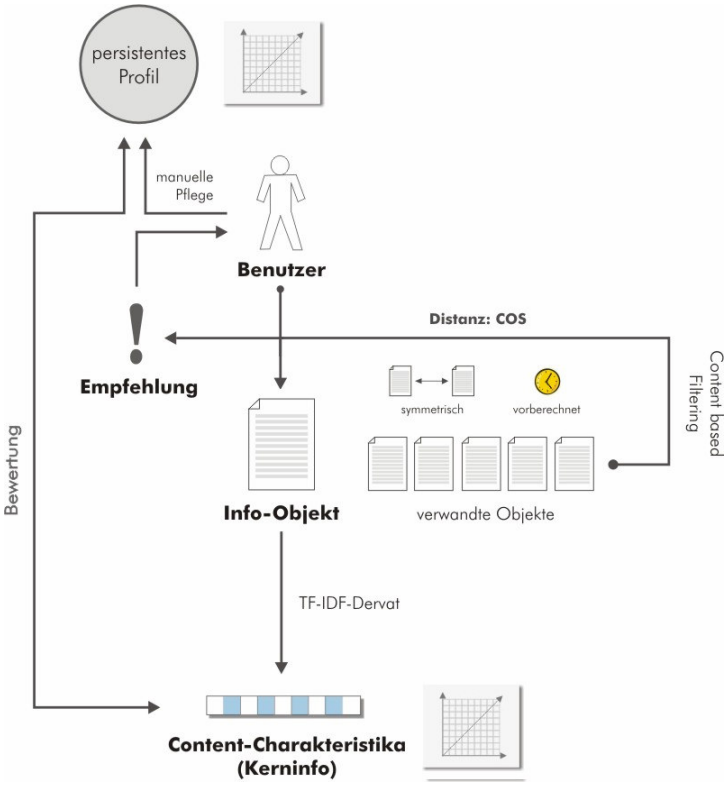


Abbildung 38: Webmate

Webmate verwendet Benutzerprofile, die aus gewichteten Wortvektoren bestehen. Diese Vektoren werden auf Basis eines TF-IDF-Derivates aus Texten mit positiven Benutzerbewertungen gewonnen. Die Zahl der Wortvektoren, die für "Interessen" eines Benutzers stehen, wird durch kontinuierliche Zusammenfassung von Vektoren mit großer Ähnlichkeit gering gehalten. Die Ähnlichkeit wird mit dem Kosinus Ähnlichkeitsmaß bestimmt. Die Wortvektoren des Benutzerprofils werden dann mit den ebenfalls gewichteten Wortvektoren neuer Webseiten auf Basis des Kosinus Ähnlichkeitsmaßes verglichen. Übersteigt die Ähnlichkeit der Vektoren einen Schwellwert, so werden die Webseiten dem Benutzer in Form einer "virtuellen Zeitung" mit absteigend nach Relevanz (Ähnlichkeit) sortierten Links empfohlen. Die Berechnung der Empfehlungen erfolgt "offline" (vorberechnet), wenn genügend Rechenleistung verfügbar ist.

5.4.2.12 SLIDER [BAL1998]

In SLIDER werden Webseiten und Benutzerprofile durch gewichtete Wortvektoren repräsentiert. Jeder Benutzer besitzt mehrere Profile. Der Benutzer kann Webseiten explizit Profilen zuordnen, innerhalb der Profile verschieben und auch wieder komplett entfernen. Ferner kann der Benutzer die vorgeschlagenen Webseiten bewerten (binäre Wertung durch "Star"-Auszeichnung). Das Lesen einer Webseite wird implizit bewertet. Auf Basis dieser Aktionen werden die Wortvektoren der Profile durch die Wortvektoren der Webseiten verändert. Gelöschte Webseiten werden als Ausschlusskriterien für die Empfehlungen verwendet, bleiben in SLIDER also als Information erhalten. Die Gewichtung (β) gestaltet sich dabei wie folgt:

- zuordnen: 3
- löschen: 3
- lesen: 0,5
- bewerten: 3
- keine Aktion: 0,25

Zu jedem Profil (*topic*) werden dem Benutzer die sechs relevantesten, noch nicht vom Benutzer gesehenen Webseiten angezeigt. Außerdem gibt es ein *topic* "other news", in dem der Benutzer bisher ungesehene Webseiten sieht, die mit keinem seiner Profile übereinstimmen. Dadurch soll ein "Tunnelblick" (*overspecialization*) vermieden werden.

Der Wortvektor eines Textes wird durch ein TF-IDF-Derivat gebildet. Die Relevanz eines Textes für einen bestimmten Benutzer wird durch *sim*, ein Derivat des Kosinus Ähnlichkeitsmaßes zwischen Text- und Profil-Vektor bestimmt. Ein neuer Text wird in Form seines Wortvektors wie folgt in das Profil übernommen:

$$P' = P + \beta D$$

Wobei P' das neue und P das alte Profil sowie D den Wortvektor des neuen Textes repräsentieren. Dabei steht β für die Gewichtung der Webseite (siehe oben).

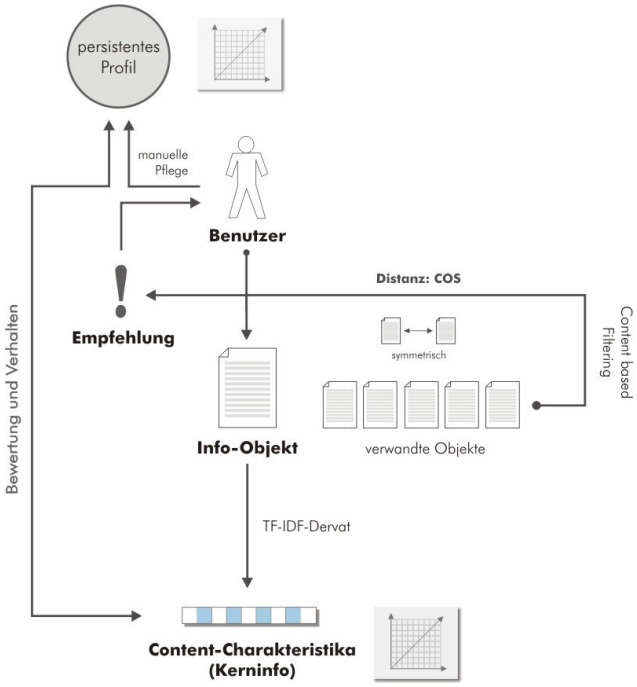


Abbildung 39: SLIDER

5.4.2.13 LexicalChainer [GRE1998]

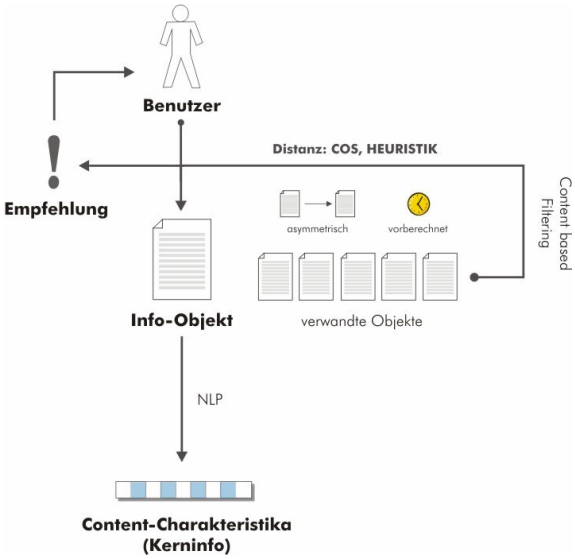


Abbildung 40: LexicalChainer

Das LexicalChainer Verfahren erzeugt Links zwischen Dokumenten, empfiehlt also zu einem Text andere Texte. Es verwendet "lexical chaining", wobei eine lexical chain (*LC*) eine Folge semantisch verbundener Worte ist. Die Intention ist es, Textfragmente zu ermitteln, die das gleiche Thema haben. Dadurch soll die Ambiguität, die bei Verwendung einzelner Worte entsteht, reduziert werden. Als Basis für die LC Bildung kann jedes

Verzeichnis verwendet werden, das Worte mit ihren Bedeutungen verbindet (NLP). Beim LexicalChainer kommt *WordNet* [FEL1998] zum Einsatz. Unter anderem gruppiert WordNet Worte nach Synonymgruppen (als "Synsets" bezeichnet), die noch mit Hyperonym/Hyponym-Beziehungen verknüpft sind. Worte, die über die Beziehungen in WordNet verbunden sind, bilden eine LC.

Jeder Text T wird durch zwei Vektoren v_1, v_2 repräsentiert:

$$v_1=(g_{11},\ldots,g_{1n})$$

mit g_{1i}= Gewicht des synsets i im Text = Anzahl des Vorkommens des Synsets in den LC des Textes

$$v_2=(g_{21},\ldots,g_{2n})$$

mit g_{2i}= Gewicht des synsets i im Text = Anzahl des Vorkommens des Synsets in WordNet-Relation (Hyperonym, Hyponym et cetera) zu einem Synset das in v_1 ein Gewicht größer "0" hat.

Die Distanz zweier Texte A und B wird durch das Kosinus Ähnlichkeitsmaß zwischen den Vektoren v_{A1}, v_{A2} von Text A und v_{B1}, v_{B2} von Text B mit folgender Heuristik ermittelt:

(i) $D_1=COS(v_{A1},v_{B1})$

(ii) $D_2=COS(v_{A1},v_{B2})$

(iii) $D_3=COS(v_{A2},v_{B1})$

Erreicht die Summe

$$D=\sum_{i=1\ldots3}D_i$$

dieser Distanzen einen bestimmten Schwellwert, so wird der Text dem Benutzer empfohlen.

5.4.2.14 NewsDude [BIL1999]

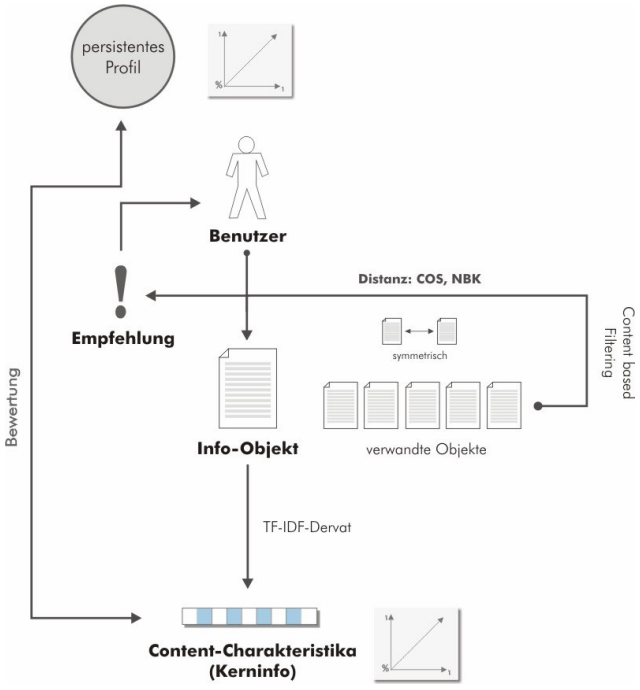


Abbildung 41: NewsDude

Das NewsDude Verfahren empfiehlt dem Benutzer auf Basis seines Profils neue Texte. Dabei werden ein aktuelles und ein langfristiges Profil unterschieden. Um Empfehlungen auszusprechen, werden Texte zunächst mit dem kurzfristigen Profil verglichen; erst wenn dies keine ausreichende Ähnlichkeit ergibt, kommt das langfristige Profil mit der Aufteilung in "interessant" oder "nicht interessant" zum Einsatz.

Das aktuelle Profil besteht aus den vom Benutzer bewerteten Texten. Die Texte werden als Wortvektoren im Profil abgelegt. Die Empfehlung neuer Texte erfolgt dann durch Transformation derselben in Wortvektoren und Bestimmung der Ähnlichkeit zum Profil auf Basis des Kosinus Ähnlichkeitsmaßes.

Das langfristige Profil beruht auf der Naiven Bayes-Klassifikation der booleschen Wortvektoren der Texte und besteht aus den Klassen "interessant" und "nicht interessant". Texte werden bei NewsDude nur dann klassifiziert, wenn diese mindestens `n` (konkret 3) Worte der betreffenden Klasse enthalten.

Eine Besonderheit von NewsDude ist, dass dem Benutzer ausgesprochene Empfehlungen erklärt werden. Wird ein Text auf Basis des aktuellen Profils empfohlen, so wird die Anmerkung ausgegeben, dass der Text empfohlen wurde, weil zuvor einmal der Text `t` positiv bewertet wurde. Dabei steht `t` für den Text aus dem aktuellen Profil mit dem größten Kosinus Ähnlichkeitsmaß. Wurde die Textempfehlung auf Basis des langfristigen Profils ermittelt, so gib NewsDude die Worte des Textes an, die maßgeblich zur Zuordnung des Textes zur Klasse der "interessanten" Texte geführt haben.

5.4.2.15 SIFT [YAN1999]

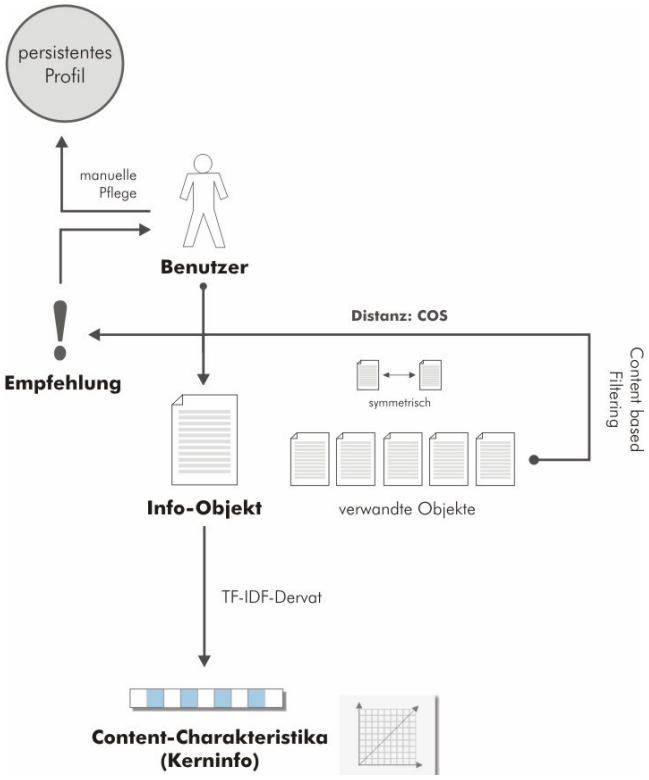


Abbildung 42: SIFT

In SIFT ("Stanford Information filtering tool") kann der Benutzer beliebig viele explizite Profile in Form von Anfragen (Queries) samt deren Parametern (Benachrichtigungsfrequenz, Anzahl der gewünschten Ergebnisse et cetera) anlegen. Mit booleschen Anfragen (gewünschte und nicht gewünschte Worte) und Vektorraum-Anfragen (VSM; vector space model; es wird eine Wordmenge als Anfrage verwendet) bestehen zwei verschiedene Anfragetypen. Bei Vektorraum-Anfragen kommt das TF-IDF-Verfahren zur Erzeugung der gewichteten Wortvektoren zum Einsatz. Die Ähnlichkeiten werden dann mittels Kosinus Ähnlichkeitsmaß ermittelt.

Bei der Anfrageverarbeitung werden drei Ansätze zur Bearbeitung vorgestellt. Beim BF-Ansatz (brute force) wird für jedes neue Dokument jede Anfrage erneut ausgeführt, um zu entscheiden, ob dem Benutzer das Objekt empfohlen werden soll. Der QI-Ansatz (Query Indexing) reduziert die Anzahl der auszuführenden Anfragen, indem nur solche ausgeführt werden, die mindestens ein Wort des neuen Dokumentes enthalten. Dazu wird ein inverser Index mit den Worten der Anfragen verwaltet, in dem von jedem Wort die Anfragen, in denen es verwendet wird, erreichbar sind. Der SQI-Ansatz (Selective Query Indexing) reduziert zusätzlich noch die im inversen Index geführten Worte einer Anfrage, indem nur die signifikanten Worte der Anfrage verwendet werden. Die Selektion der signifikanten Worte erfolgt auf Basis eines IDF-Derivates.

5.4.2.16 Watson [BUD1999],[BUD2000],[BUD2001]

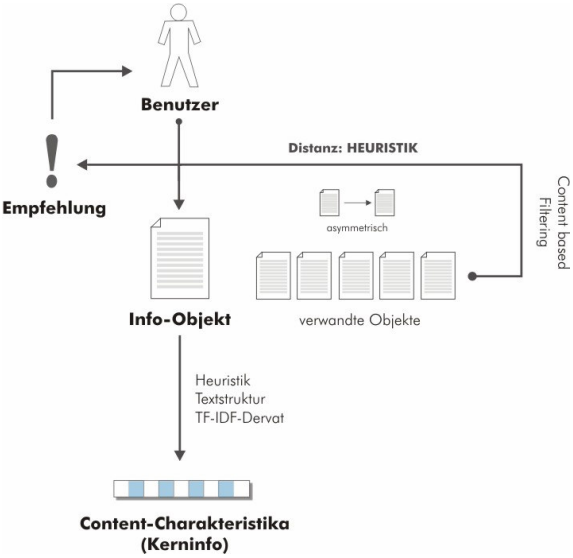


Abbildung 43: Watson

Das WATSON Verfahren liefert dem Benutzer zu seinem in einer Applikation (Textverarbeitung, Browser et cetera) angezeigten Text inhaltlich verwandte Texte. Dazu verwendet WATSON Plug-Ins (*application adapters*), um Zugriff auf den in einer Applikation angezeigten Text zu haben.

Aufgrund des Textes wird dann, wie im Folgenden beschrieben, eine Suchanfrage erzeugt. Da WATSON die Texte, gegen die die Suchanfragen laufen sollen, nicht in der eigenen Datenbasis vorhält und auf diese in der Regel nur in Form eines unstrukturierten Textes Zugriff hat, kann kein vektorbasiertes Verfahren zum Einsatz kommen. Stattdessen verwendet WATSON ein Bündel von Heuristiken, um aus einem Text `T` die für die Suchanfrage relevanten Worte in Form einer sortierten Liste zu selektieren. Diese Heuristiken sind:

- Entfernung der Stoppworte; dabei verlässt sich WATSON allerdings auch auf die Quellsysteme (siehe unten); das heißt, WATSON entfernt nur die Stoppworte aus der Suchanfrage, die in einer statischen Stoppwortliste definiert wurden
- Je häufiger ein Wort vorkommt, desto wichtiger ist es (TF-IDF-Derivat)
- Hervorgehobene Worte (fett et cetera) sind wichtiger
- Worte am Anfang eines Textes sind wichtiger, als jene am Ende des Textes (Textstruktur)
- Worte, die in kleinerer Schrift gehalten sind, sind weniger wichtig
- Die Textposition (siehe oben) für Worte in Auflistungen wird ignoriert

Diese Suchanfrage wird dann in einer standardisierten internen, formalen Form an die Schnittstellen (*information adapters*) der verschiedenen Quellsysteme gesendet. Diese Schnittstellen transformieren das interne Format in das proprietäre Anfrageformat des jeweiligen Quellsystems. Bei WATSON werden alle Anfragen "online" (keine Vorberechnung) durchgeführt. WATSON hält selbst keine Informationen und ist damit auf die Verfügbarkeit der Quellsysteme angewiesen.

Die Anfragen liefern die Ergebnisse, die dann von WATSON auf Redundanz geprüft werden. Dazu vergleicht WATSON den Titel und die URL der Ergebnisse und betrachtet zwei Texte als identisch, wenn die Ähnlichkeit von Titel und URL einen bestimmten Schwellwert überschreitet.

5.4.2.17 LIBRA [MOO2000]

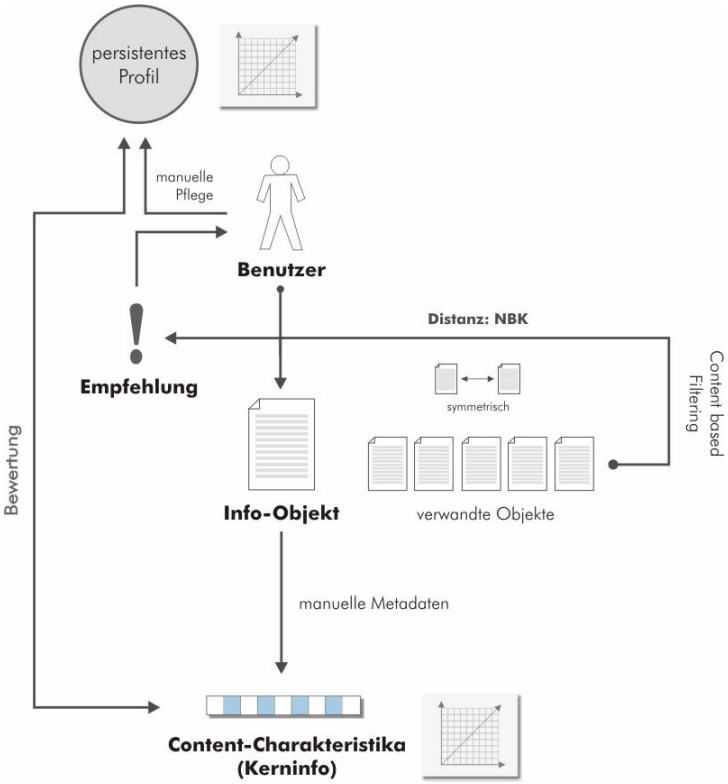


Abbildung 44: LIBRA

Beim LIBRA Verfahren fußen die Empfehlungen auf Metadaten, die von Amazon.com durch einen automatischen Extraktor (zu LIBRA Zeiten waren die Amazon-Webservices noch nicht verfügbar) gewonnen werden. Die Metadaten, zu denen auch die Collaborative-Daten (ähnliche Bücher) von Amazon.com zählen, die aber in LIBRA als "normaler" Content verwendet werden, werden komplett (kein TF-IDF) als gewichteter Wortvektor implementiert.

Die Benutzer müssen zunächst eine Reihe von Büchern auf einer Skala von 1 bis 10 bewerten, um ein Profil zu bilden. Das Profil besteht aus einem gewichteten Wortvektor. Zur Klassifikation kommt ein leicht modifizierter Naiver Bayes-Klassifikator zum Einsatz.

5.4.2.18 Margin Notes [RHO2000]

In Margin Notes werden zunächst Webseiten mittels *Savant*, einem TF-IDF-Derivat, in Wortvektoren transformiert. Dies erfolgt "offline" (vorberechnet), sobald die Webseiten verfügbar sind. Ruft der Benutzer im Browser eine Webseite auf, so werden "online" (beim angezeigten Text wird also nicht auf die Vorberechnung zurückgegriffen) mit *Savant* - einer von Rhodes und Jan Nelson entwickelten Suchmaschine, die mit dem *Okapi* Verfahren [ROB1992], [WAL1998] arbeitet - die ähnlichsten Texte bestimmt. Dabei wird den Worten im ersten Absatz des Textes mehr Relevanz zugesprochen (Textstruktur). Diese werden dann in Form der 5 "relevantesten" Texte am rechten Rand (daher "margin notes") in der Webseite eingebettet angezeigt (durch Rewriting des html-Codes auf einem Proxy-Server). Zu Evaluationszwecken wird außerdem eine Bewertung auf einer Skala von 1-5 ermöglicht, die aber nicht in die zukünftigen Empfehlungen einfließt.

Das Okapi Verfahren (in der Fassung von TREC-6, wie es von Margin Notes verwendet wird) basiert auf einem TF-IDF-Derivat, das statt der IDF die durchschnittliche Worthäufigkeit im Text selbst zur Normalisierung heranzieht. Kürzere Dokumente werden bei Empfehlungen längeren Dokumenten vorgezogen.

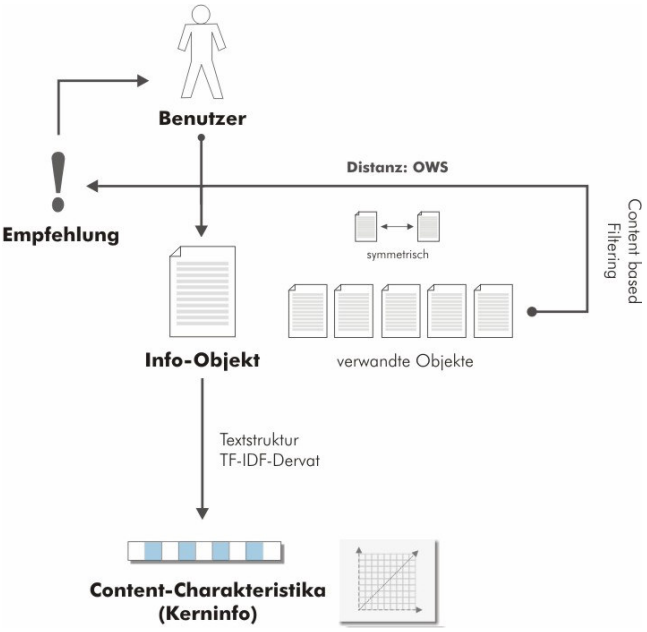


Abbildung 45: Margin Notes

5.4.2.19 Jimminy [RHO2000a]

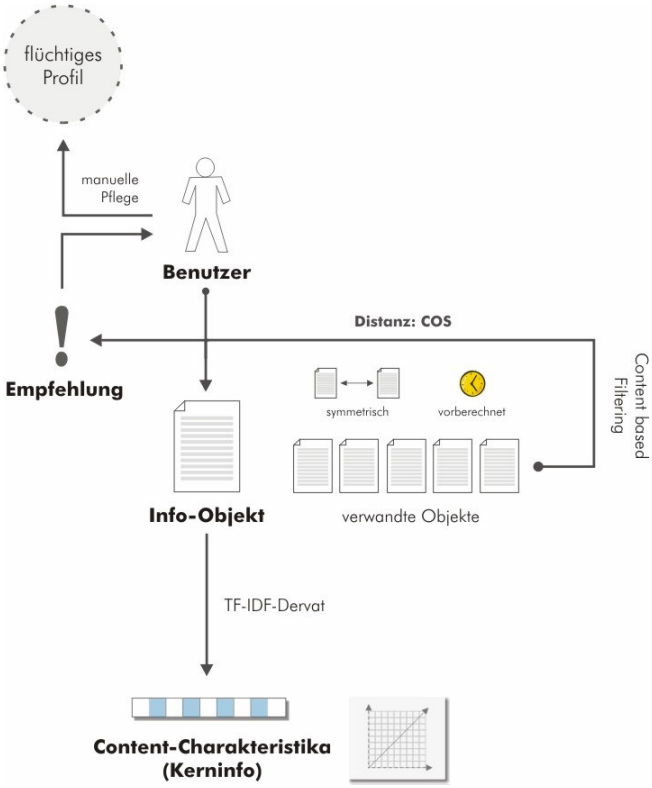


Abbildung 46: Jimminy

Im Paper *Just-in-time information retrieval agents* [RHO2000a] stellt Rhodes neben Margin Notes (siehe 5.4.2.18) und dem Remembrance Agent (siehe 5.4.2.7) mit Jimminy ein weiteres CBF Verfahren vor. Jimminy ist allerdings bezüglich des CBF Konzeptes identisch zum Remembrance Agent. Lediglich die Benutzerschnittstelle (tragbare Sichtbrille und Tastatur) unterscheidet sich. Das Verfahren ist dennoch interessant, da es einen Ausblick auf die potenziellen Möglichkeiten kontextbasierter Empfehlungen im "echten Leben" gibt. Mehr Details zum Jimminy Ansatz findet man in Rhodes Patentschrift dazu [RHO2001].

5.4.2.20 SUITOR [MAG2000]

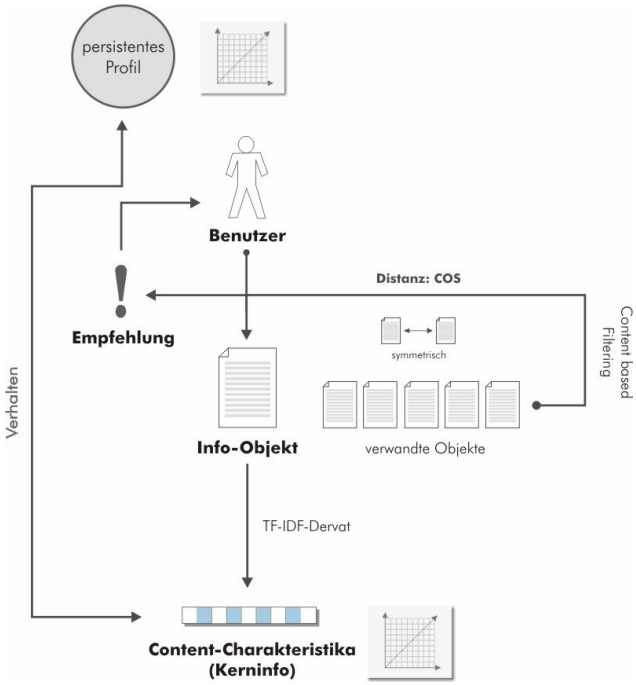


Abbildung 47: SUITOR

Die Besonderheit des CBF Verfahrens SUITOR besteht darin, dass es einen sehr umfassenden Zugang zum aktuellen Kontext des Benutzers sucht. So protokolliert es nicht nur klassisch sein Verhalten im Web-Browser, sondern überwacht auch seine Tastatureingaben und seinen Blickfokus durch eine entsprechende Hardware. Alle Analysemethoden liefern Text. Dieser wird kontinuierlich in das Benutzerprofil übernommen. Und zwar in Form eines TF-IDF-Derivates, das bedeutende Worte durch deren Häufigkeit selektiert. Verändert sich das Interesse eines Benutzers, wird langsam - indem mehr und mehr Worte mit dem neuen Thema des Interesses aufgenommen werden - auch das Profil verändert. Damit Interessensänderungen schneller berücksichtigt werden können, partitioniert SUITOR das Benutzerprofil auf der Zeitachse in zwei Teile: kurzfristiges Profil und langfristiges Profil. Über die Implementation der Profile berichtet Maglio leider nicht. Aus dem Zusammenhang lässt sich aber auf eine vektorbasierte Darstellung schließen.

SUITOR verwendet verschiedene *Agenten*. Die potenziell zu empfehlenden Texte werden durch *investigator agents* gesammelt. So genannte *reflector agents* bewerten dann die Texte im Hinblick auf die Benutzerprofile. Ab einer bestimmten Überdeckung (vermutlich TF-IDF-Derivat mit Kosinus Ähnlichkeitsmaß; siehe "vektorbasiert" oben) von Text und kurzfristigem oder langfristigem Profil wird eine Empfehlung ausgesprochen.

5.4.2.21 PRES [MET2000]

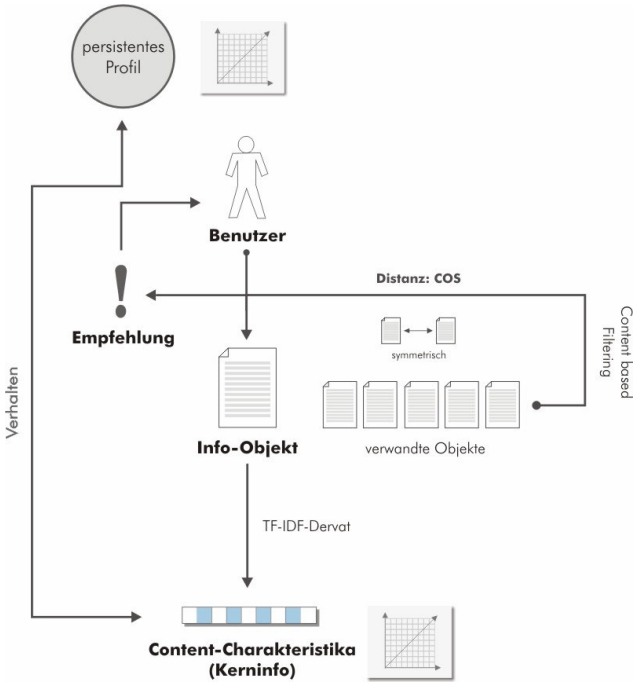


Abbildung 48: PRES

In PRES werden Webseiten und Benutzerprofile durch gewichtete Wortvektoren repräsentiert. Jeder Benutzer besitzt ein Profil, das implizit durch das Betrachten von Webseiten (mit einer Mindestlesedauer) gebildet wird. Der Wortvektor eines Textes wird durch ein TF-IDF-Derivat gebildet. Die Relevanz eines Textes für einen bestimmten Benutzer wird durch das Kosinus Ähnlichkeitsmaß zwischen Text- und Profil-Vektor bestimmt. Ein neuer Text wird in Form seines Wortvektors wie folgt in das Profil übernommen:

$$P' = \alpha P + D$$

Wobei P' das neue und P das alte Profil sowie D den Wortvektor des neuen Textes repräsentieren. Durch α (ein Wert zwischen 0 und 1) werden die Gewichte der bestehenden Worte im bestehenden Profil mit jedem neuen Text verringert. Dadurch "verblassen" nicht mehr frequentierte Worte nach und nach.

5.4.2.22 WebSail [CHE2000]

Das WebSail Verfahren basiert auf Profilen, die aus einem gewichteten Wortvektor bestehen. Vom Benutzer bewertete Webseiten verändern das Profil in Form binärer Wortvektoren, die durch ein TF-IDF-Derivat gebildet werden. Die Bewertung der Webseiten durch den Benutzer erfolgt zweiwertig in Form von "relevant (ja/nein)". Angezeigt werden die zu bewertenden Webseiten bei WebSail allerdings nicht im Kontext anderer Webseiten, sondern in Form eines Suchergebnisses, das dann mit WebSail iterativ durch Bewertungen verfeinert wird. Die Profile sind daher flüchtig und nur während einer Such-Iterationsschleife präsent. Auf Basis des *TW2* [CHE2000][LIT1987] Algorithmus wird ein Wort p_v im Wortvektor des Profils durch ein Wort w_V im Wortvektor einer positiv bewerteten Webseite (deren Wortvektor mit einem TF-IDF-Derivat ermittelt wird) wie folgt beeinflusst (wobei α für den *Motivations*- beziehungsweise *Demotivationsfaktor* mit $\alpha>1$ steht):

- $p_v = w_v$, falls $w_v=0$
- $p_v = \alpha$, falls $w_v=1$ und $p_v=0$
- $p_v = \alpha * p_v$, falls $w_v=1$ und $p_v>0$

Bei einer negativ bewerteten Webseite gilt:

- $p_v = p_v/\alpha$

Es wird also im Falle einer negativ bewerteten Webseite eine Abwertung des kompletten Profils vorgenommen. Die Empfehlung neuer Webseiten basiert auf dem ebenfalls im Rahmen von *TW2* vorgestellten Distanzmaß. Es handelt sich um das Skalarprodukt des Profil- und Webseiten-Wortvektors und damit um eine Abwandlung des Kosinus Ähnlichkeitsmaßes.

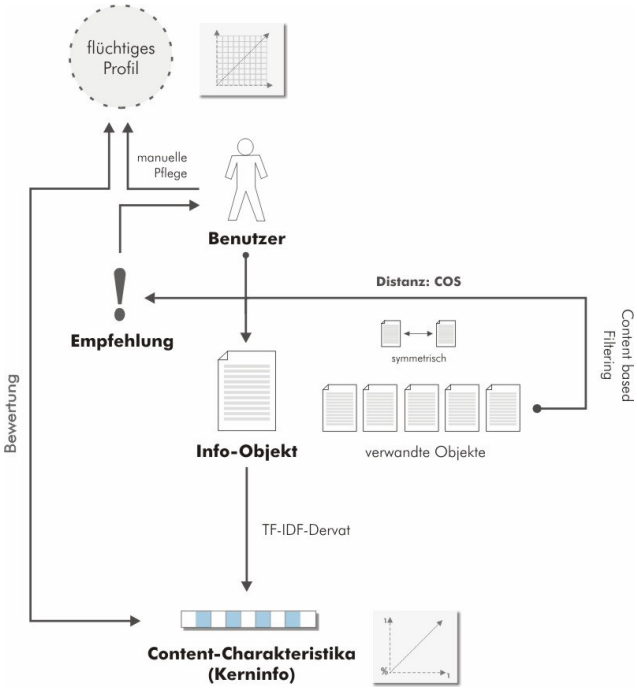


Abbildung 49: WebSail

5.4.2.23 WAIR [SEO2000],[ZHA2001]

Das WAIR Verfahren nutzt ein Benutzerprofil in Form eines gewichteten Wortvektors. Zu Beginn gibt der Benutzer n Worte in Form einer Suchanfrage ein und schafft somit ein implizites Basisprofil. Aufgrund der Worte im Benutzerprofil werden Anfragen an Standardsuchmaschinen gestellt. Die zurück gelieferten Webseiten transformiert das Verfahren dann mit einem TF-IDF-Derivat (ohne IDF-Komponente, da WAIR keinen eigenen Index aufbaut) in binäre Wortvektoren. Die Relevanz einer Webseite wird wie folgt berechnet:

$R_T=\Sigma_{k=1\ldots n}\ tf_k*w_{pk}$, falls $k \in T$

wobei $k=1\ldots n$ die Worte im Profilvektor, $p=1\ldots m$ die Profile des Benutzer und T der Text der Webseite sind. Die n relevantesten Webseiten (Maximalwerte R_t der in Frage kommenden Webseiten) werden dem Benutzer dann vorgeschlagen.

Aufgrund des Verhaltens des Benutzers in Bezug auf die empfohlenen Webseiten wird das Benutzerprofil adaptiert. Dabei werden pro Webseite protokolliert:

- Lesedauer
- Setzen eines Bookmarks
- Scrollen
- Benutzen von Hyperlinks (in der Webseite)

Diese vier Parameter werden mit einem mehrstufigen neuronalen Netz gewichtet. Ein Profil wird in Abhängigkeit vom Wert dieser vier gewichteten Parameter angepasst. Dabei wird jedes Wort des Profilvektors erhöht, wenn die empfohlene Webseite es enthält und erniedrigt, wenn es nicht enthalten ist. Der so angepasste gewichtete Profilvektor wird verwendet, um m neue Worte für eine Suchanfrage zu selektieren. Dabei sind $\varepsilon - \varepsilon*m$ Worte solche mit den größten Gewichten und $\varepsilon*m$ Worte zufällige aus dem Profil. Letzteres soll die Flexibilität ("exploration") der Profile für Neues gewährleisten.

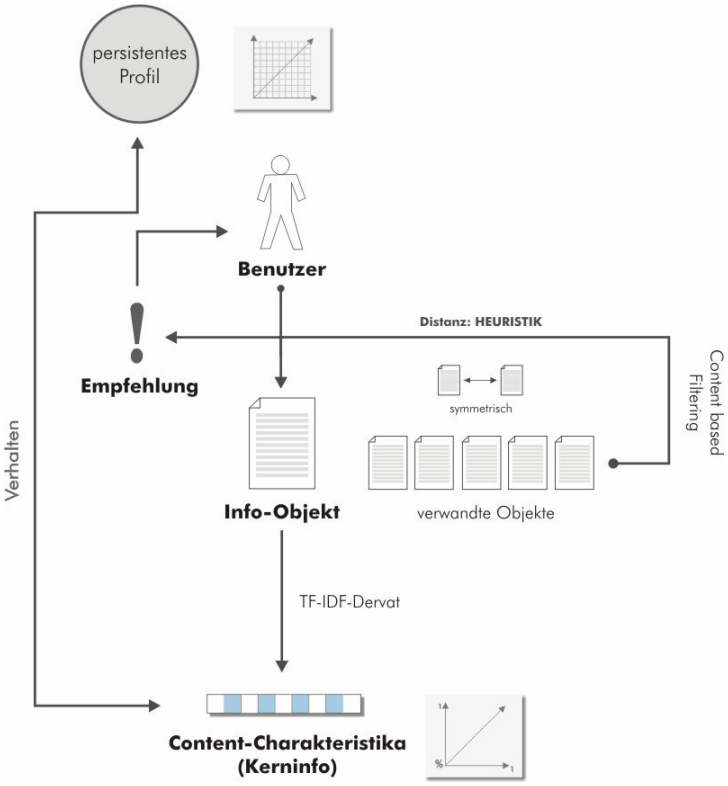


Abbildung 50: WAIR

5.4.2.24 Powerscout [LIE2001]

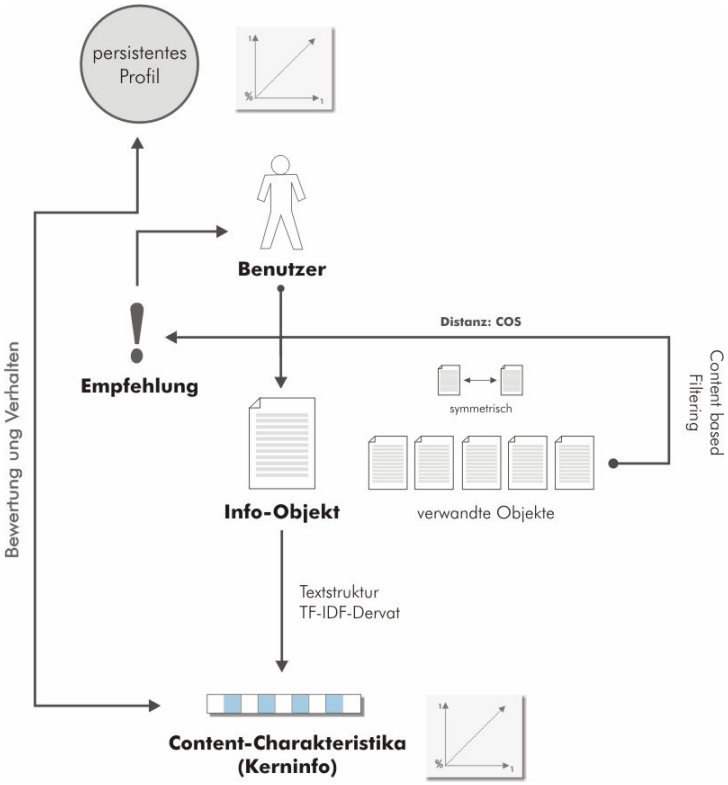


Abbildung 51: Powerscout

Das Powerscout Verfahren ist eine Weiterentwicklung von Letizia (siehe 5.4.2.4). Statt einem Profil werden nun beliebig viele Profile angeboten. Ein Profil besteht weiterhin aus einem Wortvektor, in dem allerdings zu jedem Wort zusätzlich hinterlegt ist, weshalb es sich im Profil befindet. Gründe dafür können sein:

- betrachtete Webseite, aus der das Wort ins Profil übernommen wurde
- Suchanfrage, die das Wort enthielt

Für die Selektion der Schlüsselworte eines Texte wird neben einem TF-IDF-Derivat auch die Typographie und die Textposition ausgewertet. Der Benutzer entscheidet dann manuell, ob und zu welchem Profil die Schlüsselworte des gerade betrachteten Texts hinzugefügt werden sollen (Bewertung).

Powerscout analysiert nicht mehr die von der aktuellen Webseite aus verlinkten Webseiten, sondern nutzt die Worte eines Profils als Anfrage an eine herkömmliche Suchmaschine. Die eingehenden Webseiten werden in Wortvektoren transformiert. Wenn diese zum zugehörigen Profil (aufgrund dessen die Anfrage gestellt wurde) nach dem Kosinus Ähnlichkeitsmaß eine bestimmte Nähe aufweisen, werden sie – jeweils unter "ihrem" Profil – empfohlen.

5.4.2.25 CALVIN [BAU2001],[BAU2002],[LEA2000]

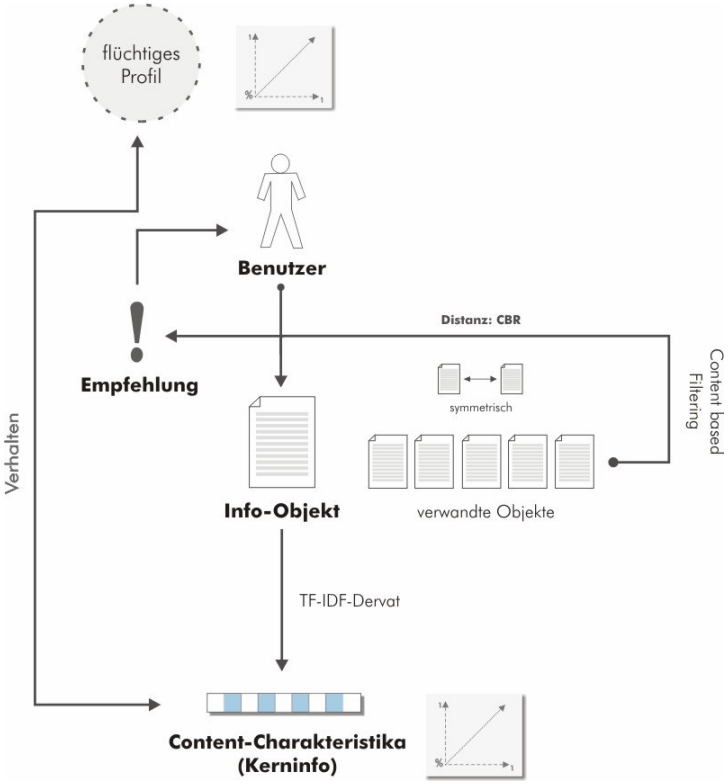


Abbildung 52: CALVIN

Das CALVIN Verfahren arbeitet mit fallbasiertem Schließen (case-based reasoning), indem es die Verwendung von Webseiten in Kontexten protokolliert und auf Basis dieser Fälle später, in ähnlichen Kontexten, diese Webseiten empfiehlt. Der Kontext besteht bei CALVIN aus drei Bestandteilen: dem *Task* (eine vom Benutzer manuell angegebene Aufgabe in Form von Worten), mehreren *Topics* (ebenfalls vom Benutzer manuell in Form von Worten angegeben) und schließlich den Webseiten (in Form automatisch auf Basis der Texte ermittelter Schlüsselworte), die im Rahmen von *Task* und *Topics* betrachtet wurden. Die Webseiten werden dabei durch ein TF-IDF-Derivat in einen Wortvektor mit maximaler Wortanzahl (einmal definiert) transformiert.

Die Fälle (Kontexte) werden zentral gespeichert und benutzerübergreifend verfügbar gemacht. Eine CALVIN-Sitzung muss nicht zwingend mit der manuellen Angabe von *Task* und *Topic* beginnen, da auch alleine durch Webseiten ein Kontext aufgebaut wird, der mit der Fall-Basis verglichen werden kann. Erkennt CALVIN einen ähnlichen Kontext, so empfiehlt es nicht nur die Webseiten, die im gespeicherten Fall (Kontext) zusätzlich betrachtet wurden, sondern zeigt auch den Weg (Clickstream; Folge von Webseiten) zu dieser empfohlenen Webseite. Ähnliche Kontexte werden mit dem *Contrast Model of categorization* [TVE1977] ermittelt. Die Profile der Benutzer sind flüchtig, die Fälle werden neutral als Fälle verwaltet.

5.4.2.26 WordSieve [BAU2001a]

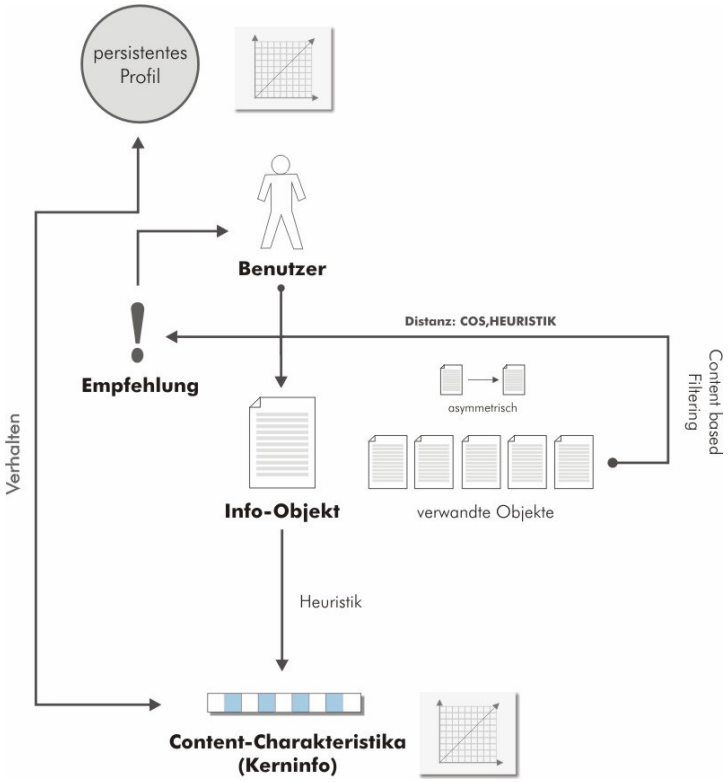


Abbildung 53: WordSieve

Das Grundprinzip von Wordsieve besteht aus einem dreistufigen "Sieb" (daher Word*Sieve*), durch das die Worte aller Texte, die der Nutzer betrachtet, geschickt werden.

Die erste Stufe bildet die prominentesten Worte im aktuellen Benutzerkontext ab. Dazu wird eine überschaubare Zahl an Knoten (konkret 150) mit den Werten "Wort" und "Prominenz" (sinngemäß "*excitement*") belegt. Mit jeweils 100 neu geprüften Worten verblasst (sinngemäß "*decay*") die Prominenz aller Knoten um den Wert "`b`". Neue Worte belegen solange die Knoten, bis alle belegt sind. Danach wird bei Worten, die bereits einen Knoten belegen, deren Prominenz um einen konstanten Wert erhöht (`b/g`). Einem neuen Wort, das noch keinen Knoten besitzt, wird dann zufällig ein Knoten zugeordnet. Es übernimmt diesen dauerhaft mit der Zufallswahrscheinlichkeit von $0,0001*(Prominenz-100)^2$. Dabei steht Prominenz für die des Wortes, das den Knoten derzeit belegt. Worte mit großer Prominenz werden daher mit sehr geringer Wahrscheinlichkeit ersetzt. Da die Einstellung des Parameters `g` für die erste Stufe des Siebs essentiell ist, wird es mit jedem neuen Wort, neu im Verhältnis zu den Knoten mit kumulierter Prominenz kalibriert.

Die zweite Stufe des Siebs besteht aus mehr Knoten (konkret 500), die neben dem Wort und der Prominenz auch noch eine "Aktivierungsstärke" (sinngemäß "*priming*") besitzen. In diese zweite Stufe gelangen nur Worte, die bereits Knoten in Stufe 1 belegen. Mit jedem neuen Wort auf Stufe 1 wird ein kompletter "Lauf" für die Worte der Stufe 2 durchgeführt. Mit jedem "Lauf" werden die Prominenz und Aktivierungsstärke aller Knoten um einen festen Wert abgesenkt. Die Prominenz der Knoten auf Stufe 2 wächst mit jedem Lauf in dem das Wort einen Knoten auf Stufe 1 besetzt hält. Das Wachstum ist dabei abhängig von der Aktivierungsstärke. Letztere wird dabei ebenfalls angehoben. Neue Worte ersetzen bestehende in Stufe 2 analog zum Verfahren auf Stufe 1. Die Stufe zwei bildet im Gegensatz zum "Kurzzeitgedächtnis" der Stufe 1 quasi ein "Langzeitgedächtnis".

Die dritte Stufe ist analog zur zweiten aufgebaut. Allerdings erhöht sich hier die Prominenz erst dann, wenn ein Wort lange Zeit nicht mehr in Stufe 1 existiert. Da klassische Stoppworte quasi immer existent sind, erreichen diese auf dieser Stufe keine hohe Prominenz.

Der gewichtete Wortvektor eines Textes wird mit WordSieve ermittelt, indem das Dokument durch ein leeres Sieb der Stufe 1 läuft und die Prominenz der Stufe 1 dann mit den korrespondieren Werten der Knoten in Stufe 2 und 3 multipliziert wird (Heuristik). Der aktuelle Kontext des Benutzers wird als gewichteter Wortvektor ebenfalls durch Multiplikation der Prominenz der Stufe 1 mit den korrespondieren Werten der Stufen 2 und 3 ermittelt. Die Ähnlichkeit eines Textes zu einem Profil wird dann mittels des Kosinus Ähnlichkeitsmaß bestimmt.

5.4.2.27 MetaMarker [PAI2001]

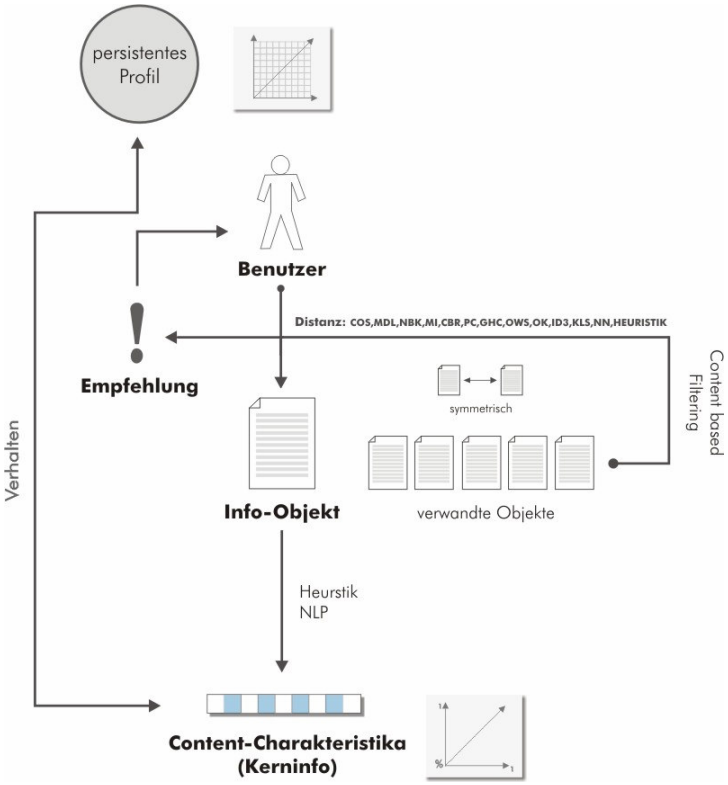


Abbildung 54: MetaMarker

Das MetaMarker Verfahren arbeitet auf E-Mails. Es erzeugt mittels NLP automatisiert Metadaten. Dabei kommt folgende Heuristik zum Einsatz, die den Text wie angegeben "tagged":

(i) Satz-Ermittlung (`<s> … </s>`)

(ii) Satzteil-Ermittlung (`"laufen"|Verb`)

(iii) Morphologie-Analyse (`"laufen"|Verb|Wortwurzel="lauf"`)

(iv) Mehrwort-Begriffe (`<pn>Müller,GmbH</pn>="Müller GmbH"`)

(v) Begriffskategorisierung (`<pn cat=Firma> Müller,GmbH</pn>`)

(vi) Implizite Metadaten (beispielsweise "`<urgency>`" aus E-Mail-Status)

(vii) Benutzerpräferenz (`<like>,<dislike>,<interested>,<not interested>`)

Die Benutzerpräferenz wird anhand von Schlüsselworten in die vier Werte "like", "dislike", "interested" beziehungsweise "not interested" transformiert und dann ins Benutzerprofil übernommen. Dem Benutzer werden dann Texte empfohlen, die Begriffe enthalten, die ihn interessieren oder die er gerne hat, und die keine (oder verhältnismäßig wenig) Worte enthalten, die ihn nicht interessieren oder die er nicht mag ("dislike"). Die Einschätzung erfolgt durch Klassifikation (vermutlich NBK) des Wortvektors eines Textes in Bezug auf den Klassenvektor (Benutzerprofil).

5.4.2.28 WebTop [WOL2002 und WOL2004]

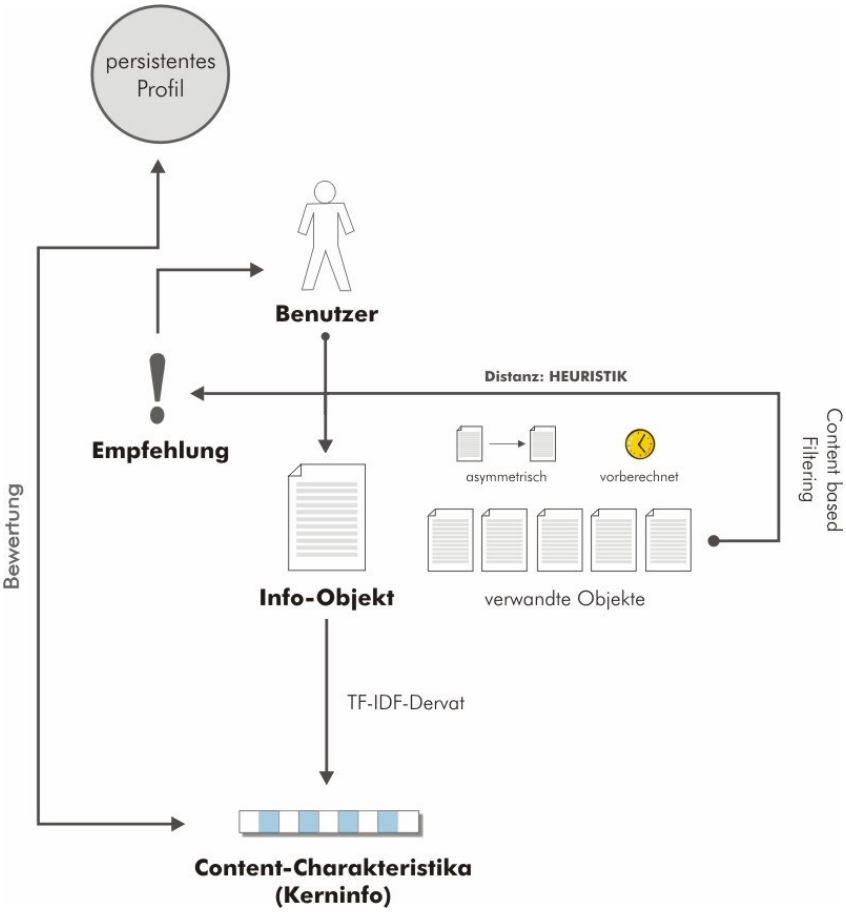


Abbildung 55: WebTop

Webtop empfiehlt dem Benutzer ähnliche Webseiten auf Basis der gerade angezeigten Webseite. Dabei werden die bedeutenden Worte der aktuell angezeigten Webseite durch ein TF-IDF-Derivat ermittelt. Diese Wortmenge wird dann als Anfrage für bestehende Suchmaschinen verwendet. Die zurück gelieferten Texte werden in Form von (Titel,Link)-Tupeln angeboten (heuristisches Verfahren zur Bestimmung der Content-Distanz). Der Benutzer kann Empfehlungen als "edge links" markieren (explizite Profilbildung), die dann dauerhaft - aber entfernbar - mit der angezeigten Webseite in seinem Profil verbunden bleiben. Neben den *edge links* und den verwandten Webseiten einer Webseite werden dem Benutzer noch die eingehenden und ausgehenden Hyperlinks der Webseite angezeigt.

5.4.2.29 INFOS [MOC1996]

Das INFOS Verfahren arbeitet auf Newsgroup-Beiträgen. Der Benutzer kann Beiträge als interessant, uninteressant und indifferent bewerten. Auf Basis dieser Bewertungen kommen zwei Verfahren zum Einsatz. Das global hill climbing (GHC) Modell greift NBK und TF-IDF auf und entwickelt daraus ein eigenes Verfahren. Es speichert zu jedem Wort die Anzahl der seitens des Benutzers als interessant (`I(w)`) und als uninteressant (`U(w)`) bewerteten Texte. Für einen neuen Text `T` wird dann die Summe der Bewertungen der anschließend darin enthaltenen Worte gebildet und dann entschieden, ob eine Empfehlung ausgesprochen wird:

$$\frac{\sum_{i=1...|T|} I(w_i)}{\sum_{i=1...|T|} O(w_i)} - \frac{\sum_{i=1...|T|} U(w_i)}{\sum_{i=1...|T|} O(w_i)} > A \Rightarrow empfehlen$$

mit `O(w)` als der Gesamtzahl der Instanzen von `w` in allen Texten und A als Pufferzone (auf 0,15 gesetzt) zwischen *empfehlen* und *nicht empfehlen*.

Wenn das GHC einen Beitrag nicht klassifizieren kann, wird das Verfahren der case based abstraction hierarchy (case based reasoning; CBR) verwendet. Es ist ein Modell, das die "Bedeutung" der Worte von Beiträgen für die Klassifikation verwendet. Dazu werden die "bedeutenden" Worte der Texte als Schlüsselworte extrahiert. Dafür kommen einige Heuristiken zum Einsatz. So werden beispielsweise die ersten beiden und der letzte Satz eines Absatzes sowie die Überschriften höher bewertet (Textstruktur). Und vorangegangene Signalworte wie "*wichtig(es)*" erhöhen die Relevanz eines Wortes. Mittels WordNet (http://wordnet.princeton.edu; Basiskorpus) werden auch die Synonyme eines Wortes ermittelt. Auf Basis dieser Heuristiken werden dann die bedeutenden Worte eines Textes selektiert und anschließend mit den bedeutenden Worten der bereits abgelegten "cases" (bereits bewertete Beiträge) verglichen. Dabei wird auch auf die Hyperonym-Struktur von WordNet zurückgegriffen, um die Worte begrifflich einzuordnen. Schließlich werden die neuen Beiträge empfohlen, wenn sie eine hohe Übereinstimmung mit positiv bewerteten Beiträgen besitzen. Es werden dem Benutzer auch die bestimmenden Eigenschaften (Worte) angezeigt, die zur Empfehlung geführt haben.

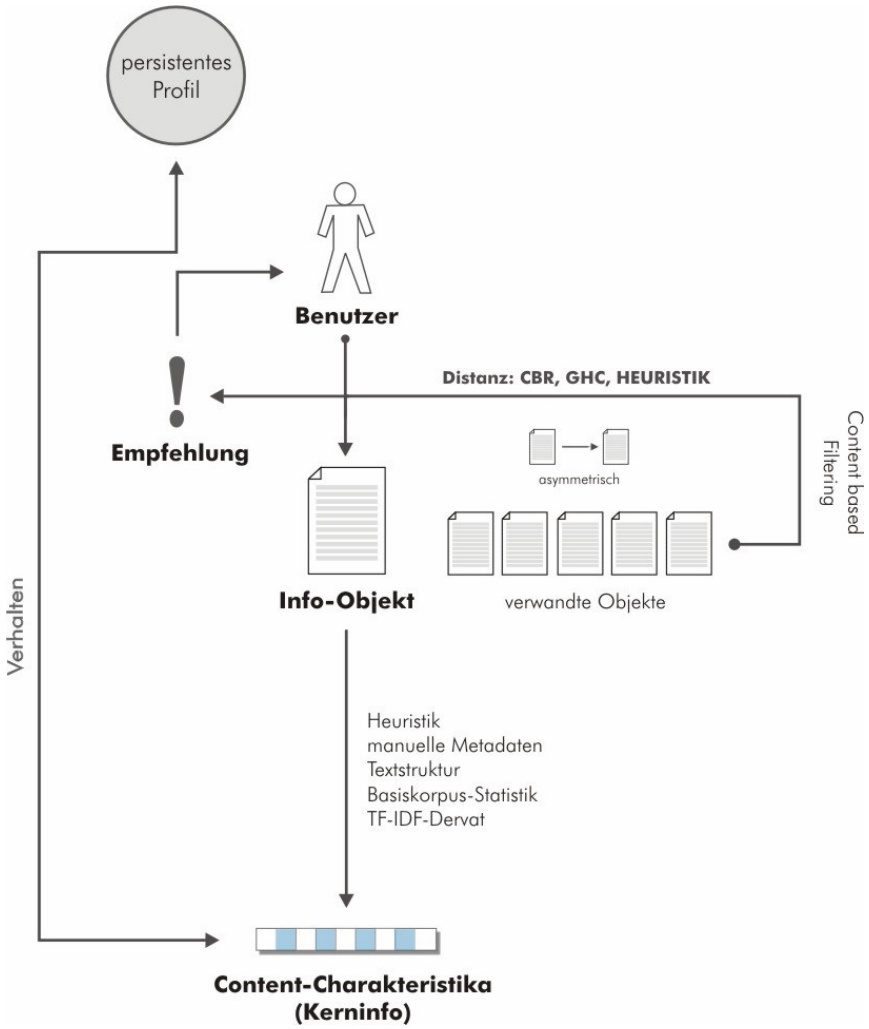


Abbildung 56: INFOS

5.4.3 Hybrid-Systeme

5.4.3.1 Fab [BAL1997]

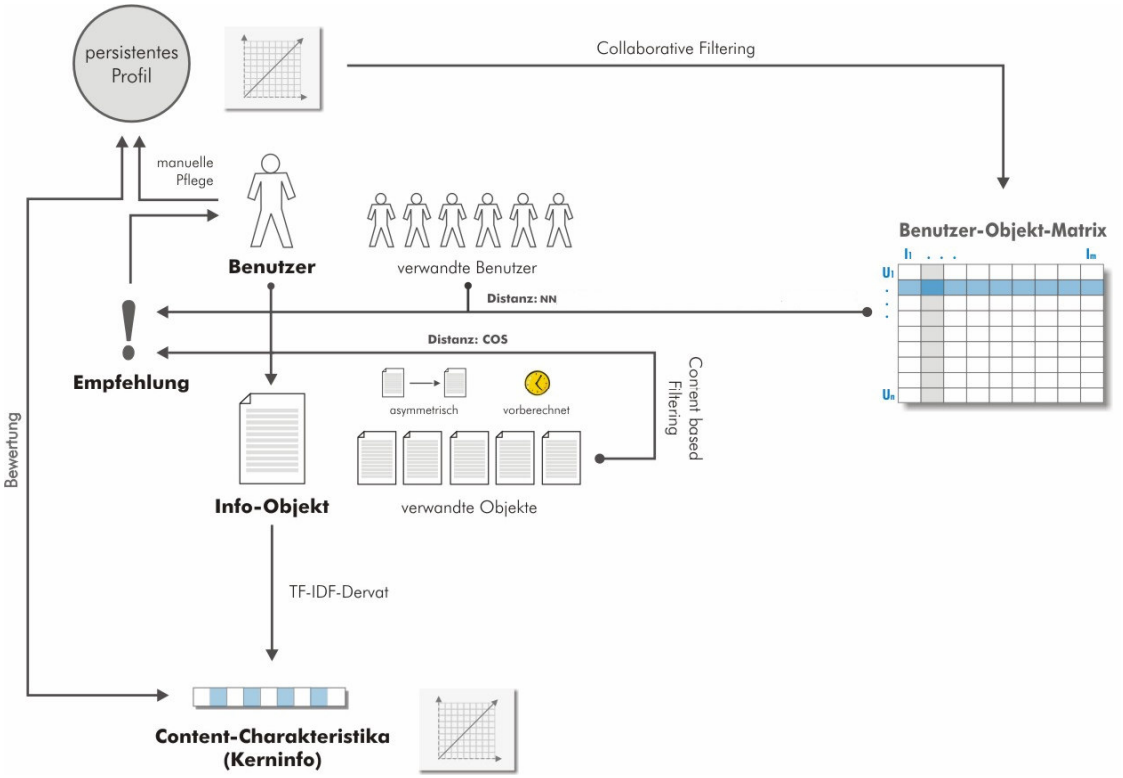


Abbildung 57: Fab

Das hybride Fab Verfahren nutzt Profile (*Selection Agent* genannt) in Form von Wortvektoren, die auf Basis der Bewertung von Texten auf einer Skala von 1-7 mittels TF-IDF-Derivat generiert werden. Eine sehr gut bewertete Seite wird direkt ähnlichen Benutzern empfohlen (CF).

Ferner kommen "Collection Agents" zum Einsatz, die auf Basis der Profile vieler Benutzer generiert werden und dann neue Texte sammeln und den Benutzern vorschlagen, deren Profile den Texten über einen bestimmten Schwellwert ähneln (CBF; Kosinus Ähnlichkeitsmaß). Es werden Benutzern auch Texte vorgeschlagen, die zum Profil ähnlicher Benutzern (ermittelt durch Nearest Neighbours Klassifikation) passen. Collection Agents, deren Empfehlungen nicht ausreichend Resonanz finden, werden entfernt.

5.4.3.2 PHOAKS [TER1997]

Beim PHOAKS ("People Help One Another Know Stuff") Verfahren werden manuelle Links (URLs) in Einträgen aus Newsgroups als positive Bewertung der verlinkten Webseite interpretiert. PHOAKS verfolgt auf den ersten Blick einen CF Ansatz, da Benutzer die News schreiben ("provider"), die Empfehlungen für lesende Benutzer ("recipients") erzeugen. Je mehr schreibende Benutzer den gleichen Link verwenden, desto höher wird die verlinkte Webseite gewertet. Folglich gilt sie als relevanter für Benutzer, die die betreffende Newsgroup besuchen. Dass die Links manuell gesetzt wurden, spricht sicher für die Einordnung als CF Verfahren.

Bei PHOAKS erhält der Benutzer also nach Newsgroups gruppierte Empfehlungen relevanter Webseiten. Welche Newsgroup den Benutzer interessiert, wählt er manuell, was quasi ein flüchtiges Profil darstellt.

Die Empfehlungen werden allerdings nach einem klassischen CBF Verfahren (Kategorie bestimmt Empfehlungen) ermittelt. Wir ordnen PHOAKS daher als hybrides Verfahren ein.

Ein Link wird nur dann "gewertet" (Heuristik zur Ermittlung der Content-Charakteristika), wenn er folgende vier Tests besteht:

- die Nachricht, die den Link enthält, wurde nicht in verschiedenen Newsgroups geposted (Annahme: sonst thematisch zu unpräzise)
- der Link ist nicht Bestandteil der Benutzersignatur

- der Link befindet sich nicht in einem zitierten Textbereich
- die Worte im Umfeld des Links
    - deuten auf Empfehlung hin
    - deuten nicht auf Werbung oder Ankündigung hin

Da der Link die Bedeutung der Seite in Bezug auf die Kategorie (= flüchtiges Profil) bestimmt, liegt ein Distanzmaß mit heuristischem Verfahren vor.

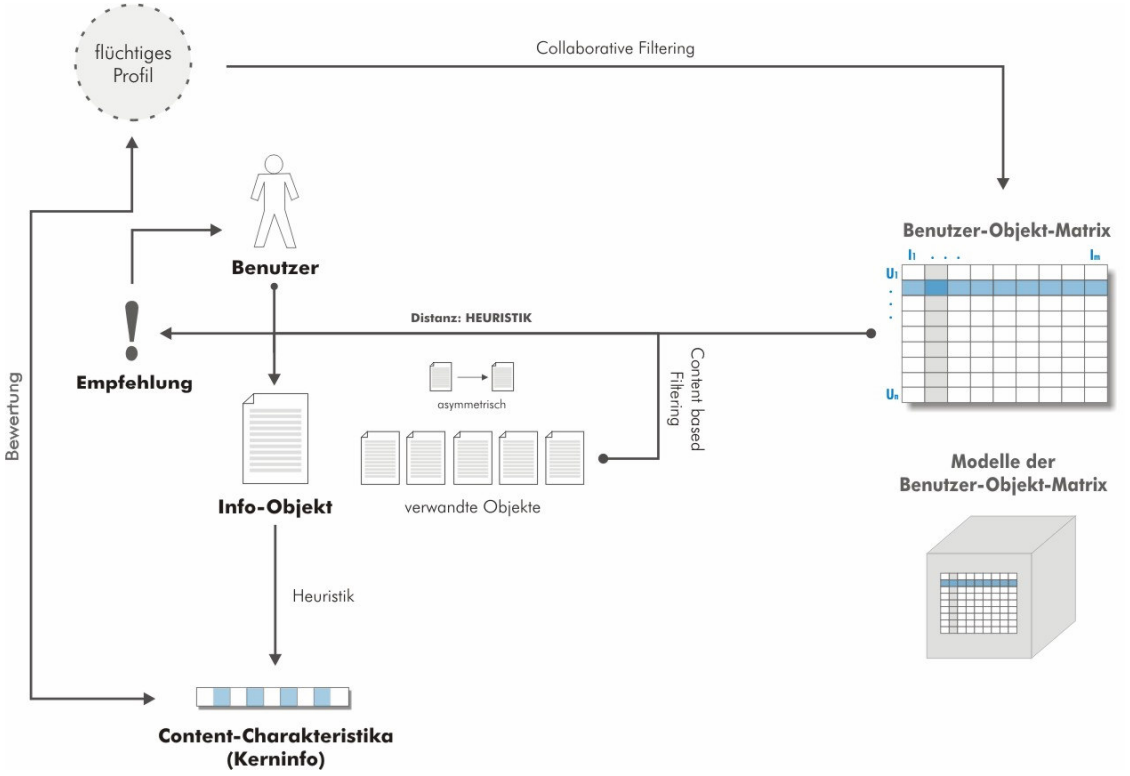


Abbildung 58: PHOAKS

5.4.3.3 Let´s Browse [LIE1998]

Das Let´s Browse Verfahren ist eine Weiterentwicklung des CBF Verfahrens Letizia (siehe 5.4.2.4) um einen CF Ansatz. Anders als bei den meisten CF Verfahren wird hier eine "physische" Benutzergruppe, die das gleiche Endgerät benutzt, betrachtet (Kiosk-System). Die Benutzer werden dabei ohne Login – anhand eines persönlichen elektronischen Ausweises, der als "meme tag" bezeichnet wird – erkannt.

Die Benutzerprofile werden zunächst initialisiert, um das Kaltstart-Problem (siehe Seite 18) zu vermeiden. Die Initialisierung erfolgt durch die Analyse einer "Start-Webseite" (einfaches explizite Strukturprofil) und durch die Verfolgung der Links zu damit verbundenen Webseiten. Für Letzteres wird ein Zeitlimit verwendet, so dass pro Benutzer zwischen 10 bis 50 Webseiten analysiert werden. Die Startseite ist dabei die persönliche Webseite des Benutzers oder seines Unternehmens. Die Analyse erfolgt mit einem TF-IDF-Derivat, um die bestimmenden Worte der Webseiten zu ermitteln. Das Benutzerprofil wird als gewichteter Wortvektor verwaltet. Im Gegensatz zum reinen CBF Verfahren Letizia werden beim Let´s Browse Verfahren bis zu 50 Worte, statt 10 Worten pro Profil, verwaltet. Tests haben diese Zahl als erforderlich erscheinen lassen, um verwandte Benutzer auch anhand weniger wichtiger Worte identifizieren zu können.

Die Empfehlungen werden auf Basis der Profile der derzeit im Einzugsbereich des Endgerätes befindlichen Benutzer ermittelt. Dazu wird ein "gemeinsames Profil" (daher als benutzerbezogenes, speicherbasiertes CF eingestuft) in Form einer einfachen Linearkombination der Wortvektoren ermittelt:

$(g_1, g_2, g_3, \ldots, g_{50}) = (b_{I1}+b_{II1}+\ldots+b_{N1}, \ \ldots, \ b_{In}+b_{IIn}+\ldots+b_{Nn})$, mit $I, II, \ldots, N$ als Benutzer.

Dann werden alle mit der aktuell angezeigten Seite verlinkten (und bei genügend Analysezeit auch weitere Link-Ebenen) Webseiten in Wortvektoren transformiert und mit dem Wortvektor des gemeinsamen Profils verglichen.

Der Vergleich erfolgt durch das Kosinus Ähnlichkeitsmaß der Vektoren. Es werden also die verlinkten Seiten angezeigt, die dem gemeinsamen Profil der Benutzer am ähnlichsten sind.

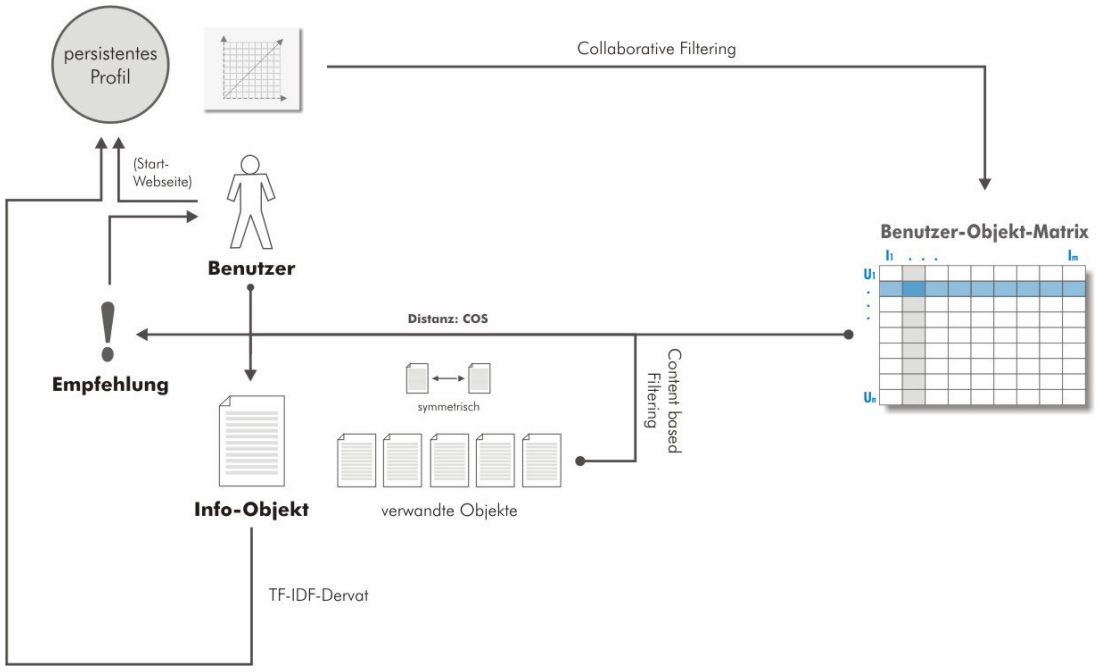


Abbildung 59: Let´s Browse

5.4.3.4 CASMIR [BER1999]

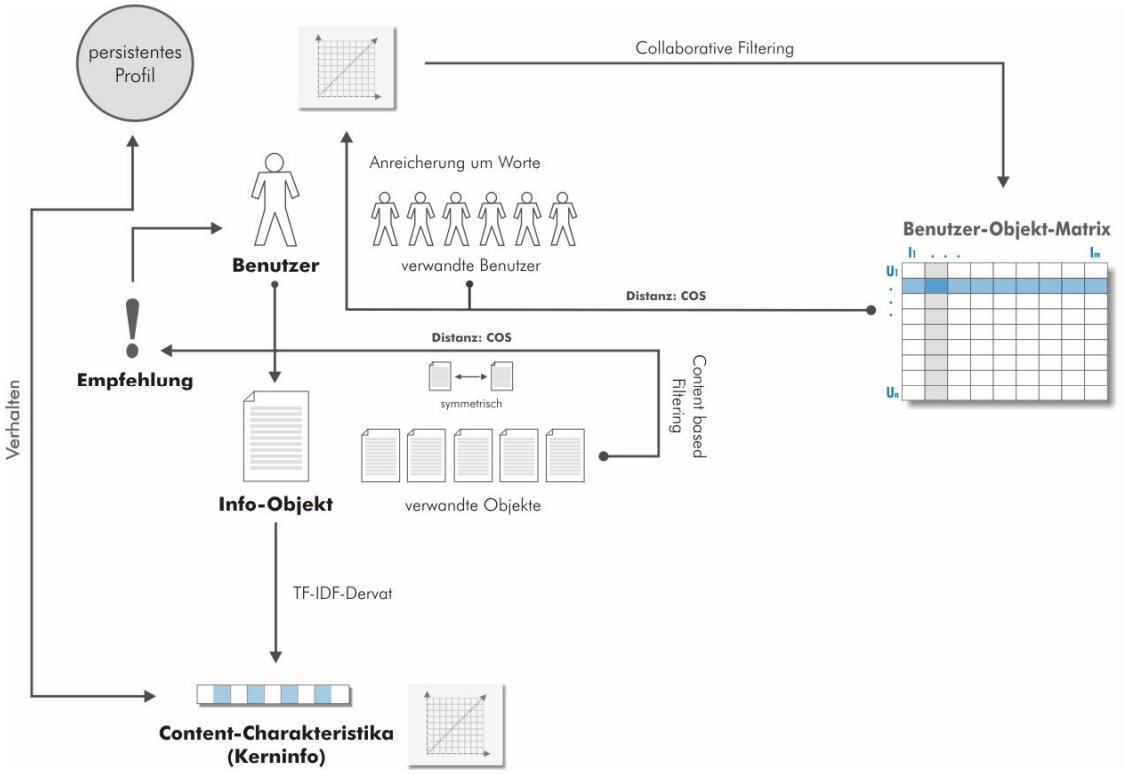


Abbildung 60: CASMIR

Das CASMIR Verfahren verwendet verschiedene Agenten, um Empfehlungen zu ermitteln. Die *Document Agents* ermitteln aufgrund einer Anfrage eines *Search Agent* die relevanten Texte ihres Textpools. Sie arbeiten

also ähnlich einer Suchmaschine. Dabei werden Anfrage und Text mit einem TF-IDF-Derivat zu Wortvektoren transformiert. Die relevanten Texte werden dann mit dem Kosinus Ähnlichkeitsmaß bestimmt.

Der *Search Agent* nimmt die Anfragen des Benutzers entgegen, selektiert die relevanten Dokumente durch Weitergabe der Anfrage an die *Document Agents* und sortiert die gefundenen Dokumenten dann nach ihrer Relevanz.

Außerdem bestimmt er die wichtigsten Worte einer Anfrage und gibt diese an den *User Assistent* weiter. Welche Worte das sind, bestimmt der Search Agent auf Basis des Verhaltens des Anwenders. Worte, die in mehr als einem Text, den der Benutzer im Suchergebnis wählt (implizites positives Feedback), vorhanden sind, werden gewählt und im Benutzerprofil aufgewertet. Worte, die in nicht selektierten Texten vorkommen, werden abhängig von der Anzahl der Texte abgewertet (implizites negatives Feedback).

Der *User Assistent* verwaltet das Benutzerprofil in Form von gewichteten Wortvektoren. Er bestimmt welchem Wortvektor ("Interesse") die vom *Search Agent* ausgewählten Worte hinzugefügt werden. Wenn der Benutzer nicht aktiv ist, führt der User Assistent automatisch Abfragen mit den am stärksten gewichteten Worten im Benutzerprofil durch.

Der *User Agent* arbeitet mit anderen *User Agents* zusammen. Wenn er nach Ausführung einer Anfrage keinen passenden Wortvektor ("Interesse") findet, gibt er die Anfrage an andere *User Agents* weiter und übernimmt einen, der von diesen angebotenen Wortvektoren ("Interesse"), als "Starthilfe". Wenn der Benutzer das System nicht benutzt, sucht der *User Agent* verwandte Wortvektoren in anderen Benutzerprofilen und reichert seine Wortvektoren mit deren Worten an (CF).

5.4.3.5 LaboUr [POH1999],[POH1997],[POH1996]

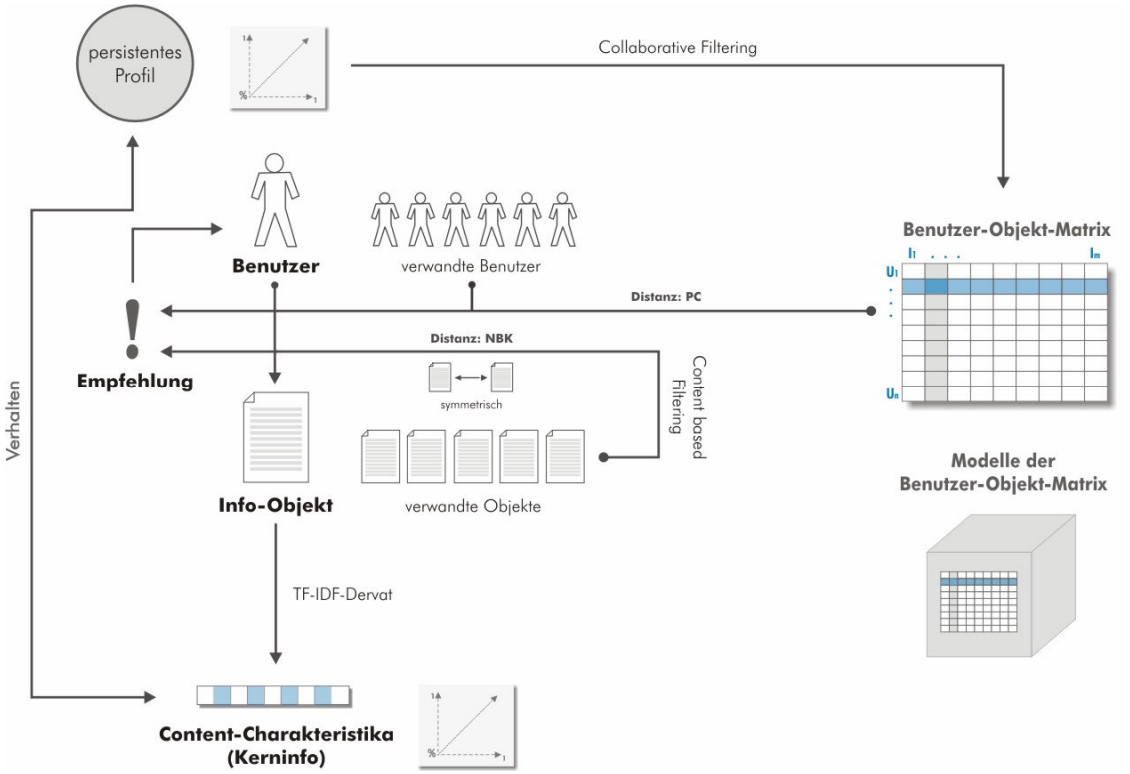


Abbildung 61: LaboUr

Das LaboUr (Learning about the user) Verfahren generiert aus dem impliziten Benutzerverhalten ein Benutzerprofil. Wenn ein Benutzer aufeinander folgend Webseiten wählt, die das gleiche Thema in Form eines Wortes behandeln, so wird dieses als bedeutsam eingestuft. Formal erfolgt dies durch einen Naiven Bayes Klassifikator (siehe Seite 22), der den Text einem Profil zuordnet. Die Texte werden mit einem TF-IDF-Derivat (Wort vorhanden: ja/nein) analysiert und in Wortvektoren gewandelt. Immer wieder aufgerufene Webseiten werden zusätzlich in absteigender Reihenfolge im Profil verwaltet und dem Benutzer in einem Baum (erste Ebene: häufig aufgerufen, zweite Ebene: weniger häufig aufgerufen) zur Verfügung gestellt. Auf Basis des Benutzerprofils werden verwandte Benutzer mit dem Korrelationskoeffizient (siehe Seite 25) gesucht.

5.4.3.6 Tango [CLA1999]

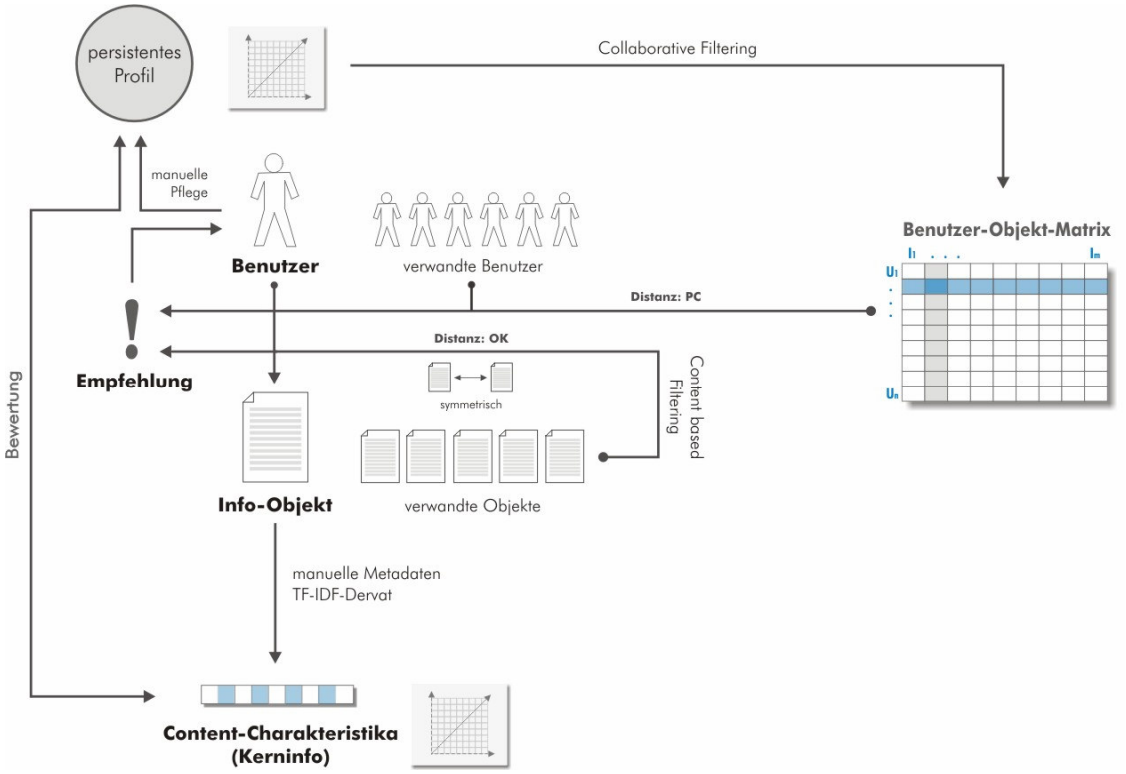


Abbildung 62: Tango

Das Tango Verfahren kombiniert CF und CBF, um die Probleme "Kaltstart" und "Spärlichkeit" (beide siehe Seite 18) sowie "Graue Schafe" (*grey sheep*) zu beseitigen. Letzteres liegt vor, wenn ein Benutzer bezüglich seiner Bewertungen zwar Profil-Überdeckungen mit anderen Benutzern hat, diese aber in Sachen Bewertung so stark variieren, dass keine "verwandten" Benutzer selektiert werden können.

Tango wird von seinen Autoren als nicht hybrides Empfehlungssystem bezeichnet, da es die Verfahren des CF und CBF autonom nutzt und deren Empfehlungen – in Abhängigkeit von deren "Stärke" – dann in einem gewichteten Mittel zusammenführt. Da unsere Definition von "hybrid" die Verbindung von CF und CBF ist, fällt es im Rahmen dieser Arbeit dennoch in die Kategorie "Hybrid-Systeme".

Der Benutzer meldet sich bei Tango an, um dann in einem Browser Zeitungsartikel aufzurufen, zu lesen und zu bewerten. Außerdem kann der Benutzer sein Profil bearbeiten.

Das Profil besteht aus einer Reihe fester Kategorien (*News*, *Business* et cetera), die der Benutzer explizit aus- und abwählen kann, sowie impliziten und expliziten Schlüsselworten. Beide sind einer Kategorie zugeordnet und Letztere als Freitext durch den Benutzer veränderbar. Erstere werden aus den am besten bewerteten Artikeln (die besten 25%) gewonnen, wobei die Kategorie-Zuordnung implizit durch die des Artikels erfolgt. Es sind manuelle Metadaten in Form der Kategorie am Artikel erforderlich. Die Auswahl der Kategorien kann der Benutzer jederzeit treffen (explizites Strukturprofil).

Die Bewertung eines Artikels (Webseite) erfolgt auf einer Skala von 1 bis 10 mit einem "Schieberegler" und gibt an mit welchem Grad (10 = am höchsten) der Benutzer ähnliche Artikel wie diesen lesen möchte.

Die Anzeige der empfohlenen Artikel erfolgt dann durch "Auszeichnung" der Artikel auf den durch normales Navigieren erreichten Übersichtsseiten. Die Auszeichnung kann dabei wahlweise durch Hervorhebung der empfehlenswerten Artikel mit "blauem Hintergrund" oder durch "Sterne" von 1 bis 5 (in Abhängigkeit von der Stärke der Empfehlung) erfolgen. Zusätzlich zeigt Tango die 10 empfehlenswertesten vom Benutzer noch nicht gelesenen Artikel an.

Der CF Teil von Tango arbeitet mit einem benutzerbezogenen Ansatz und bestimmt verwandte Benutzer mittels Korrelationskoeffizient (siehe Seite 25).

Der CBF Teil von Tango extrahiert die "relevanten" Worte eines Artikels mit einem TF-IDF-Derivat (inklusive Stoppwort-Entfernung und *Stemming*; beide siehe Seite 15). Deren Übereinstimmung zum Benutzerprofil (gleiche Kategorie wie der Artikel) wird dann mittels Overlap Koeffizient (siehe Seite 26) bestimmt. Und zwar

zum impliziten und expliziten Wort-Profil. Anschließend werden die Übereinstimmungen dieser beiden Profilarten und die Kategorie zu je einem Drittel in eine durchschnittliche Gesamtbewertung eingebracht.

Aus dem Ergebnis des CF und CBF wird dann eine Empfehlung in Form des gewichteten Mittels berechnet. Die Gewichtung wird dabei mit der vermeintlichen Höhe der Genauigkeit des Ergebnisses korreliert. So wird das CF abgewertet, wenn die Anzahl und/oder der Grad der Benutzerübereinstimmung gering ist. Das CBF hingegen verliert an Bedeutung, wenn der Benutzer keine expliziten Schlüsselworte angegeben hat oder zu wenige Artikel gut bewertet hat und daher keine impliziten Schlüsselworte genutzt werden können.

5.4.3.7 Nakif [FUN2001]

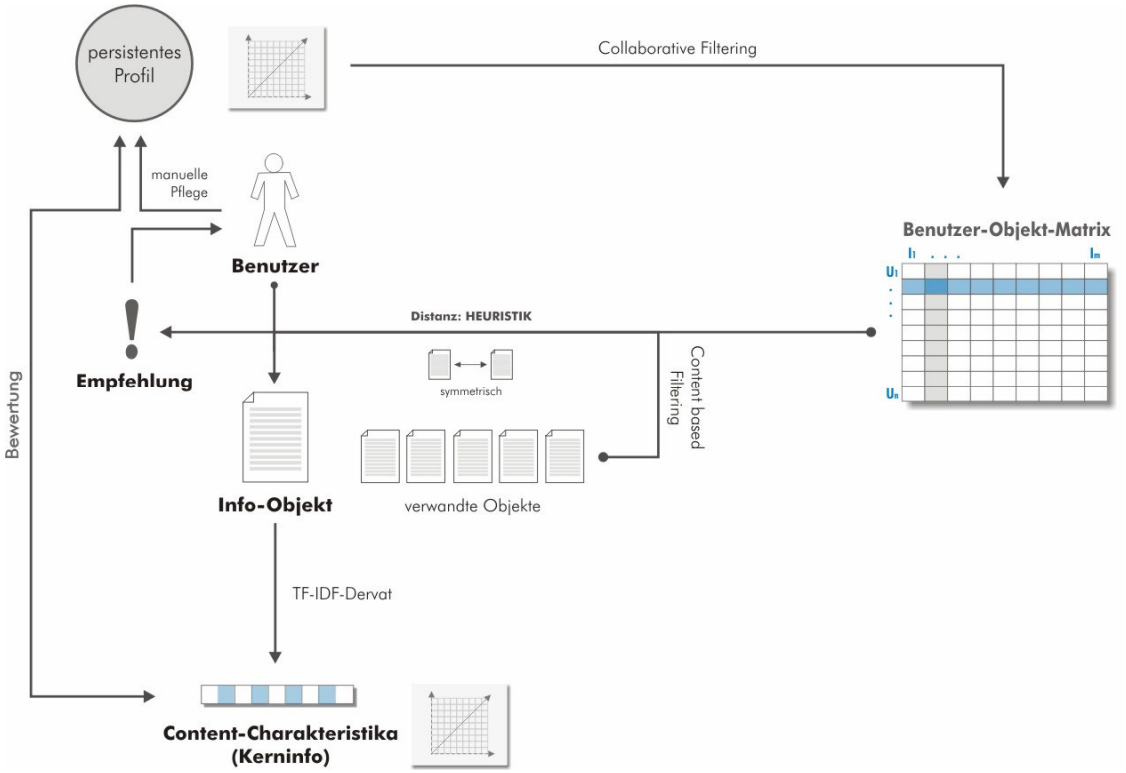


Abbildung 63: Nakif

Beim hybriden Nakif Verfahren besitzen neben den Benutzern auch die Informationsobjekte (konkret: Filme) eigene "Profile". Bei näherer Betrachtung erweisen sich diese "Profile" aber als Kombination von "Wertungsspalten" (wobei $eval_{hj}$ die Wertung des Objektes `h` durch Benutzer `j` ist) der Benutzer-Objekt-Matrix (siehe 5.2.2) und einem einfachem Wortvektor (wobei $ovec_{hk}$ für Wort `k` in Objekt `h` steht), der mit einem TF-IDF-Derivat (aus dem Text der Filmbeschreibung) gebildet wird. Die fünf ganzzahligen Wertungsoptionen liegen im Intervall [-2,2].

Das Benutzerprofil wird durch eine Matrix repräsentiert. Diese besteht aus `n` Wortvektoren - für jeden Benutzer einen. Sei v_{ij} ein solcher Wortvektor für Benutzer `j` im Profil des Benutzers `i`. Dann gibt v_{ij} (w) an, wie genau Benutzer `i` und `j` bei der Bewertung von Texten mit dem Wort w übereinstimmen.

Wie stark nun ein Text des Informationsobjektes `h` mit einem Benutzerprofil des Benutzers `i` übereinstimmt, wird dann wie folgt berechnet:

$$Match(i,h) = \Sigma_{j=1\ldots u} \Sigma_{k=1\ldots s}\ v_{ij}(w_k) * eval_{hj} * ovec_{hk}$$

mit `u` = Anzahl der Benutzer und `s` = Anzahl der Worte insgesamt. Eine Ähnlichkeit zum Korrelationskoeffizienten (siehe Seite 25) ist zwar vorhanden, allerdings weicht der Ansatz aufgrund der "Wortwertungen" statt der "Objektwertungen" doch erheblich ab und wird daher als "Heuristik" eingeordnet.

5.4.3.8 MovieLens [GOO1999]

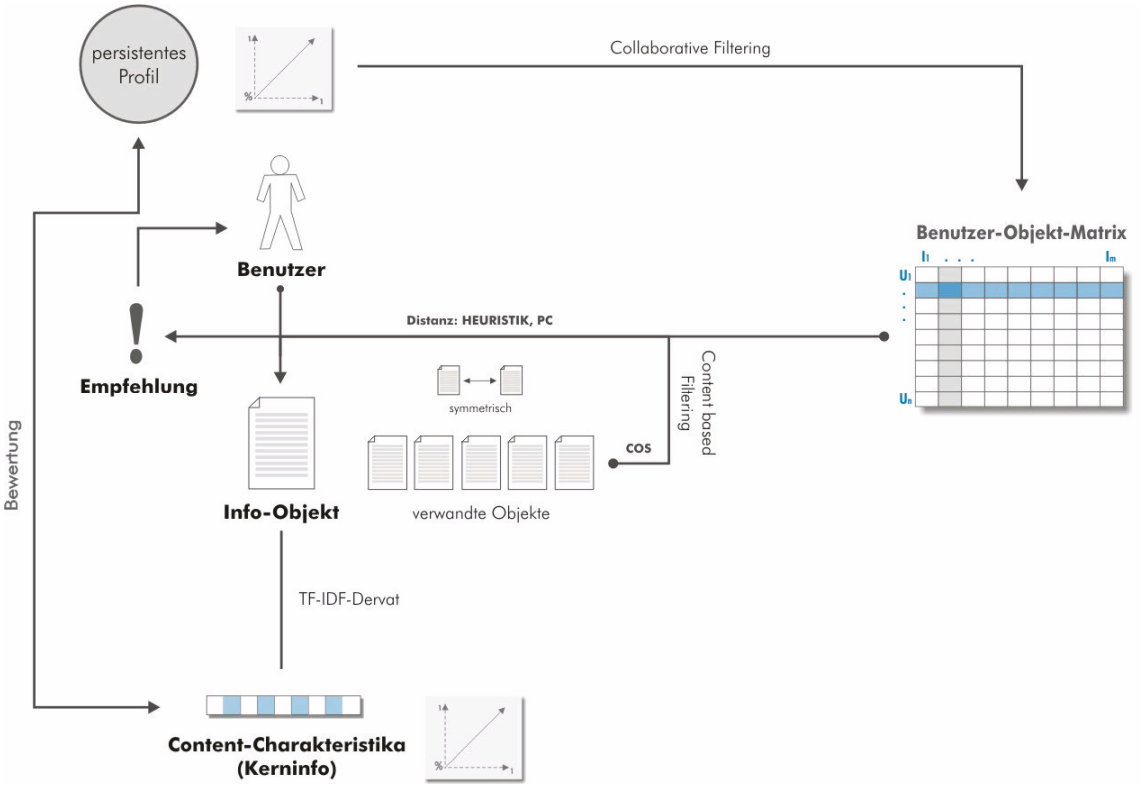


Abbildung 64: MovieLens

Das MovieLens Verfahren empfiehlt Benutzern Filme und arbeitet als Hybridsystem mit mehreren Verfahren. Zum einen kommen mehrere so genannter IF Agents (CBF) zum Einsatz:

- DoppelgaengerBots
- RipperBot
- GenreBots

Zum anderen werden diese Bots in das benutzerbezogene CF wie Benutzer einbezogen. Letzteres basiert auf der DBLens Recommendation Engine, die verwandte Benutzer mit dem Pearson Korrelationskoeffizienten ermittelt.

Die DoppelgaengerBots basieren auf einem TF-IDF-Derivat und analysieren die Filmbeschreibung. Zunächst wird die IDF als `log(N/O)` gebildet, wobei `N` die Gesamtzahl der Filme und `O` die Zahl der Filme, in denen ein Wort vorkommt, darstellen. Der TF-Anteil wird als binärer Vektor für das Vorkommen oder Nicht-Vorkommen eines Wortes gebildet. Auf Basis der Filmbewertungen eines Benutzers wird dann mit den TF-IDF-Werten ein Profil in Form eines Wortvektors gebildet. Auf Basis des Skalarproduktes des Profilvektors und der TF-Vektoren der Filme werden dann Empfehlungen ausgesprochen. Es handelt sich also um ein Derivat des Kosinus Ähnlichkeitsmaßes.

Der RipperBots setzt den Regelgenerator Ripper [COH1995] ein. Dieser basiert auf einer iterativen Regelerzeugung, die ineffiziente Regeln auf Basis der Fehlerrate verwirft. Es wurden verschiedene RipperBots durch unterschiedliche Trainingssets erzeugt.

Die GenreBots bewerten Filme in Abhängigkeit Ihres Genres, so dass *Toy Story* vom Kinderfilm-GenreBot beispielsweise "5" Punkte erhält, wohingegen dieser GenreBot dem Film *The Shining* keinen Punkt gibt. Die GenreBots arbeiten auf manuellen Metadaten.

Dadurch dass die Bots wie normale Benutzer in das benutzerbezogene CF Verfahren einbezogen werden, werden klassische Probleme reiner CF Verfahren wie "Spärlichkeit" verhindert.

5.4.4 Tabellarische Übersicht der Klassifikation der Empfehlungssysteme

In der nachfolgenden tabellarischen Übersicht werden nochmals alle untersuchten Verfahren mit Ihren Merkmalen zur Klassifikation aufgeführt. Dabei steht ein "●" für ein vorhandenes und "X" für ein nicht vorhandenes Merkmal.

Verfahren	Quelle	Jahr	Citeseer Gruppe	Citeseer	Informationsobjekt	Konzept		Content-Charakteristika Ermittlung						Profilbildung				
						CF	CBF	Manuelle Metadaten	Text-Struktur	NLP	Basis-korpus	TF-IDF Derivat	Heuristik	Manuelle Pflege	Bewertung von Objekten	Benutzer-verhalten	Flüchtiges Profil	Persistentes Profil
CRIC	-		-	-	Texte	x	●	x	●	x	x	●	●	x	x	X	X	x
AgentDLS	CAR1998	1998	10	4	Webseiten	x	●	x	x	x	x	x	●	x	x	x	X	x
Amalthaea	MOU1997	1997	50	40	Webseiten	x	●	x	x	x	x	●	x	●	●	●	x	●
Amazon.com	LIN2003	2003	10	6	Bücher	●	x	x	x	x	x	x	x	x	x	●	●	x
CALVIN	BAU2001	2001	10	1	Webseiten	x	●	x	x	x	x	●	x	x	x	●	●	x
CASMIR	BER1999	1999	10	7	Texte	●	●	x	x	x	x	●	x	x	x	●	x	●
Fab	BAL1997	1997	30	27	Webseiten	●	●	x	x	x	x	●	x	x	●	x	x	●
GroupLens	RES1994	1994	240	231	Newsgroup-Text	●	x	x	x	x	x	x	x	x	●	x	x	●
InfoFinder	KRU1997	1997	20	11	Webseiten	x	●	x	●	x	x	x	●	x	●	x	x	●
INFOS	MOC1996	1996	10	2	Newsgroup-Text	●	●	●	●	x	●	●	●	x	●	x	x	●
Infoscope	FIS1991	1991	20	15	Newsgroup-Text	x	●	x	x	x	x	x	x	●	x	●	x	●
Jester (Eigentaste)	GOL2000	2000	30	26	Texte	●	x	x	x	x	x	x	x	x	●	x	x	●
Jimminy	RHO2000a	2000	20	16	Texte	x	●	x	x	x	x	●	x	x	x	x	x	x
LaboUr	POH1999	1999	10	7	Webseiten	●	●	x	x	x	x	●	x	x	x	●	x	●
Let´s browse	LIE1998	1998	30	21	Webseiten	●	●	x	x	x	x	●	x	●	x	x	x	●
Letizia	LIE1995	1995	230	221	Webseiten	x	●	x	x	x	x	●	x	●	x	●	x	●
LexicalChainer	GRE1998	1998	10	8	Texte	x	●	x	x	●	x	x	x	x	x	x	x	x
LIBRA	MOO2000	2000	20	19	Bücher	x	●	●	x	x	x	x	x	x	●	x	x	●
Margin Notes	RHO2000	2000	20	16	Webseiten	x	●	x	●	x	x	●	x	x	x	x	x	x
MetaMarker	PAI2001	2001	10	3	E-Mails	x	●	x	x	●	x	x	●	x	x	●	x	●
Movielens	GOO1999	1999	60	50	Filme	●	●	●	x	x	x	●	x	x	●	x	x	●
Nakif	FUN2001	2001	10	0	Filme	●	●	x	x	x	x	●	x	x	●	x	x	●

Verfahren	Quelle	Jahr	Citeseer Gruppe	Citeseer	Informationsobjekt	Konzept		Content-Charakteristika Ermittlung						Profilbildung				
						CF	CBF	Manuelle Metadaten	Text-Struktur	NLP	Basiskorpus	TF-IDF Derivat	Heuristik	Manuelle Pflege	Bewertung von Objekten	Benutzerverhalten	Flüchtiges Profil	Persistentes Profil
News Dude	BIL1999	1999	10	4	Texte	x	●	x	x	x	x	●	x	x	●	x	x	●
Newsweeder	LAN1995	1995	30	20	Newsgroup-Text	x	●	x	x	x	x	●	x	x	●	x	x	●
PACT	MOB2000	2000	20	12	Newsgroup-Text	●	x	x	x	x	x	x	x	x	x	●	●	x
PHOAKS	TER1997	1997	70	62	Webseiten	●	●	x	x	x	x	x	●	x	●	x	●	x
PocketLens	MIL2004	2004	10	3	Filme	●	x	x	x	x	x	x	x	●	●	x	x	●
PowerScout	LIE2001	2001	20	18	Webseiten	x	●	x	●	x	x	●	x	x	x	●	x	●
PRES	MET2000	2000	10	0	Webseiten	x	●	x	x	x	x	●	x	x	x	●	x	●
Remembrance Agent	RHO199x	1996	30	27	Texte	x	●	x	x	x	x	●	x	x	x	x	x	x
Ringo	SHA1995	1995	90	87	Musik	●	x	x	x	x	x	x	x	x	●	x	x	●
SIFT	YAN1999	1999	40	32	Newsgroup-Text	x	●	x	x	x	x	●	x	●	x	x	x	●
Siteseer	RUC1997	1997	40	37	Webseiten	●	x	●	x	x	x	x	x	●	●	x	x	●
SLIDER	BAL1998	1998	20	10	Webseiten	x	●	x	x	x	x	●	x	●	●	●	x	●
SUITOR	MAG2000	2000	20	10	Webseiten	x	●	x	x	x	x	●	x	x	x	●	x	●
SurfLen	FU2000	2000	20	16	Webseiten	●	x	x	x	x	x	x	x	x	x	●	●	x
Syskill & Webert	PAZ1996	1996	20	14	Webseiten	x	●	x	x	x	x	●	x	x	●	x	x	●
Tango	CLA1999	1999	30	28	Webseiten	●	●	●	x	x	x	●	x	●	●	x	x	●
Tapestry	GOL1992	1992	220	212	E-Mails	●	x	x	x	x	x	x	x	●	●	x	x	●
The information lense	MAL1986	1986	30	23	E-Mails	x	●	●	x	x	x	x	x	●	x		x	x
WAIR	SEO2000	2000	10	4	Webseiten	x	●	x	x	x	x	●	x	x	x	●	x	●
Watson	BUD1999	1999	20	17	Texte	x	●	x	●	x	x	●	●	x	x	x	x	x
Webmate	CHE1998	1998	60	55	Webseiten	x	●	x	x	x	x	●	x	x	●	x	x	●
Websail	CHE2000	2000	20	12	Webseiten	x	●	x	x	x	x	●	x	x	●	x	●	●
WebTop	WOL2002	2002	10	3	Webseiten	x	●	x	x	x	x	●	x	●	x	x	x	●
WebWatcher	ARM1998	1998	210	201	Webseiten	x	●	x	x	x	x	●	x	●	x	x	x	●
WordSieve	BAU2001a	2001	10	3	Texte	x	●	x	x	x	x	x	●	x	x	●	x	●

Verfahren	Quelle	Jahr	Citeseer Gruppe	Citeseer	Informationsobjekt	Berechnung des Distanz							CF-Technik			
						asymmetrisch	symmetrisch	Vektorbasiert	Standard-Verfahren	Regelbasiert	Heuristik	vorberechnet	speicherbasiert	modellbasiert	benutzerbezogen	objektbezogen
CRIC	-		-	-	Texte	●	x	x	x	x	●	●	x	x	x	x
AgentDLS	CAR1998	1998	10	4	Webseiten	x	●	●	NBK	x	x	x	x	x	x	x
Amalthaea	MOU1997	1997	50	40	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
Amazon.com	LIN2003	2003	10	6	Bücher	x	●	●	COS	x	x	●	X	●	x	●
CALVIN	BAU2001	2001	10	1	Webseiten	x	●	●	CBR	x	x	x	●	x	x	●
CASMIR	BER1999	1999	10	7	Texte	x	●	●	COS	x	x	x	●	x	●	x
Fab	BAL1997	1997	30	27	Webseiten	x	●	●	COS	x	x	x	●	x	●	x
GroupLens	RES1994	1994	240	231	Newsgroup-Text	x	●	x	PC	x	x	x	X	●	●	x
InfoFinder	KRU1997	1997	20	11	Webseiten	●	x	x	ID3	x	●	●	x	x	x	x
INFOS	MOC1996	1996	10	2	Newsgroup-Text	x	x	X	GHC+CBR	x	●	x	X	x	x	X
Infoscope	FIS1991	1991	20	15	Newsgroup-Text	x	●	●	x	●	●	x	x	x	x	x
Jester (Eigentaste)	GOL2000	2000	30	26	Texte	x	x	x	NN+KMC	x	x	x	x	●	●	x
Jimminy	RHO2000a	2000	20	16	Texte	x	●	●	COS	x	x	●	x	x	x	x
LaboUr	POH1999	1999	10	7	Webseiten	x	●	●	NBK	x	x	x	x	●	●	x
Let´s browse	LIE1998	1998	30	21	Webseiten	x	●	●	COS	x	x	x	●	x	●	x
Letizia	LIE1995	1995	230	221	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
LexicalChainer	GRE1998	1998	10	8	Texte	x	●	●	COS	x	●	x	x	x	x	x
LIBRA	MOO2000	2000	20	19	Bücher	x	●	●	NBK	x	x	x	x	x	x	x
Margin Notes	RHO2000	2000	20	16	Webseiten	x	●	●	OWS	x	x	●	x	x	x	x
MetaMarker	PAI2001	2001	10	3	E-Mails	x	●	●	NBK	x	x	x	x	x	x	x
Movielens	GOO1999	1999	60	50	Filme	x	●	●	x	●	●	●	●	x	●	x
Nakif	FUN2001	2001	10	0	Filme	x	●	●	x	x	●	x	●	x	x	x
News Dude	BIL1999	1999	10	4	Texte	x	●	●	NBK	x	x	x	x	x	x	x
Newsweeder	LAN1995	1995	30	20	Newsgroup-Text	x	●	●	MDL	x	x	x	x	x	x	x
PACT	MOB2000	2000	20	12	Newsgroup-Text	x	●	●	KMC+COS	x	x	x	x	●	●	x
PHOAKS	TER1997	1997	70	62	Webseiten	x	x	x	x	x	●	x	x	x	x	x

Verfahren	Quelle	Jahr	Cite-seer Gruppe	Cite-seer	Informations-objekt	Berechnung des Distanz							CF-Technik			
						asym-metrisch	sym-metrisch	Vektor-basiert	Standard-Verfahren	Regel-basiert	Heuristik	vorbe-rechnet	speicher-basiert	modell-basiert	benutzer-bezogen	objekt-bezogen
PocketLens	MIL2004	2004	10	3	Filme	x	●	●	COS	x	x	x	x	●	x	●
PowerScout	LIE2001	2001	20	18	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
PRES	MET2000	2000	10	0	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
Remembrance Agent	RHO199x	1996	30	27	Texte	x	●	●	COS	x	x	●	x	x	x	x
Ringo	SHA1995	1995	90	87	Musik	x	●	x	PC	x	x	x	●	x	●	x
SIFT	YAN1999	1999	40	32	Newsgroup-Text	x	●	●	COS	x	x	x	x	x	x	x
Siteseer	RUC1997	1997	40	37	Webseiten	●	x	●	NN	x	x	x	●	x	●	x
SLIDER	BAL1998	1998	20	10	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
SUITOR	MAG2000	2000	20	10	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
SurfLen	FU2000	2000	20	16	Webseiten	●	x	x	x	●	●	●	●	x	x	●
Syskill & Webert	PAZ1996	1996	20	14	Webseiten	x	●	●	NBK	x	x	x	x	x	x	x
Tango	CLA1999	1999	30	28	Webseiten	x	●	x	OK	x	x	x	●	x	●	x
Tapestry	GOL1992	1992	220	212	E-Mails	●	x	x	x	●	x	x	x	●	●	x
The information lense	MAL1986	1986	30	23	E-Mails	x	x	x	x	●	x	x	x	x	x	x
WAIR	SEO2000	2000	10	4	Webseiten	x	●	●	x	x	●	x	x	x	x	x
Watson	BUD1999	1999	20	17	Texte	●	x	x	x	x	●	x	x	x	x	x
Webmate	CHE1998	1998	60	55	Webseiten	x	●	●	COS	x	x	●	x	x	x	x
Websail	CHE2000	2000	20	12	Webseiten	x	●	●	COS	x	x	x	x	x	x	x
WebTop	WOL2002	2002	10	3	Webseiten	●	x	x	x	x	●	x	x	x	x	x
WebWatcher	ARM1998	1998	210	201	Webseiten	x	●	●	MI	x	x	x	x	x	x	x
WordSieve	BAU2001a	2001	10	3	Texte	x	●	●	COS	x	●	x	x	x	x	x

TEIL III – BESCHREIBUNG UND EVALUATION DES VERFAHRENS

Zusammenfassung – Teil III
Abgrenzung des CRIC Verfahrens zu anderen Empfehlungssystemen.
Die Prämissen des Verfahrens werden kurz wiederholt.
Das Konzept wird grob beschrieben.
Das Verfahren wird im Detail vorgestellt.
Es wird ein Verfahren zur Ableitung der Laufzeitkomplexität für SQL-Anweisungen auf Basis von DBMS entwickelt, um damit die theoretische Laufzeitkomplexität des im Rahmen der Arbeit vorgestellten Verfahrens ermitteln zu können.
Die theoretische Laufzeitkomplexität des Verfahrens wird hergeleitet und Optimierungen erläutert.
Die Ergebnisse der realen Laufzeituntersuchungen werden vorgestellt.
Die Ergebnisse der empirischen qualitativen Evaluation werden vorgestellt.

6 Der gewählte Lösungsansatz

Der im Folgenden beschriebene Ansatz stellt mit *CRIC*[6] ein neues Verfahren für ein *Empfehlungssystem* vor. Dieser Ansatz unterscheidet sich von ähnlichen Verfahren im Wesentlichen dadurch, dass eine eigene Heuristik zum Einsatz kommt, die ein TF-IDF-Derivat (siehe Seite 19) mit Eigenschaften der Textstruktur verbindet, eine dynamische Stoppwortliste erzeugt und darauf eine asymmetrische vorberechnete Distanzmatrix[7] aufbaut. Als Basis können beliebige unstrukturierte Texte verwendet werden. Insbesondere werden keine Metadaten, aber auch keine Thesauri oder Korpora benötigt.

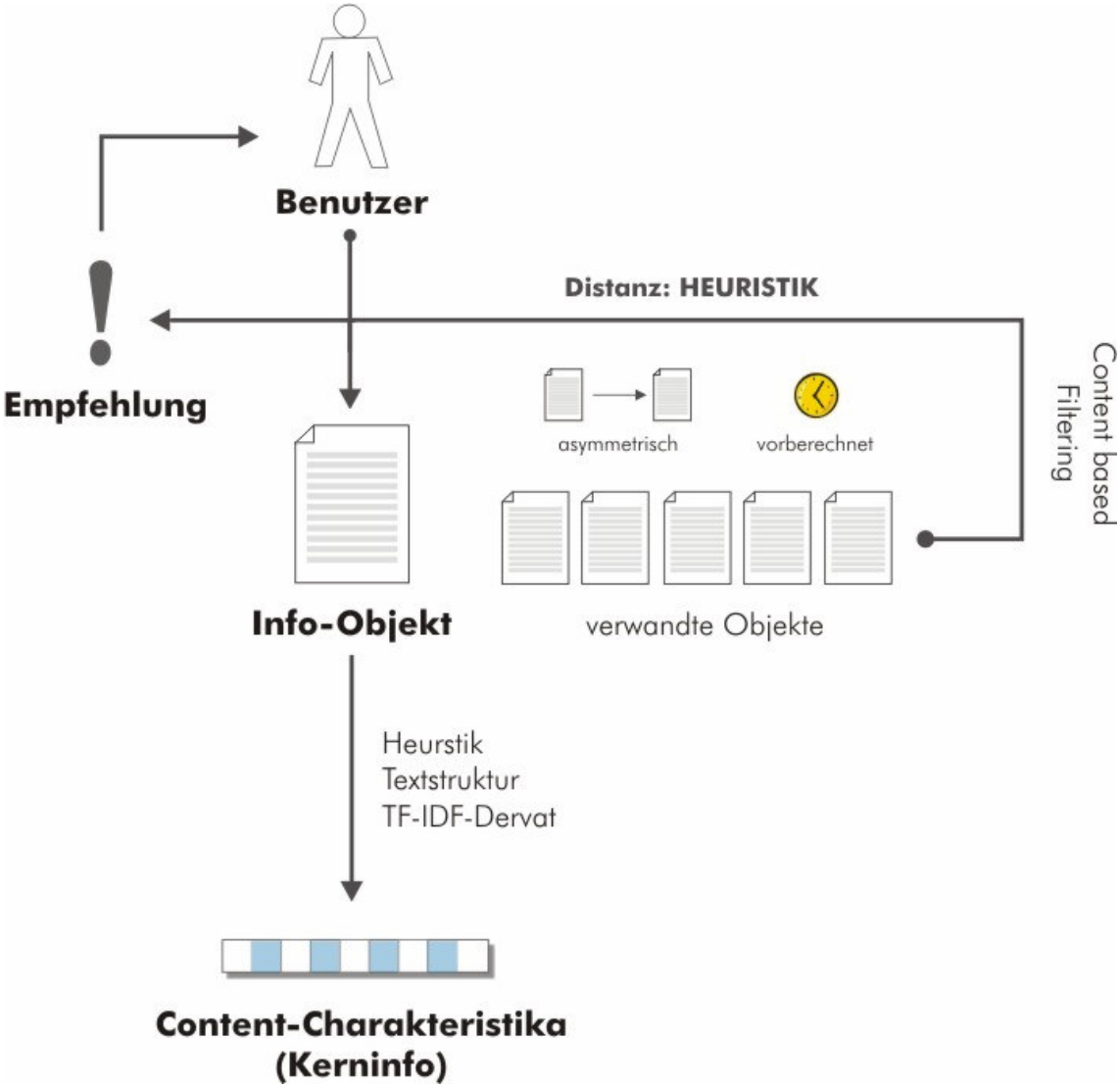


Abbildung 65: CRIC

[6] Content Recommendation in Context

[7] Für die Proximität oder inhaltlich Nähe zweier Informationsobjekte wird im Rahmen von Empfehlungssystemen oft der Begriff der Metrik verwendet. Da aber nicht alle Verfahren symmetrische Proximität liefern und daher dem formalen Begriff der Metrik nicht gerecht werden, verwenden wir den Begriff der Distanz.

Die semantischen Beziehungen zwischen Textobjekten werden komplett automatisiert und sprachunabhängig[8] hergestellt. Das Verfahren sollte folgende Prämissen erfüllen:

(i) Akzeptanz beim Benutzer
   c. Konstant auch über längere Zeitintervalle

(ii) Garantiert schnelle Antwortzeiten
   d. Auch bei großen Textmengen
   e. Auch in Hochlastumgebungen (hohe Nutzungsrate)

(iii) Erschließung aller vorhandenen Quellen
   f. Unstrukturierte Texte ohne Metadaten
   g. Unterschiedliche Dateiformate

(iv) Automatisierte Verarbeitung
   h. Kein manueller Eingriff durch den Autor
   i. Kein manueller Eingriff durch den Benutzer

CRIC ist als Funktion $f_n(I)$ interpretierbar, die für ein Informationsobjekt I (einen Text) die n "ähnlichsten" Informationsobjekte $I_{i=1\ldots n}$ ("semantisch verwandte" Texte) liefert. Die durch $f_n(I)$ berechneten n Texte mit dem geringsten semantischen Abstand zu Text I werden in einer rekursiven $N:M$ Relation, die auf der die Textbasis repräsentierenden Relation T aufsetzt, gespeichert.

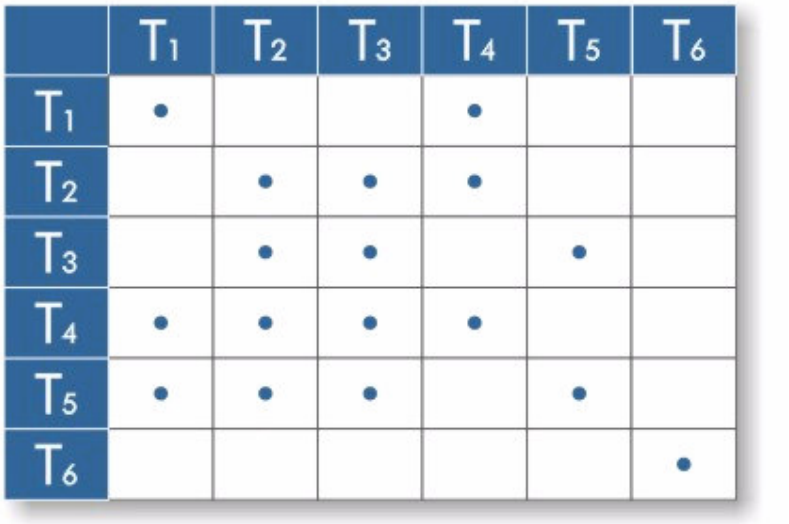


Abbildung 66: Die asymmetrische Distanzmatrix ist nicht voll besetzt

Konkret implementiert wird die Distanzmatrix durch die Hilfstabelle V. Da die Matrix aufgrund der Limitation auf "n" verwandte Texte nur spärlich besetzt ist, ist dies eine effiziente Datenstruktur.

[8] Die Sprachunabhängigkeit gilt für romanische und indogermanische Sprachen, für welche die entwickelte Heuristik unverändert angewendet werden kann.

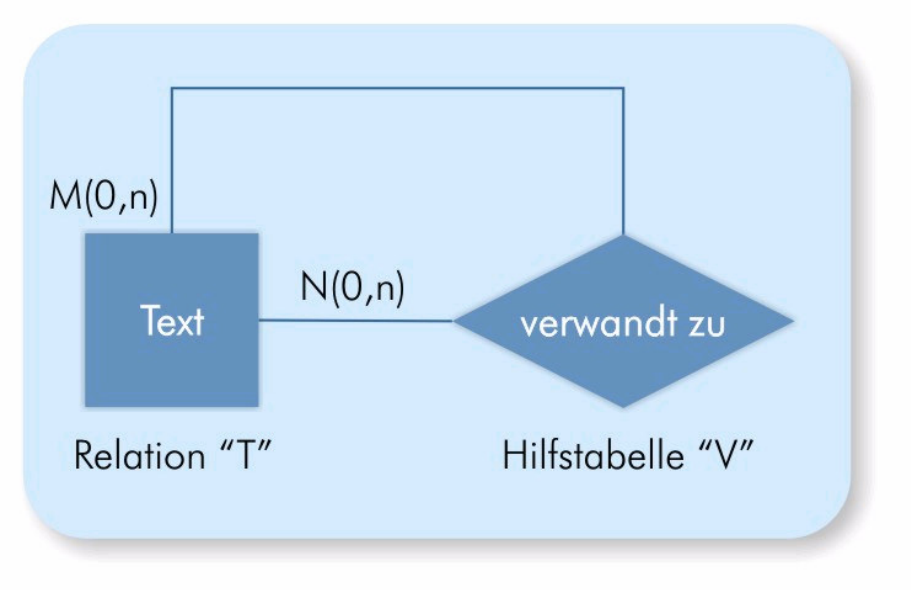


Abbildung 67: ER-Diagramm der Datenstruktur: die Hilfstabelle "V" enthält die vorberechneten Empfehlungen und repräsentiert die Distanzmatrix

Die Funktion `f` zur Berechnung der semantischen Distanz ist im Vergleich zu den meisten vektorbasierten Verfahren nicht symmetrisch. Das Verfahren sollte im Echtzeitbetrieb für große Websites und Intranets einsetzbar sein. Daher werden die Distanzen zwischen den Informationsobjekten vorberechnet.

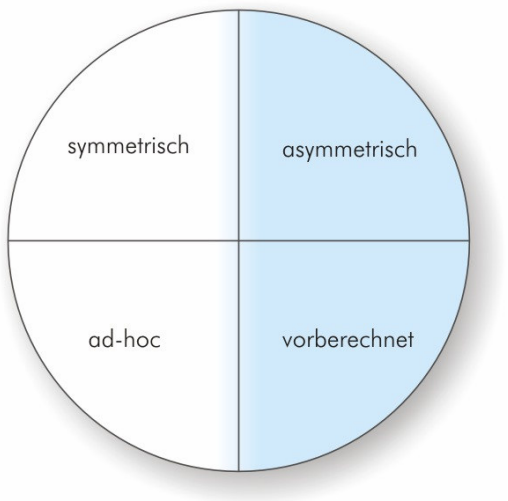


Abbildung 68: CRIC arbeitet asymmetrisch und berechnet die Distanzen "offline" vor

Eine asymmetrische Distanzfunktion wirft das Problem auf, dass mit jedem neuen Informationsobjekt die komplette Distanzmatrix neu berechnet werden muss:

Durch eine Vereinfachung finden bei CRIC neue Texte aber Eingang in `V`, ohne dass $f_n(I)$ für alle Texte in `T` (Textbasis = Menge aller Informationsobjekte) neu berechnet werden muss.

Um die Effizienz des Verfahrens in Bezug auf die Geschwindigkeit und die Qualität der Empfehlungen zu prüfen, wurden umfangreiche Praxistests durchgeführt. Zur Evaluation im Hochlastbetrieb konnte ein großes Internet-Portal mit durchschnittlich mehr als 100.000.000 Seitenabrufen pro Monat verwendet werden. Als Testbasis für die Qualität der Empfehlungen von CRIC konnte ein Portal eines Verlages verwendet werden. Dadurch war der direkte Vergleich der von CRIC und von professionellen Redakteuren gemachten Empfehlungen möglich. Gemessen wurde dabei die Nutzung der ausgesprochenen Empfehlungen.

6.1 Eine effiziente semantische Distanzfunktion

Es gibt zahlreiche Ansätze semantisch verwandte Texte für einen gegebenen Text `T` zu berechnen. Die Problemstellung lässt sich in zwei Schritte zerlegen. Zunächst werden die Texte in eine vergleichbare Form transformiert. Erst dann kann ein Distanzmaß den semantischen Abstand berechnen.

Die bekanntesten Transformationsvarianten ermitteln dazu die charakteristischen Attribute eines Informationsobjektes. Diese Attribute sind bei Texten in der Regel Fragmente (Worte, Sätze et cetera) des Objektes. Prominente Ansätze dafür sind:

- manuellen Metadaten
- Textstruktur
- NLP (natural language processing)
- Basiskorpus (Kollokation et cetera)
- TF-IDF-Derivat

oder - wie auch bei CRIC - eine eigene Heuristik. Auf Basis der untersuchten Empfehlungssysteme ergibt sich folgende Verteilung:

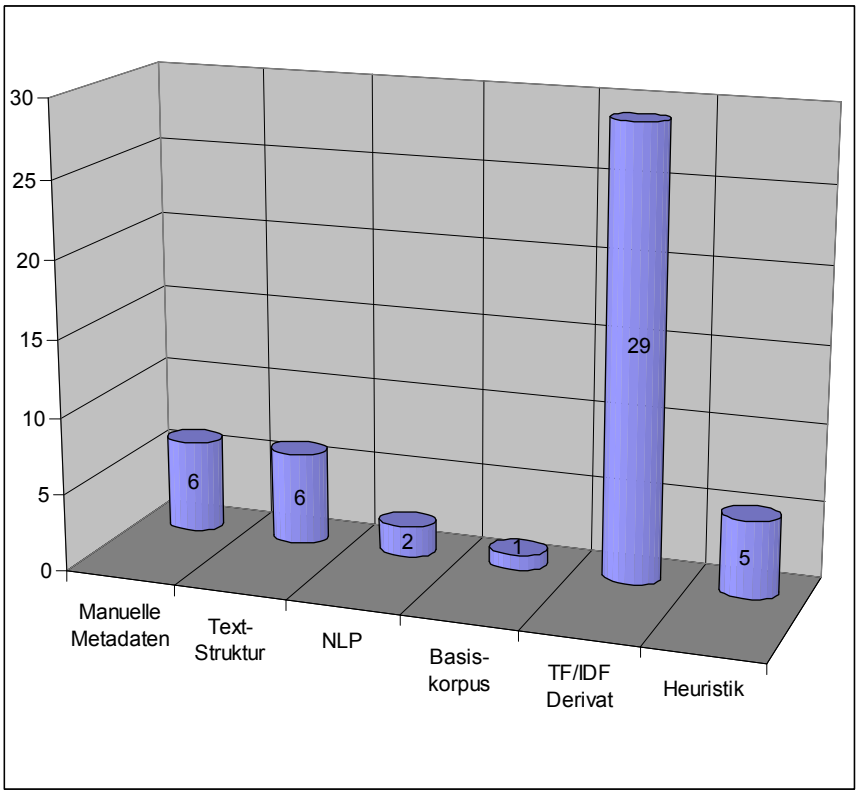


Abbildung 69:Ein Großteil der betrachteten Verfahren arbeitet mit einem TF-IDF-Derivat (die y-Achse gibt die Zahl der betroffenen Verfahren an).

Bei der Distanzermittlung eines Verfahrens spielen folgende Eigenschaften eine Rolle:

- Asymmetrisches Distanzmaß
- Symmetrisches Distanzmaß
- Distanzmaß basiert auf Vektoren
- Standardverfahren
- Regelbasiert
- Mittels Heuristik
- Vorberechnet

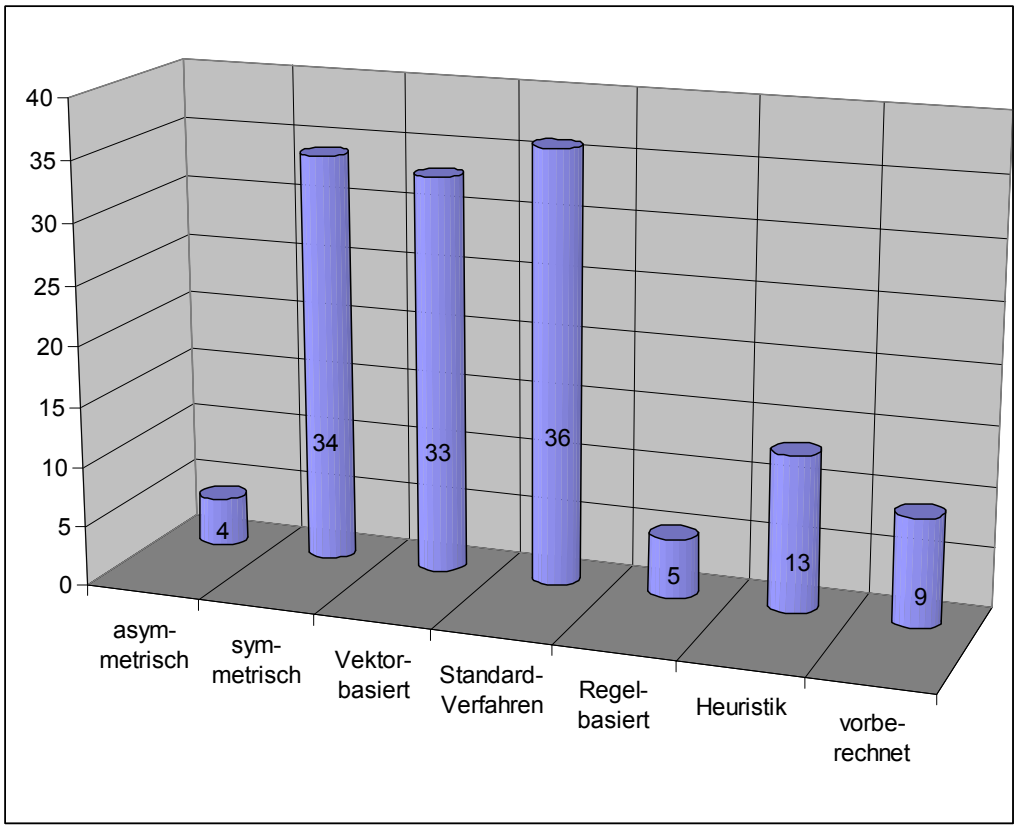


Abbildung 70:Bei den Verfahren zur Distanzermittlung kommen vorrangig symmetrische, vektorbasierte Standardverfahren zum Einsatz (die y-Achse gibt die Zahl der betroffenen Verfahren an).

Die Distanz wird dann in der Regel mit einem der folgenden Standardverfahren berechnet:

- Kosinus Ähnlichkeitsmaß (COS)
- Minimum Description Length (MDL)
- Naiver Bayes-Klassifikator (NBK)
- Mutual Information (MI)
- Case based reasoning (CBR)
- Korrelationskoeffizient (PC)
- Global Hill Climbing (GHC)
- Okapi Weighting Scheme (OWS)
- Overlap Koeffizient (OK)
- ID3 (Entscheidungsbaum)
- K-Means-Clustering (KMC)
- Nearest Neighbour (NN)

Alternativ kommt auch hier wieder eine Heuristik zum Einsatz.

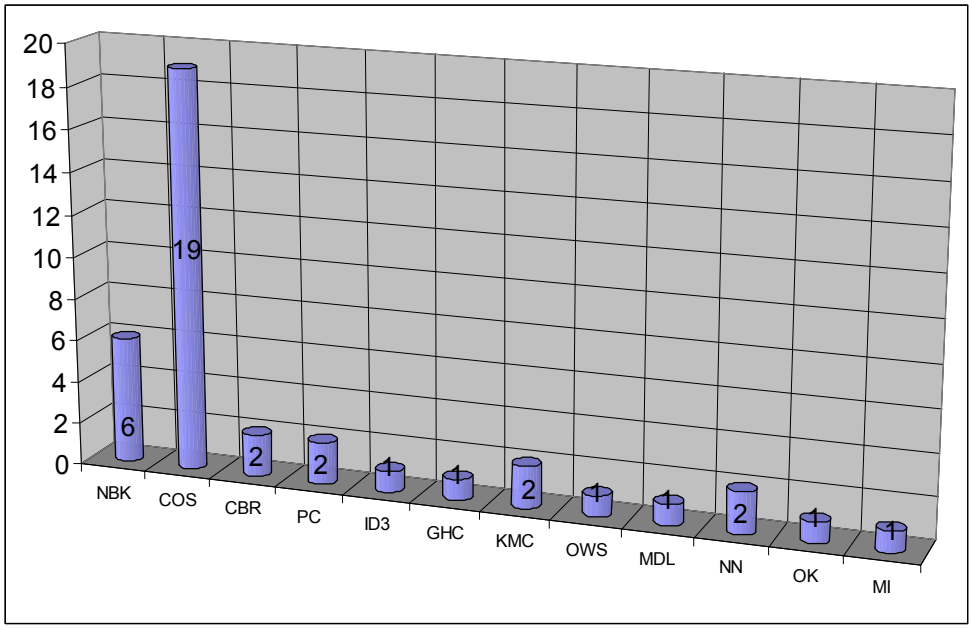


Abbildung 71:Bei den Verfahren zur Distanzermittlung wird überwiegend das Kosinus Ähnlichkeitsmaß gefolgt vom Naiven Bayes Klassifikator eingesetzt (die y-Achse gibt die Zahl der betroffenen Verfahren an).

Bei CRIC kommt eine eigene Heuristik zum Einsatz, die im Weiteren beschrieben wird.

6.1.1 Schlüsselworte ermitteln

Als Content Charakteristika (charakteristische Attribute des Textes) verwendet CRIC die Schlüsselworte eines Textes. Es existieren Standardverfahren, um die Schlüsselworte eines unstrukturierten Textes zu ermitteln. Diese basieren in der Regel darauf die vermeintliche Bedeutung eines Wortes für einen gegebenen Text zu bestimmen und die wichtigsten Worte als Schlüsselworte abzuleiten.

6.1.1.1 Manuelle Verfahren

Neben zahlreichen informellen Anleitungen zur manuellen Selektion von Schlüsselworten (manuelle Indexierung) gibt es auch standardisierte Vorschriften [UML1998],[NIS1997]. Allerdings sucht man auch in diesen vergebens nach einem algorithmisch verwertbaren Verfahren zur Bestimmung von Schlüsselworten oder Schlagworten[9].

So geben die Regeln für den Schlagwortkatalog [UML1998] unter §4 (Inhaltsanalyse) vor:

- *Feststellen des Inhalts bzw. der inhaltlichen Schwerpunkte eines vorliegenden Dokuments, also der darin behandelten Gegenstände. Maßgebend für die Wahl der Schlagwörter ist der Inhalt, nicht die jeweilige Titelfassung mit den sich daraus ergebenden Stichwörtern.*
- *Gewichtung und Auswahl der zu erschließenden inhaltlichen Aspekte unter Berücksichtigung der Aufnahmeprinzipien für den Schlagwortkatalog (vgl. § 3) und der Grundprinzipien der Schlagwortkatalogisierung (vgl. § 6).*
- *Ermittlung eines oder mehrerer Begriffe, die den wesentlichen Inhaltskomponenten eines Dokuments entsprechen.*
- *Umsetzung der ausgewählten Begriffe in prägnante Bezeichnungen zum Zweck der möglichen Ansetzung einzelner Schlagwörter (vgl. § 9).*

Damit wird zwar die Zielsetzung, nicht aber der Prozess, definiert.

[9] Schlüsselworte (oder Stichwörter) sind Terme, die dem Dokument zur Beschreibung entnommen werden. Schlagworte sind Terme, die dem Dokument zur Inhaltsbeschreibung zugeordnet werden. Schlagworte müssen nicht unbedingt im Dokument selbst enthalten sein.

6.1.1.2 TF-IDF

Das bekannteste Verfahren ist sicherlich der TF-IDF Ansatz. Es handelt sich um ein einfaches Verfahren zur Bestimmung von Wort-Gewichtungen, welches das häufige Auftreten eines Wortes im konkreten Text "belohnt" und ein häufiges Vorkommen über alle Texte "bestraft".

Abbildung 72:Die repräsentativen Schlüsselworte eines Textes zu ermitteln beeinflusst die nachfolgende Distanzermittlung und damit die Qualität der Empfehlungen maßgeblich

Formal gilt beim TF-IDF Verfahren für die Bedeutung (oder das Gewicht) `w(W)` eines Wort `W`:

`w(W)=TF*IDF` mit $TF=N_T(W)$ und $IDF=\log(|C|/I_C(W))$

also

$$w(W) = N_T(W) * \log_2 \frac{|C|}{I_C(W)}$$

mit

$N_T(W)$: Anzahl der Instanzen ("frequency"; Häufigkeit) von `W` im Text `T`

$I_C(W)$: Anzahl der Texte in der Textbasis $C = \{T_1, \ldots, T_n\}$ mit mindestens einer Instanz von `W`

$|C|$: Anzahl der Texte der Textbasis `C`

6.1.1.3 Die CRIC Heuristik

Das CRIC Verfahren zur Schlüsselwortselektion stellt ein TF-IDF-Derivat dar, das die Bedeutung des Wortes `W` eines gegebenen Textes `T`, der Bestandteil der Textbasis `C` ist, aus folgenden Parametern ableitet:

- $N_T(W)$: Anzahl der Instanzen des Wortes `W` in Text `T`
- $N_C(W)$: Anzahl der Instanzen des Wortes `W` in der Textbasis `C` mit $C=\{T_1, \ldots, T_n\}$
- $P_T(W)$: ein Faktor, der von der Position `P` des Wortes `W` im Text `T` abhängt
- $I_C(W)$: Anzahl der Texte mit mindestens einer Instanz des Wortes `W`
- W_i: `i`-tes Vorkommen des Wortes `w` in Text `T` (Laufvariable)
- `I`: obere Schranke für den Anteil der Texte, in denen `W` vorkommen darf, ohne Stoppwort zu werden
- `S`: obere Schranke für das Vorkommen, die `W` in `C` haben darf , ohne Stoppwort zu werden

Bei gleichzeitigem Überschreiten von `I` und `S` ist `W` ein Element der Stoppwortliste `SWL`. Eine Stoppwortliste `SWL` ist eine Menge von Worten. Alle Worte aus `SWL` werden im Rahmen der Textanalyse nicht weiter betrachtet.

Es gilt formal:

$$w(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} \left(N_c(W_i) \right)} \geq S \right) \wedge \left(\frac{I_C(W)}{|C|} \geq I \right) \\ \sum_{i=1 \ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

Im Gegensatz zu anderen TF-IDF-Derivaten wird bei CRIC also der IDF-Anteil nicht als Faktor in die Wortgewichtung einbezogen, sondern quasi als Vorfilter eingesetzt. Nur Worte, die diesen Vorfilter passieren, werden dann mit dem TF-Anteil gewertet.

Eine genauere Betrachtung der Unterschiede zwischen klassischem TF-IDF und dem CRIC-Derivat folgt weiter unten (siehe 6.1.2).

Es muss eine Worterkennung (Token-Bildung) auf Text `T` durchgeführt werden, um dies anwenden zu können. Im Zuge der Worterkennung wird die Stoppwortliste `SWL` um `W` erweitert, wenn `w(W)=0` gilt.

6.1.1.4 Token-Bildung

Bei der Wort-Erkennung wird der Text `T` Zeichen für Zeichen durchlaufen. Dabei werden alle Großbuchstaben in Kleinbuchstaben gewandelt. Worte werden anhand von begrenzenden nicht alphanumerischen Zeichen (Leerzeichen, Satzzeichen et cetera) erkannt.

Ein besonderes Problem bei der Worterkennung stellt die Behandlung des "Bindestrich" dar. Trifft die Analyse auf einen Bindestrich, so wird geprüft, ob davor ein Buchstabe steht. In diesem Fall wird postuliert, dass es sich um eine mit Bindestrichen verbundene Wortfolge oder einen typographischen Trennungsstrich und nicht um eine Aufzählung oder ähnliches handelt. Steht ein Buchstabe vor dem Bindestrich, wird wie folgt verfahren:

- der Bindestrich wird durch ein Leerzeichen ersetzt
- die beiden durch den Bindestrich bisher getrennten Worte werden zusätzlich als konkateniertes neues Wort eingefügt

Bei `n` Bindestrichen in einem Wort wird das Verfahren für alle `1,2,…,n+1` Wort-Konkatenationen durchgeführt, um Worte die durch typographische Trennung zerlegt wurden garantiert als Einzelwort zusammenzuführen.

Aus der Wort-Folge

```
"Text-Verarbeitung"
```

wird also

```
"text verarbeitung textverarbeitung"
```

und aus der typographisch dreifach getrennten Wort-Folge

```
"Text-Ver-arbeit-ung"
```

wird im ersten Schritt (2-Wort Konkatenation)

```
"text ver arbeit ung
Textver verarbeit arbeitung"
```

und dann (3-Wort Konkatenation)

```
"text ver arbeit ung
Textver verarbeit arbeitung
textverarbeit verarbeitung"
```

sowie schließlich (4-Wort Konkatenation)

```
"text ver arbeit ung
Textver verarbeit arbeitung
textverarbeit verarbeitung
textverarbeitung"
```

Diese Sonderfallbehandlung für durch Bindestrich getrennte Worte gilt allerdings nur für die Bestimmung der Schlüsselworte eines Textes. In die Stoppwortliste `SWL` gehen nur die komplett konkatenierten Worte ein[10].

Hat das gefundene Wort weniger als zwei Buchstaben wird es entfernt. Bei sehr langen Worten werden nur die ersten 253 Zeichen übernommen und die weiteren ignoriert.

Jedes erkannte Wort eines Textes wird als neues Element in ein Array "Textworte" aufgenommen oder als bestehendes Element im Array "Textworte" inkrementiert. In „Textworte“ steht daher für jedes Wort der Zähler über das Vorkommen.

6.1.1.5 Textstruktur und Häufigkeit

Wenn ein Wort am Anfang oder Ende eines Textes `T` lokalisiert ist, wird es höher bewertet. Diese Gewichtung trägt der Bedeutungsdynamik in Texten Rechnung, nach der wichtige Inhalte zumeist am Anfang (Titel, Einleitung) und am Ende (Zusammenfassung, Fazit) eines Textes zu finden sind.

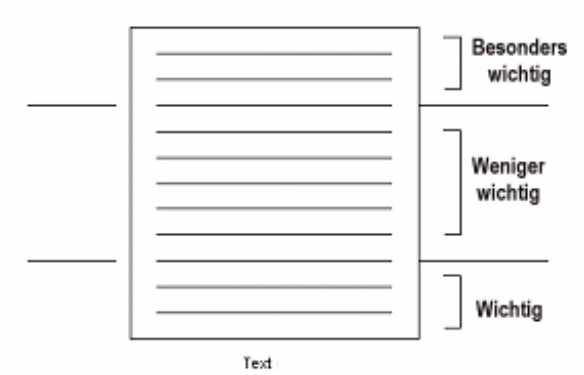


Abbildung 73:Die Position eines Wortes im Text hilft bei der Relevanzbewertung als Schlüsselwort

Die Verlaufskurve der Bedeutung durch alle Texte einer Textbasis `C` wäre vermutlich eine Kurve folgender Gestalt:

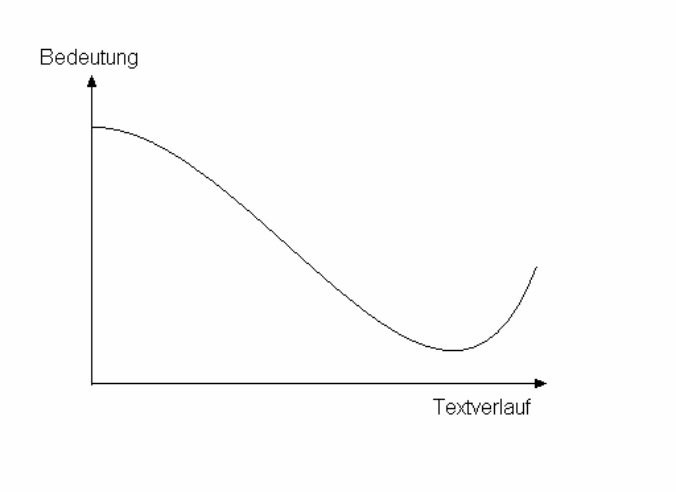


Abbildung 74:Der Bedeutungsverlauf gemittelt über alle Texte einer Textbasis dürfte in der Regel einer Kurve mit langsamen Abfall (nach Titel und Vortext) und nochmals steigendem Anstieg zum Ende (Zusammenfassung) sein.

[10] Dabei wird postuliert, dass das Wort in anderen Texten in ununterbrochener Form vorkommt und so in die Wortliste aufgenommen wird.

Nach zahlreichern Tests hat sich die folgende Aufteilung mit den angegebenen Gewichtungen als besonders effizient erwiesen:

- Position liegt in den ersten 20 Prozent des Textes: Gewichtung "4,5"
- Position liegt nach den ersten 20 und vor den letzten 10 Prozent des Textes: Gewichtung "2"
- Position liegt in den letzten 10 Prozent des Textes: Gewichtung "3

Damit wird der Bedeutungsverlauf durch eine Treppenfunktion approximiert:

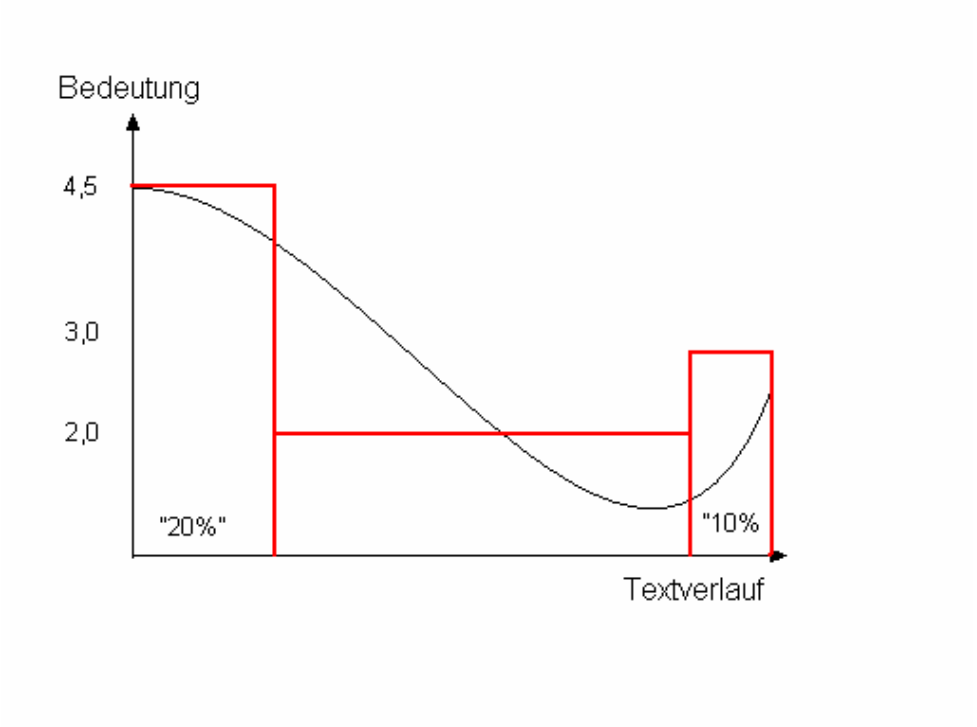


Abbildung 75:Approximation des Bedeutungsverlaufes mit einer Treppenfunktion

Dadurch muss ein Wort im "mittleren" Textteil beispielsweise dreimal vorkommen, um ein Vorkommen im Anfangsteil zu übertreffen.

Ist der Text sehr kurz (kleiner als 100 Zeichen), wird keine Aufgliederung in Bewertungsbereiche vorgenommen, sondern der komplette Text wie ein Textanfang (Gewichtung "4,5") gewertet. Ist der Text sehr groß (größer als 10.000 Zeichen) wird der Textanfang fix auf 2.000 Zeichen und das Ende fix auf 1.000 Zeichen gesetzt. Formal ergibt sich für das Wort `W` damit eine Gewichtung als Summe über alle Vorkommen des Wortes `W` im Text `T` wie folgt:

$$\sum_{i=1\ldots N_T(W)} P_T(W_i)$$

mit

$$P_T(W_i) = \begin{cases} 4{,}5 & falls \quad Pos_T(W) \le |T| * 0{,}20 \\ 3 & falls \quad Pos_T(W) > |T| * 0{,}90 \\ 2 & sonst \end{cases}$$

wobei

$Pos_T(W)$ = Position (in Zeichen) von `W` in `T` und

$|T|$ =Anzahl der Zeichen in Text `T`

Ein Beispiel soll den Einfluss der Position und Häufigkeit eines Wortes im Text T illustrieren. Es sei der folgende Text T gegeben:

> Der Titel eines Textes ist besonders wichtig
>
> Denn im Titel eines Textes versucht der Autor das Thema des Textes in wenigen Worten zu beschreiben. Auch der Vortext hat oft eine besondere Bedeutung, da er in der Regel den Inhalt eines Textes zusammenfasst. Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann einmal etwas anderes in Form eines neuen Begriffes der mit dem Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum". Hier steht dann etwas Fülltext in Form von "lore ipsum".
>
> Der eigentliche Text hat dann normalerweise weniger Relevanz. Allerdings ändert sich das gegen Ende des Textes wieder. Denn dort zieht der Autor oftmals ein zusammenfassendes Fazit.

Dieser Beispieltext enthält 1.232 Zeichen woraus folgt:

- *Position liegt in den ersten 246 Zeichen des Textes: Gewichtung "4,5"*
- *Position liegt zwischen 247 und 1109 Zeichen des Textes: Gewichtung "2"*
- *Position liegt nach 1109 Zeichen des Textes: Gewichtung "3*

Für die Worte "Textes" (W_1) und "Begriffes" (W_2) gilt:

$w(W_1)=4*4,5+1*3=21$

$w(W_2)=1*2=2$

Damit wird W_1 zehnfach höher bewertet als W_2 und stellt eines der Schlüsselworte des Beispieltextes dar.

6.1.1.6 Empirische Untersuchungen zur Gewichtung anhand der Textposition

Die in 6.1.1.5 beschriebene Heuristik kann durch empirische Untersuchungen gestützt werden. So lies Baxendale [BAX1958] Probanden den aus Ihrer Sicht jeweils repräsentativen Satz aus 200 Texten bestimmen. Die Selektion lieferte in 85 Prozent aller Fälle den ersten und in sieben Prozent aller Fälle den letzten Satz.

Edmundson [EDM1969] verwendet ein automatisches Verfahren, um die wichtigsten Sätze aus Texten zu selektieren. Dieses bewertete den ersten und den letzten Satz besonders hoch. Die Übereinstimmung mit manuell selektierten "wichtigsten Sätzen" lag in zwei Untersuchungen bei 40 beziehungsweise 53 Prozent.

Auch neuere Arbeiten [LIN1997] stärken die in CRIC verwendete Wortgewichtung anhand der Textposition. Besonders interessant an dieser Untersuchung ist das Ergebnis, dass nur 30 Prozent der Worte, die Menschen in manuell verfassten Zusammenfassungen verwenden, nicht im eigentlichen Text vorkommen.

Daraus schließen Lin und Howy, dass maximal 30 Prozent des "Themas" eines Textes durch einen menschlichen Inferenz-Prozess abgeleitet werden.

Damit können potenziell bis zu 70% der relevanten Worte von einem Algorithmus, der auf der Selektion von Worten aus einem Text basiert, erfasst werden. Diese Prozentzahl ist eine Untergrenze, da in den 30% der nicht enthaltenen Worte auch einfache Synonyme et cetera enthalten sind.

6.1.1.7 Stoppwortliste (SWL)

Sei W ein Wort mit $I_C(W)>0$, dann ist W kein Stoppwort ($W \notin SWL$) der Stoppwortliste SWL, wenn die Anzahl der Instanzen von W in Texten aus C ($N_C(W)$) kleiner als ein bestimmter Prozentsatz "S" bezogen auf das meist

vorkommende Wort; max(N$_C$(W$_i$)) für alle W$_i$ aus C ist oder die Anzahl der Texte aus C in denen W vorkommt (I$_C$(W)) kleiner als ein bestimmter Prozentsatz "I" ist. Andernfalls ist W ein Stoppwort und wird sofern noch nicht vorhanden zur Stoppwortliste SWL hinzugefügt. Es gilt:

$$w(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq S \right) \wedge \left(\frac{I_C(W)}{|C|} \geq I \right) \\ \sum_{i=1\ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

Die Stoppwortliste SWL wird um W erweitert, wenn w(W)=0 gilt. Der "Stoppwert" S wird in Prozent der maximalen Anzahl der Instanzen eines Wortes in C im Verhältnis zur Anzahl des Wortes mit der größten Häufigkeit in C definiert. Als geeigneter Wert für S hat sich 1,0 % erwiesen.

Für I, den maximalen Prozentsatz aller Texte aus C in denen ein Wort vorkommen darf, um kein Stoppwort zu sein, wird "3,0" % verwendet.

Damit gilt dann:

$$w(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq 0{,}01 \right) \wedge \left(\frac{I_C(W)}{|C|} \geq 0{,}03 \right) \\ \sum_{i=1\ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

Die Stoppwortliste SWL wird dynamisch erzeugt, wenn ein Text oder mehrere Texte in die Textbasis C eingefügt werden. Individuell erzeugte Stoppwortlisten mit einer großen Hürde für die Aufnahme neuer Begriffe sind effizienter als statische, umfangreiche Stoppwortlisten. Denn mit jedem eliminierten Wort geht auch ein Teil der Semantik verloren [RIL1995].

Nur eine individuelle Stoppwortliste berücksichtigt die Ausprägung der Textbasis C. So ist beispielsweise der Name eines Unternehmens ("Maier KG") in der Textbasis eben dieses Unternehmens sehr wahrscheinlich ein Stoppwort, das in einem Großteil der Texte vorkommt (E-Mail Signatur, Fußzeile in Dokumenten et cetera), in einer generischen Stoppwortliste aber nicht vertreten wäre.

Das dynamische Vorgehen beim Aufbau von SWL reduziert allerdings die Qualität der Schlüsselworte der Texte, die als einzelne Texte in einer frühen Phase des Aufbaus von C eingefügt werden. Denn zu diesem Zeitpunkt ist SWL noch sehr klein und daher wenig repräsentativ.

Die Qualität der Schlüsselworte dieser Texte lässt sich durch eine Funktion zum Einfügen von mehreren Texten in „Paketen“ verbessern. Für alle Texte des Paketes wird SWL aktualisiert, bevor die Schlüsselworte der einzelnen Texte bestimmt werden. Die ersten Texte in C sollten daher in einem möglichst großen Paket eingefügt werden.

6.1.1.8 Zipf´s Gesetz und der Parameter "S"

Der Parameter S der oben beschriebenen Stoppwortbildung kann auf dem theoretischen Fundament des Zipf´schen Gesetzes analysiert und der gewählte Wert "1,0" damit untermauert werden.

Das Zipf´sche Gesetz [ZIP1949] besagt, dass die Frequenz (Anzahl der Instanzen) eines Wortes in einem Korpus C umgekehrt proportional zu seinem Rang in der Rangfolge der Frequenzen aller Worte ist. Formal ergibt dies:

$$N_C(W) * Rang(W) = k$$

Mit k als korpusabhängiger Konstante. Daraus folgt offensichtlich

$$N_C(W) = \frac{k}{Rang(W)}$$

Daraus lässt sich auch ableiten, dass relativ wenige Worte einen großen Anteil aller Texte ausmachen:

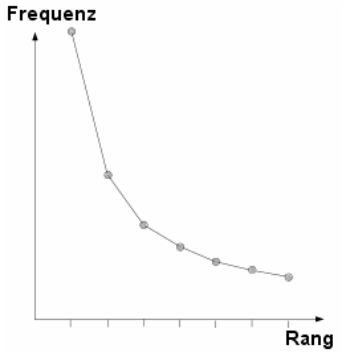


Abbildung 76:Schematisierte Zusammenhänge von Rang/Frequenz und Rang/Textanteil

Das idealisierte Verhältnis zwischen Rang und Frequenz approximiert die Realität besonders dann gut, wenn die Textbasis C sehr groß ist; also beispielsweise eine natürliche Sprache repräsentiert. Auf Basis der Analyse des Projektes "Deutscher Wortschatz" der Universität Leipzig [HEY2005] ergibt sich folgender Vergleich des Zipf´schen Gesetzes zur Realität der deutschen Sprache. Dabei wurden über 18 Millionen Wortinstanzen berücksichtigt:

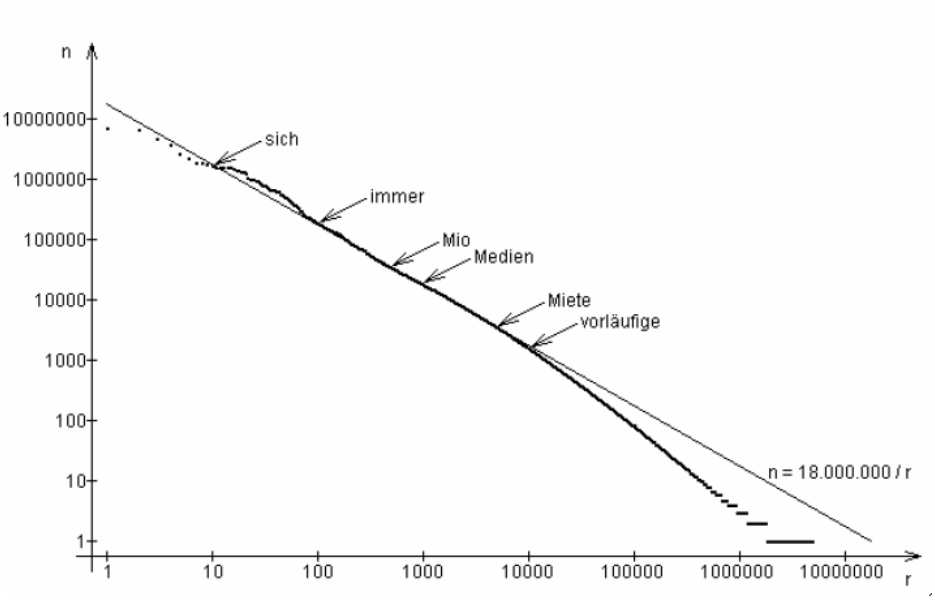


Abbildung 77:Korrelation von Rang und Frequenz nach Zipf´schen Gesetz und in der deutschen Sprache (logarithmische Achsenskalierung) aus [HEY2005]

Man sieht, dass Zipf´sches Gesetzt und reale Korrelation zwischen Rang und Frequenz eines Wortes im Bereich mittlerer Frequenzen sehr gut ist. Allerdings existieren im unteren und oberen Bereiche deutliche Abweichungen. Diese Abweichungen können bei weniger großen und thematisch stärker fragmentierten Textbasen noch stärker ausfallen.

Das CRIC Verfahren nutzt zur Erkennung von Stoppworten unter anderem die Frequenz von Worten. Dies wird durch den Teil

$$\frac{N_C(W)}{\max_{I_C(W_i)>0}\left(N_c(W_i)\right)} \geq 0{,}01$$

der Stoppwort-Formel erreicht. Alle Worte, deren Frequenz oberhalb von einem Prozent liegt – im Verhältnis zum Wort mit der größten Frequenz – sind betroffen. Diese sind dann zumindest Kandidaten für Stoppworte. Zusätzlich müssen diese allerdings auch noch in einer bestimmten Anzahl von Texten vorkommen.

Der oben dargstellte Anteil der CRIC Formel, der auf die Frequenz abstellt, kann mit dem Zipf˝schen Gesetz wie folgt umgeformt werden:

$$\frac{N_C(W)}{\max_{I_C(W_i)>0}\left(N_c(W_i)\right)} \geq 0{,}01$$

$$= \quad \frac{\frac{k}{Rang(W)}}{\max_{I_C(W_i)>0}\left(N_c(W_i)\right)} \geq 0{,}01$$

Nach Zipf besitzt das Wort mit der größten Frequenz den Rang 1. Damit folgt weiter:

$$= \quad \frac{\frac{k}{Rang(W)}}{\frac{k}{1}} \geq 0{,}01 \qquad = \quad \frac{k}{Rang(W)} * \frac{1}{k} \geq 0{,}01$$

$$= \quad \frac{1}{Rang(W)} \geq 0{,}01$$

Im idealisierten Fall blendet CRIC also die ersten 100 Worte aus (wenn auch die zweite Stoppwort Bedingung in Form der erforderlichen Zahl von Texten, welche die Worte enthalten müssen, zutrifft). Je weiter die Frequenz des häufigsten Wortes allerdings vom idealisierten Wert nach unten abweicht, desto mehr Worte werden ausgeblendet. Insbesondere bei kleineren Textbasen sind so schnell über 1.000 Worte betroffen.

Da diese Worte einen signifikanten Textanteil ausmachen, aber aufgrund ihrer Häufigkeit kaum einen Beitrag zur Diskriminierung der Texte leisten können, ist es folglich sinnvoll, diese aus dem Selektionsprozess auszuschließen.

6.1.1.9 Inverse Document Frequency und der Parameter I

Neben der Frequenz der Worte spielt bei CRIC für die Stoppwort Eigenschaft auch die Anzahl der Texte, in denen die Worte vorkommen, eine Rolle. Formal ist dies der folgende Teil der Formel:

$$\frac{I_C(W)}{|C|} \geq 0{,}03$$

Es können also nur solche Worte Stoppworte werden, die in mindestens drei Prozent aller Texte vorkommen. Ein Vergleich zum klassischen IDF (inverse document frequency) liegt nahe, da der IDF-Anteil des TF-IDF Verfahrens

$$\log \frac{|C|}{I_C(W)}$$

dem oben angeführten Formelteil recht ähnlich sieht. Allerdings wird der IDF-Anteil als Faktor verwendet, wohingegen der CRIC Formelteil bei Überschreitung des Schwellwertes von 0,03 Prozent – bei gleichzeitiger Überschreitung des Parameters `S` – das Wort zum Stoppwort macht.

Betrachtet man beide Formeln als Funktionen über $I_C(W)$ bei konstantem $|C|$, so ergibt sich mit $|C|=10.000$ beispielsweise folgende Grafik für $I_C(W)=[1,600]$:

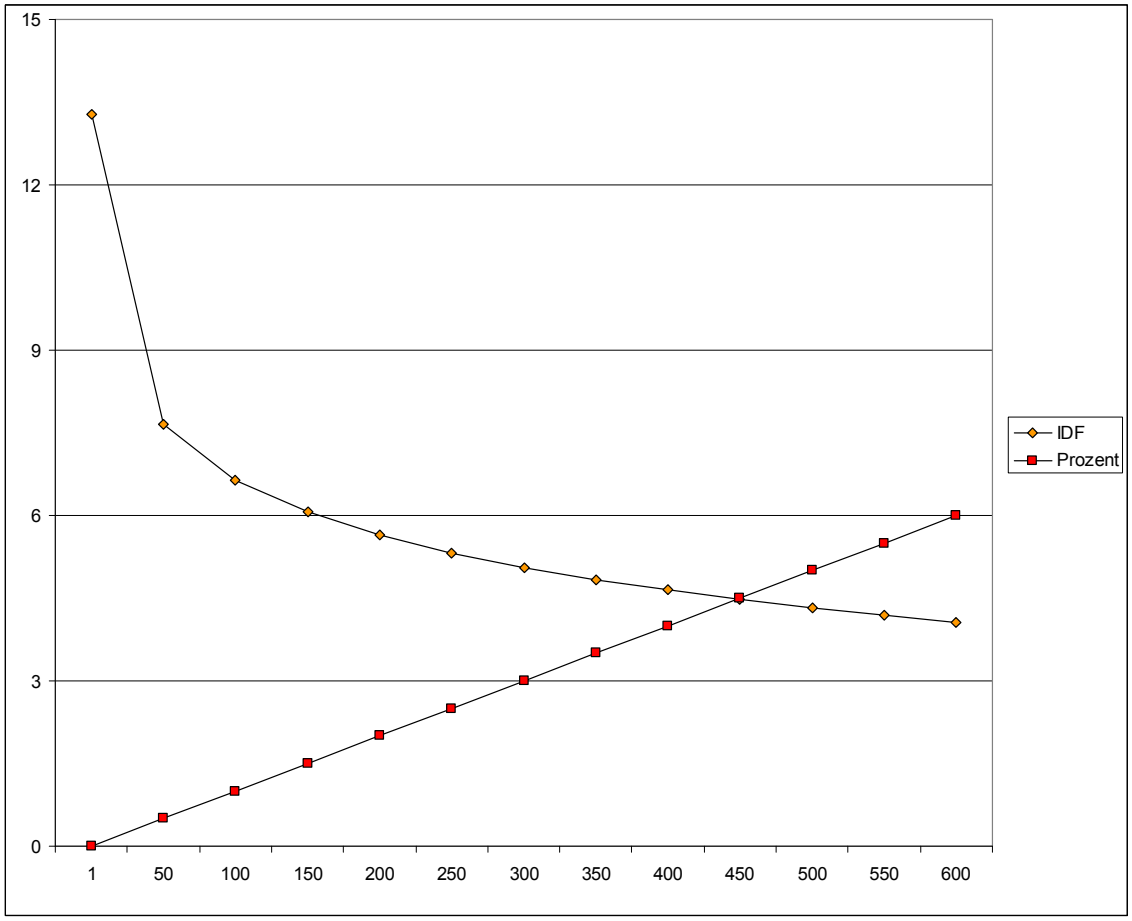


Abbildung 78:Klassisches IDF und der CRIC IDF-Anteil der Stoppwortbildung im Vergleich (y-Achse: Anzahl der Texte, die das betrachtete Wort enthalten; x-Achse: Prozentzahl der Texte [Prozent] beziehungsweise IDF-Wert [IDF)

Beim IDF Verfahren werden Worte, die in weniger als der Hälfte aller Texte vorkommen, aufgewertet, wegen

$$\log 2\left(\frac{|C|}{\frac{|C|}{2}}\right)=1$$

Die Aufwertung erfolgt umso stärker, je weniger Texte das Wort enthalten. Zwar federt der Logarithmus die positive Bewertung von Worten, die in wenigen Texten enthalten sind, etwas ab. Bei großen Textbasen von einer Million und mehr Texten wird aber |C| auch durch den Logarithmus nur unzureichend eingegrenzt, wenn ein Wort beispielsweise in weniger als 100 Texten vorkommt:

$$\log_2 \frac{1.000.000}{100} \approx 13{,}29$$

Dadurch werden Worte, die in wenigen Texten großer Textbasen vorkommen sehr stark aufgewertet. Und zwar unabhängig davon, wie oft die Worte innerhalb der Texte selbst enthalten sind.

Damit würde beispielsweise in einer kleinen Kollektion von 100 Texten eines Spezialthemas (innerhalb eines großen Korpus, der dieses Thema sonst nicht behandelt) faktisch alle speziellen Worte des Themas aufgewertet und deren Verteilung innerhalb der 100 Texte quasi unberücksichtigt bleiben. Damit entfällt die IDF-Komponente in solchen Fällen quasi.

Der CRIC Formelteil bleibt hingegen auch bei großen Textbasen unanfällig für solche Konstellationen, da unterhalb des Schwellwertes von drei Prozent (der nur greift, wenn auch die Bedingung für den `S`-Parameter zutrifft) alle Worte gleich behandelt werden.

Es sei noch erwähnt, dass es zahlreiche Arbeiten gibt, die versuchen den IDF-Anteil des TF-IDF Verfahrens auf eine theoretische Basis zu stellen. In [ROB2004] wird ein sehr guter Überblick gegeben. Robertson kommt zum Schluss, dass ein theoretische Bezug des IDF Ansatzes zu Shannons Informationstheorie – den viele Arbeiten herstellen – problematisch ist. Dies ist der Fall, weil Shannons *Nachrichten* den Worten und nicht den Dokumenten entsprechen. Der IDF Ansatz jedoch den Ereignisraum der Dokumente betrachtet.

IDF und das Probabilistische Ranking Prinzip

Das Probabilistische Ranking Prinzip (*PRP*) [ROB1977] ordnet aufgrund einer Anfrage `Q` alle Dokumente eines Korpus `C` absteigend nach ihrer Relevanz in Bezug auf die Anfrage `Q` an.

In Anlehnung an [SIN2001] soll im Folgenden der Zusammenhang zwischen IDF und PRP aufgezeigt werden.

Es sei `P(R|D)` die Wahrscheinlichkeit für die Relevanz des Dokumentes D.

Statt P(R|D) kann man

$$\log \frac{P(R|D)}{P(\overline{R}|D)}$$

betrachten, da diese "log-odds" genannte Transformation, die Ordnung des PRP aufrecht erhält[11]. Die transformierte Formel kann als Maß für die Ähnlichkeit von Q und D interpretiert werden:

$$\log \frac{P(\text{' D ist relevant für Q'})}{P(\text{D ist nicht relevant für Q})}$$

Mit dem Bayes´schen Theorem folgt

$$\log \frac{P(R|D)}{P(\overline{R}|D)} = \log \frac{\frac{P(D|R)*P(R)}{P(D)}}{\frac{P(D|\overline{R})*P(\overline{R})}{P(D)}} = \log \frac{P(D|R)*P(R)}{P(D|\overline{R})*P(\overline{R})}$$

Da P(R) unabhängig vom Dokument beziehungsweise für alle Dokumente konstant ist, kann dies zu

$$\log \frac{P(D|R)}{P(D|\overline{R})}$$

umgeformt werden, ohne die Ordnung der Dokumente zu verändern. In der Abschätzung von `P(D|R)` unterscheiden sich die verschiedenen probabilistischen Methoden.

---

[11] Es ist zu zeigen, dass aus $\log \frac{p_1}{1-p_1} < \log \frac{p_2}{1-p_2}$ folgt, dass $p_1 < p_2$. Sei $\log \frac{p_1}{1-p_1} < \log \frac{p_2}{1-p_2}$ gegeben, dann kann der `log` (da streng monoton) entfallen. Es folgt: $\frac{\frac{p_1}{1-p_1}}{\frac{p_2}{1-p_2}} < 1$, was äquivalent zu $\frac{p_1*(1-p_2)}{p_2(1-p_1)} < 1$ ist und zu $\frac{p_1 - p_1*p_2}{p_2 - p_1*p_2} < 1$ umgeformt werden kann. Daraus wird $p_1 - p_1*p_2 < p_2 - p_1*p_2$ und durch Addition von p1*p2 auf beiden Seiten dann $p_1 < p_2$, q.e.d.

Nimmt man an, die Worte eines Dokumentes seien in Bezug auf die Wahrscheinlichkeit ihres Auftretens in einem Dokument unabhängig voneinander, dann kann `P(D|R)` als Produkt der einzelnen Wortwahrscheinlichkeiten dargestellt werden:

$$P(D|R) = \prod_{W_i \in Q,D} P(W_i|R)$$

Wobei `P(Wi|R)` als die Wahrscheinlichkeit des Wortes `Wi`, in einem relevanten Dokument vorzukommen, interpretiert werden kann.

Damit kann $\log \frac{P(D|R)}{P(D|\overline{R})}$ wie folgt dargestellt werden:

$$\log \frac{\prod_{W_i \in Q,D} P(W_i|R)}{\prod_{W_i \in Q,D} P(W_i|\overline{R})} = \log \prod_{W_i \in Q,D} \frac{P(W_i|R)}{P(W_i|\overline{R})} = \sum_{W_i \in Q,D} \log \frac{P(W_i|R)}{P(W_i|\overline{R})}$$

Da `P(Wi|R)` nicht bekannt ist, muss es abgeschätzt werden. Nimmt man Croft und Harper folgend [CRO1997] an, dass fast alle Dokument eines Korpus `C` zu einer Anfrage `Q` irrelevant sind (was bei einem großen Korpus plausibel erscheint), kann man wegen

$$P(W_i|R) = 1 - P(W_i|\overline{R})$$

wie folgt abschätzen:

$$P(W_i|R) \approx 1 - \frac{I_C(W_i)}{|C|}$$

Daraus folgt dann

$$\sum_{W_i \in Q,D} \log \frac{P(W_i|R)}{P(W_i|\overline{R})} \approx \sum_{W_i \in Q,D} \log \frac{1 - \frac{I_C(W_i)}{|C|}}{\frac{I_C(W_i)}{|C|}} = \sum_{W_i \in Q,D} \log \frac{\frac{|C|}{|C|} - \frac{I_C(W_i)}{|C|}}{\frac{I_C(W_i)}{|C|}}$$

$$= \sum_{W_i \in Q,D} \log \frac{\frac{|C| - I_C(W_i)}{|C|}}{\frac{I_C(W_i)}{|C|}} = \sum_{W_i \in Q,D} \log \frac{|C| - I_C(W_i)}{I_C(W_i)}$$

was dem IDF-Anteil des TF-IDF Verfahrens sehr ähnlich ist:

$$\log \frac{|C|}{I_C(W)}$$

Das Probabilistische Ranking Prinzip und RSJ

Robertson [ROB2004] führt die Robertson-Sparck-Jones Wortgewichtung (RSJ; siehe auch Seite 26) auf das Probabilistische Ranking Prinzip zurück. Dies wird erreicht, indem man RSJ ohne Relevanzinformation betrachtet. Formal wird die RSJ Wortgewichtung `wi` für ein Wort `Wi`

$$w_i = \log \frac{\left(\frac{r_i + 0{,}5}{R - r_i + 0{,}5} \right)}{\left(\frac{n_i - r_i + 0{,}5}{N - n_i - R + r_i + 0{,}5} \right)}$$

mit

N = Größe der Textbasis C

n_i = Anzahl der Texte in C die das Wort W_i enthalten

R = bekannte Dokumente die zu einem *topic* relevant sind

r_i = Anzahl der als relevant ausgezeichneten Texte die das Wort W_i enthalten

durch setzen von r_i und R auf 0 zu

$$w_i = \log \frac{\left(\frac{0+0,5}{0-0+0,5} \right)}{\left(\frac{n_i - 0 + 0,5}{N - n_i - 0 + 0 + 0,5} \right)}$$

$$\Leftrightarrow w_i = \log \frac{1}{\left(\frac{n_i + 0,5}{N - n_i + 0,5} \right)}$$

$$\Leftrightarrow w_i = \log \frac{N - n_i + 0,5}{n_i + 0,5}$$

Damit lässt sich RSJ lässt offenbar auf PRP zurückführen. Der Unterschied von RSJ und PRP besteht lediglich in den Konstanten "0,5".

RSJ und IDF

In [ROB1997] führen Robertson und Walker außerdem aus, dass das "$-n_i$" im Zähler bei RSJ zu entfernen ist, da es in Korrelation zum PRP durch die Annahme eines konstanten P(W|R) entsteht, P(W|R) aber mit der Frequenz eines Wortes wachsen muss.

Damit wird RSJ dann zu

$$RSJ : w_i = \log \frac{N + 0,5}{n_i + 0,5}$$

Was bis auf die Konstanten "0,5" in Zähler und Nenner IDF entspricht. Somit lässt sich IDF auch auf RSJ zurückführen.

6.1.1.10 Wortliste WL und Wort-Text-Attribut "SWORTE"

Die Parameter S und I in den Formeln oben erfordern Daten zu allen Worten und Texten aus C. Es wird daher eine Datenstruktur benötigt, die nicht nur die Anzahl der Instanzen der Worte insgesamt (Wortliste WL; "Anzahl Instanzen"), sondern auch die Anzahl der Instanzen der Worte in den einzelnen Texten (Wortliste WL; "Anzahl Texte") aufnimmt. Des Weiteren müssen für jeden Text noch die Charakteristika in Form der Schlüsselworte verwaltet werden. Da dies eine feste Anzahl an Worten ist (siehe 6.1.2 unten) und diese Worte auch nur en bloc gelesen und geschrieben werden, ist ein Attribut ("SWorte") deutlich effizienter als eine Wort-Text-Relation.

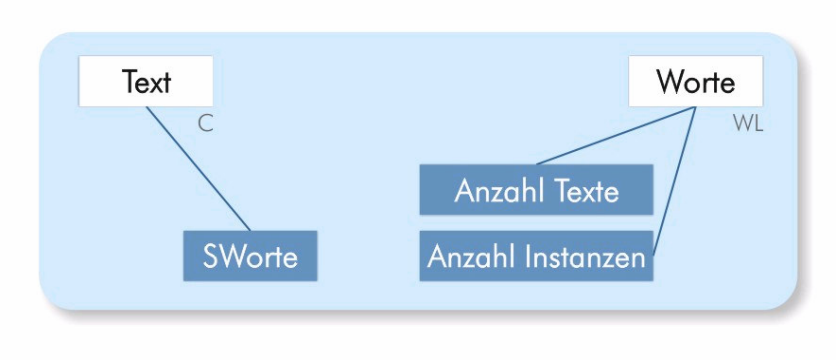


Abbildung 79:CRIC benötigt nur einen vereinfachten inversen Index. Eine Wort-Text-Relation ist nicht erforderlich, da nur die Anzahl der Texte in denen ein Wort vorkommt und die Gesamtzahl der Instanzen eines Wortes relevant sind. Diese Werte können bei neuen, geänderten und gelöschten Texten relativ verändert werden (inkrementiert und dekrementiert).

6.1.2 CRIC und TF-IDF

Das TF-IDF-Derivat in CRIC besteht aus zwei Teilen. Der TF-Anteil wird erst dann angewendet, wenn ein Wort den IDF-Anteil, der wie ein Filter wirkt, passiert hat. Im Folgenden sollen das klassische TF-IDF Verfahren und das CRIC-Derivat verglichen werden.

6.1.2.1 Entstehungsgeschichte von TF-IDF

Heute versteht man unter dem TF-IDF Ansatz die folgende Wortgewichtung:

$W = TF * IDF$ mit $TF = N_T(W)$ und $IDF = \log \frac{|C|}{I_C(W)}$

mit

$N_T(W)$ = Anzahl der Instanzen ("Frequenz") von Wort W im Text T

$I_C(W)$ =Anzahl der Texte in der Textbasis $C = \{T_1, \ldots, T_n\}$ mit mindestens einer Instanz von W

$|C|$ =Anzahl der Texte der Textbasis C

Entstanden sind TF [LUH1958] und IDF [SPA1972] aber zunächst unabhängig. Erst im Nachgang wurden dann Kombinationen der beiden empirisch erprobt [SAL1973], [SAL1975], [WHU1981].

6.1.2.2 Der TF-Anteil

Der TF-Anteil beim klassischen TF-IDF ($TF_k(W)$) ist eine einfache Wertung in Form der Anzahl eines Wortes in einem Text:

$$TF_k(W) = N_T(W)$$

Bei CRIC wird im TF-Anteil ($TF_C(W)$) zusätzlich die Position der Worte im Text berücksichtigt:

$$TF_C(W) = \sum_{i=1 \ldots N_T(W)} P_T(W_i)$$

mit

$$P_T(W_i) = \begin{cases} 4{,}5 & falls \quad Pos_T(W) \le |T| * 0{,}20 \\ 3 & falls \quad Pos_T(W) > |T| * 0{,}90 \\ 2 & sonst \end{cases}$$

wobei gilt:

$Pos_T(W_i)$ = Position (in Zeichen) von W_i in T

$|T|$ =Anzahl der Zeichen in Text T

6.1.2.3 Der IDF-Anteil

Das ursprüngliche IDF ($IDF_k(W)$) wertet Worte, die in sehr vielen Texten vorkommen, ab. Das Vorkommen innerhalb der einzelnen Texte wird dabei nicht bewertet:

$$IDF_k(W) = \log \frac{|C|}{I_C(W)}$$

Der IDF-Anteil von CRIC ($IDF_C(W)$) berücksichtigt neben dem Vorkommen in einer bestimmten Anzahl von Texten auch die Anzahl des Vorkommens insgesamt:

$$IDF_C = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq 0{,}01 \right) \wedge \left(\frac{I_C(W)}{|C|} \geq 0{,}03 \right) \\ 1 & sonst \end{cases}$$

mit

- $N_C(W)$: Anzahl der Instanzen des Wortes W in der Textbasis C mit $C=\{T_1, \ldots, T_n\}$
- $I_C(W)$: Anzahl der Texte mit mindestens einer Instanz des Wortes W

Bei Überschreiten von I und S fungiert der IDF-Anteil als Filter, der unabhängig vom TF-Anteil dafür sorgt, dass ein Wort nicht selektiert wird.

Diesem Vorgehen liegt die Annahme zugrunde, dass Worte, die mit einer bestimmten Häufigkeit in Texten beziehungsweise der Textbasis vorkommen, generell nicht zur Charakterisierung eines Textes geeignet sind.

Bei der Entwicklung des IDF-Ansatzes war, wie Karen Spärck-Jones ausführt [SPA2004], die Unterstützung von manuellen Suchanfragen ein wesentlicher Aspekt. Da Menschen auch häufig vorkommende Worte bei der Suche verwenden, dürfen diese nicht komplett eliminiert, sondern nur abgewertet werden. Durch letzteres erlangen weniger häufig auftretende Worte dann die erforderliche Relevanz, um aus vielen potenziellen "Treffern" der Suchanfrage die vermeintlich gewünschten Texte zu liefern.

Im CRIC Anwendungsszenario werden keine unwichtigen (also dem IDF-Ansatz folgend häufig vorkommende) Worte als Anfragen verwendet. Daher wird der IDF-Ansatz bei CRIC als Filter eingesetzt.

6.1.2.4 Die Kombination beider Ansätze

Beim klassischen Verfahren ergibt die Kombination der beiden Ansätze ($TFIDF_k(W)$) die folgende Wertungsformel:

$$TFIDF_k(W) = N_T(W) * \log \frac{|C|}{I_C(W)}$$

Demgegenüber sieht die CRIC-Kombination wie folgt aus:

$$TFIDF_C(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq 0{,}01 \right) \wedge \left(\frac{I_C(W)}{|C|} \geq 0{,}03 \right) \\ TF_C(W) = \sum\limits_{i=1 \ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

Der wesentliche Unterschied der beiden Verfahren besteht - neben der TF-Herleitung über die Wortposition bei CRIC - darin, dass die Wertungsfunktion von CRIC neben der Anzahl der Texte, in denen ein Wort vorkommt, auch die Frequenz des Wortes über alle Texte berücksichtigt.

6.1.3 Verwandte Texte ermitteln

Nachdem die Schlüsselworte eines Textes T ermittelt und bezüglich ihrer Gewichtung absteigend sortiert sind, werden die ersten m Schlüsselworte $W_1, \ldots, W_m$ (die vermeintlich bedeutsamsten) als Anfrage für eine Volltextsuche auf einem Datenbanksystem verwendet. Dazu muss die Textbasis C als Volltext im betreffenden Datenbanksystem gehalten werden. Weil es ein zweischneidiges Schwert darstellt (Ambiguierung durch

Wortstammreduktion; [KAN2000][HUL1996]) und Suchalgorithmen in Datenbanksystemen das Stemming ohnehin berücksichtigen, wurde bewusst nicht in das CRIC Verfahren zur Erzeugung der Schlüsselworte integriert.

Die Anzahl der Suchworte

Tests haben gezeigt, dass der optimale Wert für `m` sehr stark vom verwendeten Datenbanksystem abhängt (insbesondere von der Recall-Quote der Volltextsuche). Allerdings lagen alle Werte in einem Intervall von 5 bis 10 Schlüsselworten. Bei dem in der Evaluation eingesetzten Datenbanksystem *MySQL* wurde der Wert auf "10" gesetzt. Tendenziell steigert ein großes "m" die *Precision*, senkt aber die *Recall* Quote (siehe 5.1). Die Wahl des Wertes stellt also einen Kompromiss dar. Außerdem hängt "m" von der Größe der Texte ab. Der Wert "10" wurde für eine Textbasis mit Texten, die durchschnittlich 389 Worte in einem Intervall von [62,472] Worten, enthalten, ermittelt.

Im Jahr 1999 haben Schultz und Liberman [SCH1999] eine Untersuchung durchgeführt, um zu ermitteln, welche Wortanzahl für eine Anfrage bezüglich Precision und Recall optimal ist.

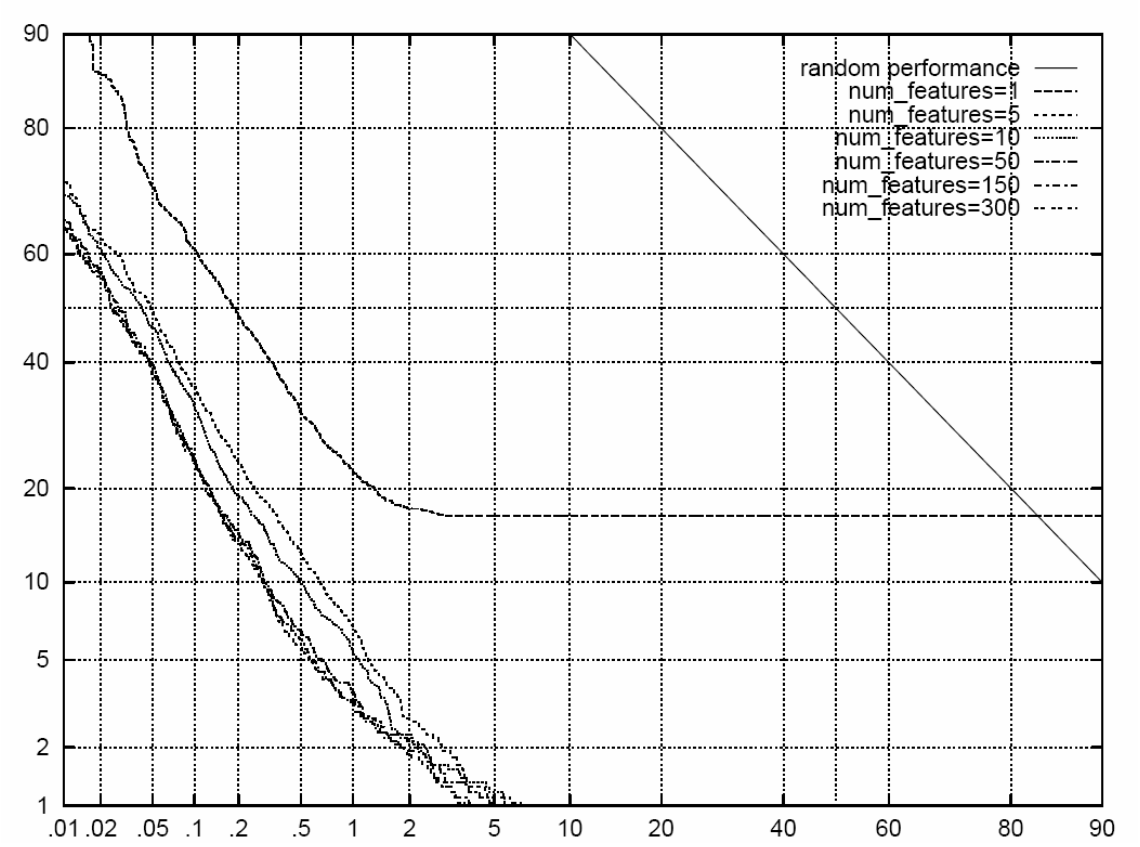


Abbildung 80:Negative Precision (y-Achse) und negative Recall (x-Achse) Quoten in Prozent für verschiedene Wortanzahlen (num_features) nach Schultz und Liberman [SCH1999]

Da neben der Anzahl aber auch die selektierten Worte bedeutend - wenn nicht gar entscheidend - sind und diese in [SCH1999] mit IDF (inverse document frequency) ermittelt wurden, sind die Ergebnisse nicht ohne weiteres für andere Verfahren adaptierbar. Schultz und Lieberman geben 50 Worte als obere Schranke für eine Verbesserung der Precision an. Da die Tests auf größeren Texten und nur mit 179 festen Stoppworten durchgeführt wurden, dürfte die obere Schranke bei der CRIC Testumgebung (kleinere Texte und dynamische Stoppworte) deutlich unter diesem Wert liegen.

Textsuche in CRIC

Das Ergebnis der Volltextsuche auf dem DBMS ist eine Menge von Texten `T` (*Resultset*), die nach der Confidence[12] (Con_T) sortiert sind. Das Resultset besteht also aus Tupeln (T,Con).

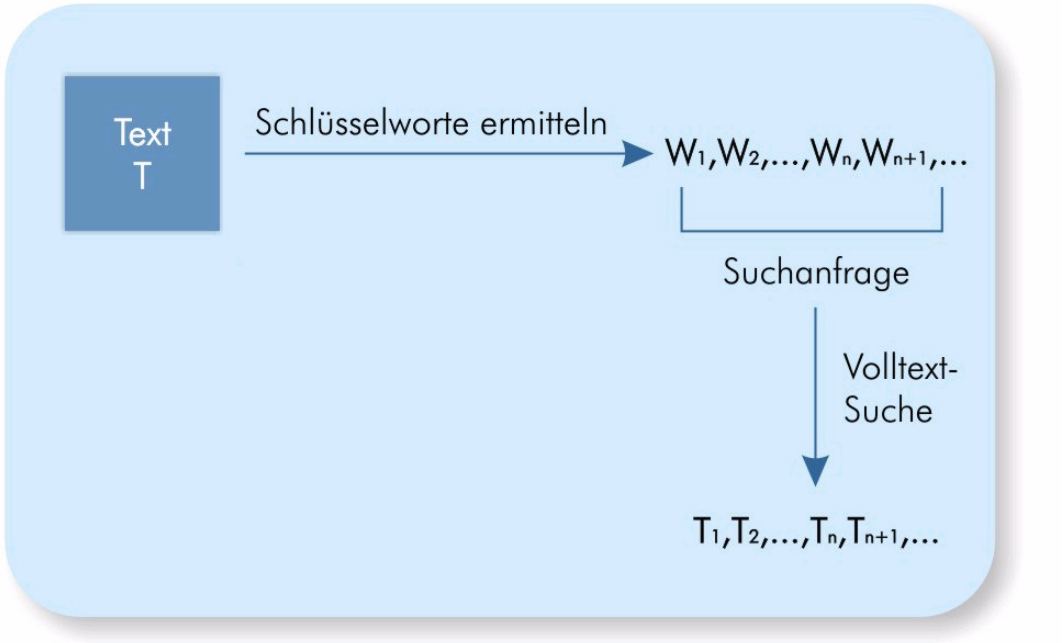


Abbildung 81:Die Schlüsselworte werden für eine Anfrage an eine Volltextsuche eines DBMS verwendet

Die ersten `n` Texte $T_{i=1\ldots n}$ mit einer Confidence $Con(T_i)$ größer dem Schwellwert `ConMin` werden selektiert. Diese Selektion bildet die Basis für die eingangs eingeführte Relation `V`, die zu jedem Text die inhaltlich verwandten Texte vorhält. Formal ergibt sich folgende Funktion `f`:

$$f_n(W_1,\ldots,W_m)=\{(T_1,Con(T_1)),\ldots,(T_n,Con(T_n))\} \text{ mit } Con(T_i)>ConMin$$

6.1.3.1 Funktionsweise der Volltextsuche

Als Datenbanksystem wurde für die Evaluation in der Praxis das Datenbanksystem *MySQL* verwendet. Da sonst nur ANSI-SQL Verwendung findet, muss lediglich der proprietäre Befehl zur Volltextsuche [GOL2003][MYS2005] bei der theoretischen Laufzeitberechnung näher betrachtet werden (siehe 8. Das theoretische Laufzeitverhalten). Die Syntax der Volltextsuche lautet

```
SELECT * FROM table
WHERE MATCH column AGAINST ('text');
```

wobei `column` die Texte `T` der Textbasis `C` enthält.

Wortgewichtung

Mit einem TF-IDF-Derivat wird die Gewichtung w_{DB} eines Wortes `W` der Suchanfrage ermittelt [GOL2003]:

$$w_{DB}(W)=\frac{\log(N_T(W))+1}{\sum_{i=1\ldots n}\log(N_T(W_i))+1}*\frac{u}{1+0{,}115*u}*\frac{\log|C|-I_C(W)}{I_C(W)}$$

[12] Fast alle Datenbanksysteme liefern einen *Confidence*- oder *Relevance*-Wert für das Ergebnis einer Volltextsuche. Die Confidence stellt den Grad der Übereinstimmung zwischen Such-Anfrage und dem Attribut des Tupels der angefragten Datenbank-Relation dar.

mit

- u: Anzahl der unterschiedlichen Worte in Text T
- |C|: Anzahl aller Texte
- $N_T(W)$: Anzahl der Instanzen von Wort W in Text T
- $N_T(W_i)$: Anzahl der Instanzen von Wort W_i in Text T
- $I_C(W)$: Anzahl der Texte mit mindestens einer Instanz von Wort W

Dabei stellt der erste Faktor den TF-Teil und der dritte Faktor den IDF-Teil dar. Durch den zweiten Faktor werden lange Texte abgewertet (und kurze aufgewertet).

Schließlich wird w_{DB} noch mit der Anzahl a des Wortes W in der Suchanfrage multipliziert, um so die Confidence zu ermitteln:

$$Con_T(W) = a * w_{DB}$$

Ein mehrfaches Vorkommen des Suchwortes in der Anfrage hat demnach eine signifikante Bedeutung. Dieses Verfahren wird für alle Suchworte der Anfrage durchgeführt.

Inverser Index

Das Datenbanksystem benötigt für eine effiziente Volltextsuche eine Datenstruktur. Diese wird als inverser Index (reverse Index) bezeichnet und ähnelt der CRIC Datenstruktur für Texte C und Wortliste WL. Allerdings benötigt das Datenbanksystem nicht nur die Anzahl der Instanzen der Worte insgesamt (Wortliste WL) und die Anzahl der Texte in denen ein Wort vorkommt, sondern auch die Anzahl der Instanzen der Worte in den einzelnen Texten (Wort-Text-Relation WT). Letztere bildet einen klassischen inversen Index:

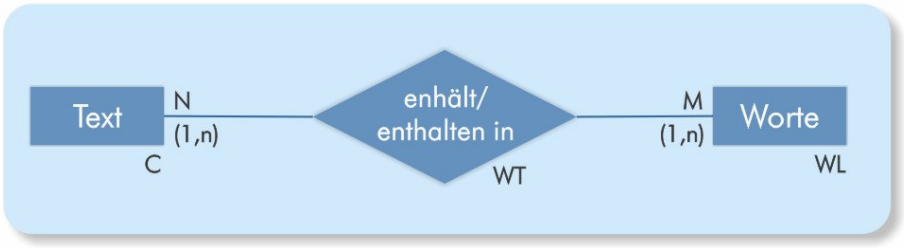


Abbildung 82:Die Wort-Text-Relation WT bildet auf dem Datenbanksystem einen klassischen inversen Index für die Volltextsuche

6.1.4 Linguistische Motivation

Der gewählte Lösungsansatz von CRIC basiert auf der fundamentalen Annahme, dass die Bedeutung eines ausreichend kurzen Textes[13] auf wenige Worte zurückgeführt werden kann [LOE2003],[GLU2000; S.93, ff.],[PUT2004]. Ferner disambiguieren sich diese Worte wechselseitig. Letzteres hat eine breite linguistische und philosophische Basis:

- Firth: „You shall know a word by the company it keeps" [FIR1957]
- Wittgenstein: „Man kann für eine große Klasse von Fällen der Benützung des Wortes ›Bedeutung‹ - wenn auch nicht für alle Fälle seiner Benützung - dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.“ [WIT1968]

[13] Bei längeren Texten (Mehrseiter, Bücher et cetera) ergibt sich die Notwendigkeit diese in Teile zu zerlegen im Wesentlichen aus zweierlei Gründen. Zum Einen wird es immer schwerer einen Text auf wenige Worte zu reduzieren je umfangreicher – und damit thematisch breiter – er wird, zum anderen muss der Benutzer innerhalb eines umfangreichen Textes die relevanten Passagen, aufgrund derer eine Empfehlung ausgesprochen wurde, erneut suchen.

6.1.4.1 Disambiguierung/Monosemierung durch Wortwolken

Zwei bedeutende Probleme mit denen Empfehlungssysteme und Suchverfahren zu kämpfen haben sind die Homonymie und die Synonymie. Beide werden durch die Wortwolken[14] (aus den n relevantesten Worten eines Textes bestehende Suchanfrage) des CRIC Ansatzes abgeschwächt:

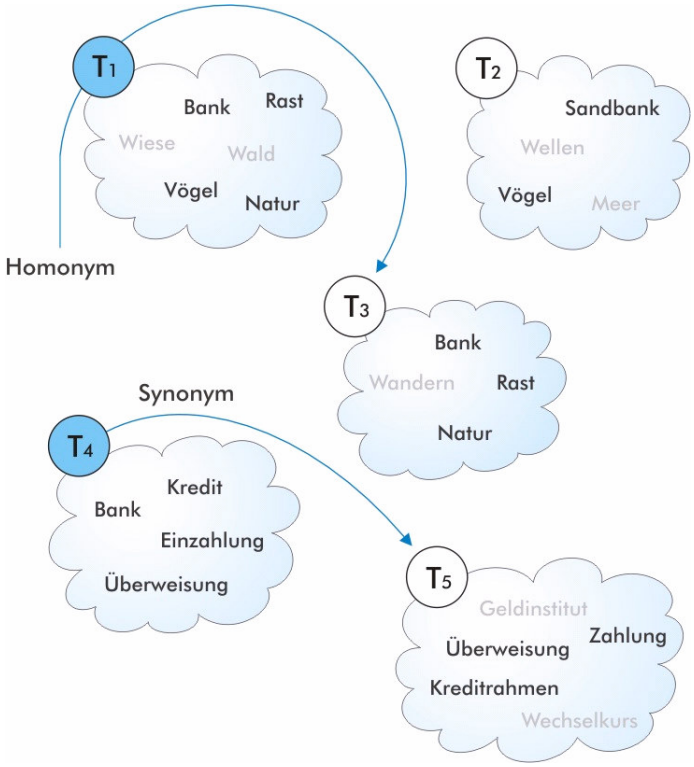


Abbildung 83:Homonyme und Synonyme beeinträchtigen Precision und Recall eines Empfehlungssystem negativ

Homonyme/Polysemie

Ein Homonym ist ein Token (genauer: ein sprachliches Zeichen) das bei identischer Schreib- und Sprechweise eine unterschiedliche Bedeutung hat – also polysem ist. Dass diese Polysemie auch Menschen Probleme bereitet, zeigen verschiedene Untersuchungen. So hat Jorgeson beispielsweise gezeigt, dass Testpersonen in nur 68% aller Fälle bei der Bestimmung der Bedeutung eines Wortes (engl. "WSD" oder "word sense disambiguation") überein stimmen [JOR1990].

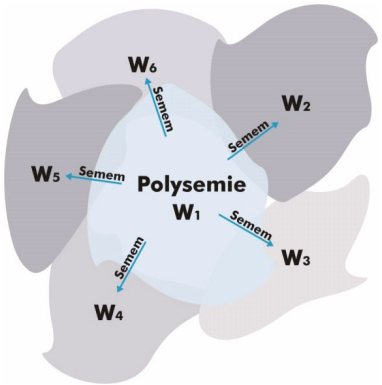


Abbildung 84:Polysemie kann durch Wortwolken aufgelöst werden; einem Lexem (Token) wird implizit ein eindeutiges Semem zugeordnet.

[14] Eine Wortwolke steht für eine Menge von Worten; in der Fachliteratur ist der Begriff "bag of words" anzutreffen.

In Abbildung 83 findet sich mit "Bank" ein prominentes Beispiel. Wird es wie in Text T_1 in einer Wortwolke mit "Rast", "Vögel" und "Natur" kombiniert, verliert es offensichtlich seine Ambiguität/Polysemie, da die Wahrscheinlichkeit einen Text zu finden, der diese Worte enthält und sich mit einer Bank im Sinne eines Geldinstitutes beschäftigt, gegen Null geht.

Diese informelle Schlussfolgerung an einem Beispiel wird durch verschiedene Untersuchungen gestützt. So hat Yarowski [YAR1993] gezeigt, dass bereits ein weiteres Wort im direkten Umfeld des Wortes genügt, um den Wortsinn mit mindestens 90% Wahrscheinlichkeit zu bestimmen (zu disambiguieren). Das gilt allerdings nur für Worte mit zwei verschiedenen Wortbedeutungen und in einem engen Mikro-Kontext von wenigen Worten um das betrachtete Wort.

Allerdings haben Gale, Church und Yarowski [GAL1993] in einer anderen Untersuchung gezeigt, dass man die Qualität der der Disambiguierung durch eine Erweiterung des Mikro-Kontexts um das betrachtete Wort von 6 Worten auf 50 Worte um 4% verbessern kann.

Das betrachtete Textfenster bei CRIC umfasste im Rahmen der qualitativen Evaluation maximal 472 und im Mittel 389 Worte. Ferner werden 10 Worte als Wortwolke für die Anfrage verwendet. Diese 10 Worte disambiguieren sich wechselseitig [IDE1998], da sie alle für die Suchanfrage im DBMS verwendet werden.

Ausgehend von [YAR1993] und unter den Annahmen

- alle Worte außerhalb des 50 Wort-Umfeldes tragen nichts zur Disambiguierung bei (worst case)
- CRIC selektiere Worte gleichmäßig über den kompletten Text (average case)

kann man eine Abschätzung der Disambiguierungswahrscheinlichkeit durch CRIC ableiten. Die Wahrscheinlichkeit, dass ein Wort durch ein weiteres nicht disambiguiert wird, ist nach [YAR1993]

$$P(W_i \text{ nicht disambig.}) = 0{,}1$$

mit W_i als den selektierten Worten. Erweitert man das Textfenster von 50 auf 500 (obere Schranke bei CRIC), so gilt

$$P(W_i \text{ nicht disambig.}) = \frac{450}{500} * 1 + \frac{50}{500} * 0{,}1 = 0{,}91$$

Berücksichtigt man nun, dass nicht nur eines, sondern neun Worte zur Disambiguierung zur Verfügung stehen, so ergibt sich

$$P(W_i \text{ nicht disambig.}) \approx 0{,}91^9 \approx 0{,}43$$

Setzt man das betrachtete Textfenster auf 389 (Mittelwert in CRIC), so ergibt sich sogar

$$P(W_i \text{ nicht disambig.}) \approx 0{,}88^9 \approx 0{,}33$$

Selbst unter der gemachten unrealistischen worst case Abschätzung bezüglich des Disambiguierungspotenzials der Worte außerhalb des 50-Worte-Fensters ist die Wahrscheinlichkeit, dass ein Wort der Wortwolke durch die anderen Worte disambiguiert wird, mit rund 0,67 bereits beachtlich.

Synonyme

Ein Synonym ist ein Token, das die (zumindest teilweise) gleiche Bedeutung wie ein Token mit unterschiedlicher Schreibweise hat. In Abbildung 83 ist das Paar "Bank" und "Geldinstitut" ein solches Beispiel. Auch hier helfen die Wortwolken wiederum. Obwohl in den Texten T_4 und T_5 nur die Synonyme Verwendung finden, ermöglichen die übrigen übereinstimmenden Worte der Wortwolken eine potenzielle wechselseitige Empfehlung.

7 Laufzeitkomplexität

Zur Ermittlung der Zeitkomplexität für einen gegebenen Algorithmus und eine gegebene Datenstruktur unter Berücksichtigung der Eigenschaften der Daten[15] kommen im Wesentlichen die Ansätze der *empirischen Untersuchung* und der *theoretischen Analyse* in Frage.

Da das im Rahmen dieser Arbeit vorgestellte Verfahren auf einem DBMS aufbaut, erscheint eine empirische Untersuchung auf den ersten Blick einfacher. Sie birgt aber eine Reihe von Nachteilen, die selbst mit großem Aufwand nicht alle durchgehend beseitigt werden können. Der Autor hat im Rahmen von [KLA1993] und [KLA1995] selbst diese Erfahrung machen müssen. Die wichtigsten dieser Nachteile seien hier genannt:

- **Abhängigkeit von einer konkreten DBMS-Implementation**

  Beispielsweise werden SQL-Anfragen von den Optimizern verschiedener DBMS unterschiedlich "optimiert"[16], woraus stark abweichende Laufzeiten resultieren können. Dieses Manko könnte nur durch Messungen mit zahlreichen unterschiedlichen DBMS behoben werden, was aber den Aufwand der Datenermittlung signifikant erhöhen würde.

- **Abhängigkeit von der eingesetzten Hardware**

  Auf den ersten Blick mag es unerheblich erscheinen, welche Hardware zum Einsatz kommt, da man auf Basis von Benchmark-Werten vermeintlich auf andere Hardware "umrechnen" kann. Dass die Hardware dennoch eine Rolle spielt, liegt darin begründet, dass diese aus vielen Einzelkomponenten besteht, die das Laufzeitverhalten unterschiedlich beeinflussen können.

- **Seiteneffekte aufgrund vorangegangener Tests**

  Alle modernen DBMS bieten ausgefeilte Cache-Mechanismen, um die Zugriffszeiten auf häufig benötigte Daten zu reduzieren. Um bei der Ermittlung empirischer Daten Seiteneffekte durch im Cache befindliche Informationen zu vermeiden, muss dieser deaktiviert oder – da eine zuverlässige Deaktivierung nicht immer möglich - vor jeder Anfrage geleert werden. Da der Cache bei vielen DBMS aber nicht auf Anfrage geleert werden kann, bleibt für zuverlässige Testergebnisse nur der komplette Neustart des Systems.

Mit der theoretischen Analyse lassen sich die oben angeführten Probleme vermeiden. Wenn sich aus einem gegebenen Verfahren, dessen SQL-Anfragen, der Datenstruktur und dem Mengengerüst eine theoretische Zeitkomplexität ableiten lässt, ist diese einem empirischen Vergleich überlegen.

7.1 Begriffsdefinitionen

Im Folgenden werden die wichtigsten Begriffe dieses Abschnittes definiert.

Laufzeitkomplexität

Unter der *Laufzeitkomplexität* oder auch *Zeitkomplexität* eines Algorithmus versteht man dessen theoretisch erforderlichen Rechenaufwand. Dabei ist zwischen der Landau- (Paul Landau, 1877-1938) oder O-Notation und der Ω-Notation zu unterscheiden. Erstere gibt den ungünstigsten Rechenaufwand (obere Schranke der Laufzeit), letztere den günstigsten Rechenaufwand (unter Schranke der Laufzeit) eines Algorithmus wieder.

Seien $f, g: N \rightarrow R$ gegeben, dann gilt

$$g(n) \in O(f(n)) \Leftrightarrow \exists n_0 > 0 \wedge \exists\, c > 0, \text{ so dass } \forall\, n >= n_0 : g(n) <= c * f(n)\}$$

[15] Eine konkrete Ausprägung der Daten darf nicht angenommen werden, da sonst keine allgemeingültigen Aussagen getroffen werden können. Sehrwohl können aber allgemeine Eigenschaften der Daten wie beispielsweise Anzahl von Datensätzen, Werteintervalle für Attribute et cetera verwendet werden.

[16] Eine Optimierung basiert auf den zugrunde liegenden Algorithmen und den verfügbaren Eigenschaften der Ausgangssituation (Kardinalitäten, Valenzen, immanente Eigenschaften der Datenstruktur, semantische Informationen et cetera) beziehungsweise auf der Abschätzung des Einflusses der Operatoren auf die Eigenschaften der Ausgangssituation. In diesen grundlegenden Punkten unterscheiden sich die Optimizer unterschiedlicher DBMS oft erheblich. Daher gleichen sich die Ausführungspläne zweier DBMS bei gleichem Datenbestand keineswegs immer.

Dabei steht n für die Anzahl der Eingangsdaten, die vom Algorithmus zu verarbeiten sind. Beispiele sind die Elemente eines zu sortierenden Arrays oder Worte eines zu parsenden Textes.

Die Funktionen der Menge O(f(n)) sind alle ab einem bestimmten n_0 (bestimmt Menge von Eingangsdaten) durch c*f(n) nach oben beschränkt. Statt von der Zugehörigkeit zur Menge O(f(n)) wird im Folgenden auch von der Laufzeitkomplexität O(f(n)) oder von der Zugehörigkeit zur Klasse O(f) gesprochen.

Die Laufzeitkomplexitäten O(f(n)) lassen sich bezüglich der Funktion f(n) in folgenden Klassen einteilen:

- Logarithmisch: O(log(n))
- linear: O(n)
- quadratisch (polynomial): $O(n^2)$
- kubisch (polynomial): $O(n^3)$
- polynomial: $O(n^c)$ mit $c \in N$
- exponentiell: $O(c^n)$ mit $c \in N$

Datenbankmanagementsystem (DBMS)

Im Jahr 1970 veröffentlichte Edgar F. Codd seine Ideen über das Relationale Datenbankmodell in Form des Beitrages „A Relational Model of Data for Large Shared Data Banks“ [COD1970].

Es dauerte rund 10 Jahre bis diese Theorie Einzug in die Praxis hielt. Mittlerweile sind Relationale Datenbanksysteme (RDBMS) der De-facto-Standard in Sachen Datenhaltung und -verarbeitung. Wenn in dieser Arbeit von DBMS, Datenbankmanagementsystem oder Datenbanksystem die Rede ist, ist immer RDBMS gemeint.

SQL

Die *Structured Query Language*, kurz *SQL*, wurde 1972 von der IBM Research Group in Yorktown Heights (N.Y., USA) unter dem Namen SEQUEL (Structured English Query Language) entwickelt.

Fünf Jahre später folgte SEQUEL2. Ab 1982 wurde die Anfragesprache für DBMS immer stärker standardisiert. Die Abkürzung wurde zu SQL modifiziert und das amerikanische ANSI, in dem auch IBM engagiert war, verabschiedete 1986 mit SQL-86 (ANSI X3.125-1986) den ersten echten Standard.

Mit SQL-89 (ANSI X3.135-1989, auch SQL-1 genannt) und SQL-92 (ANSI X3.135-1992, auch SQL-2 genannt) folgten Erweiterungen des Standards. Von der ISO wurde die ANSI-Spezifikation für SQL-92 übernommen. Beide Organisationen haben den aktuellen Standard SQL-99 (ISO/IEC 9075:1999, auch SQL-3 genannt) entwickelt.

Relation und Tupel im mathematischen Sinn

Seien $D_1, \ldots, D_g$ beliebige Mengen. Das kartesische Produkt $D_1 \times D_2 \times \ldots \times D_g$ ist dann wie folgt definiert:

$$D_1 \times D_2 \times \ldots \times D_g = \left\{ \left(d_1, d_2, \ldots, d_g \right) \middle| d_j \in D_j, j = 1,2,\ldots,g \right\}$$

mit $(d_1, d_2, \ldots, d_g)$ ist ein g-Tupel und d_j ist j-te Komponente.

Mathematisch ist eine Relation R als Teilmenge des kartesischen Produktes definiert:

$$R \subseteq D_1 \times D_2 \times \ldots \times D_g$$

Man bezeichnet R als g-stellige Relation über den Mengen $D_1, D_2, \ldots, D_g$. Dabei wird g auch der Grad der Relation genannt. Per definitionem kann eine Relation R auch als eine Menge von g-Tupeln betrachtet werden. Aus

$$t = \left(d_1, d_2, \ldots, d_g \right) \in R$$

folgt

$$t \in D_1 \times D_2 \times \ldots \times D_g$$

Relationenschema und Relation

In der Datenbank Theorie wird durch

$$RS(A_1, A_2, \ldots, A_g) = dom(A_1) \times dom(A_2) \times \ldots \times dom(A_g)$$

ein g-stelliges Relationenschema RS definiert, das aus den Attributen A_1 bis A_g besteht. Jedes Attribut repräsentiert eine Domäne $dom(A_i)$. Die Domänen stellen Wertemengen dar. Zu diesem Relationenschema RS gehören alle Relationen R mit

$$R \subseteq dom(A_1) \times dom(A_2) \times \ldots \times dom(A_g)$$

In einem DBMS existiert immer genau eine Relation R zum Relationenschema RS. Dem überwiegenden Teil der Fachliteratur folgend, soll im Folgenden die sprachliche Unterscheidung zwischen dem Relationenschema RS und der konkreten Relation R aufgehoben werden. Wir sprechen fortan nur noch von der Relation R. Ob damit das Relationenschema oder die Relation bezeichnet wird, ergibt sich aus dem Kontext.

Ergebnisrelation

Unter der Ergebnisrelation E einer Relation R soll – soweit nicht anders angegeben – immer die Relation Op(R) verstanden werden.

Tupel und Attribute

Ein Element $t=(d_1, d_2, \ldots, d_g) \in R$ ist ein g-Tupel der Relation R. Es ist $d_j=t[j]$ die j-te Komponente des Tupels. Die Anzahl der Tupel ist definiert durch $|R|$.

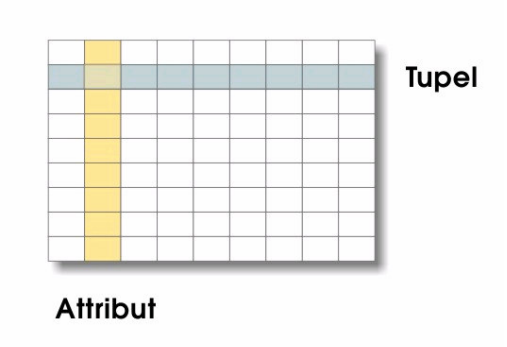


Abbildung 85: Darstellung einer Relation als Tabelle mit Spalten (Attribute) und Zeilen (Tupel)

In dieser Arbeit sind - soweit nicht explizit anders angegeben - mit $(A_1, \ldots, A_g)$ stets die Attribute der Relation R und mit $(B_1, \ldots, B_h)$ stets die Attribute der Relation S gemeint. Jedes Attribut repräsentiert wie oben angegeben eine Domäne $dom(A_i)$.

Relationen in der Praxis

In der Praxis erfüllen die Relationen (Tabellen) eines DBMS nicht die theoretische Mengeneigenschaft der Relationen. So kommen identische Tupel in einer Relation vor. Des Weiteren gibt es einen Operator, der eine Ordnung auf einer Relation erzeugt.

Um diesen Gegebenheiten gerecht zu werden, erweitern wir die Tupel jeder Relation R um ein "künstliches" Attribut "A_{ID}" (tuple identifier; kann anschaulich als "Zeilenzähler" auf den Tupeln betrachtet werden), das für jedes Tupel in R eindeutig ist und gleichzeitig eine Ordnung auf R definiert. Das Attribut A_{ID} sei für Operatoren als Parameter nicht ansprechbar und werde bei der Ausgabe in eine Ergebnisrelation eliminiert beziehungsweise in dieser wieder neu gesetzt.

Eine Relation (Tabelle) ist in einem DBMS zunächst leer. Nur durch den INSERT Befehl (oder implizit beim Aufbau einer Ergebnisrelation) können neue Tupel in die Tabelle gelangen. Für jedes neue Tupel $t=(d_1, d_2, \ldots d_g)$, das in R eingefügt wird, gelte bezüglich des künstlichen Attributes A_{ID}:

$$t[A_{ID}] = \max(t_i[A_{ID}]) + 1$$

mit $t[A_j]$ als j-ter Komponente von t und $i=1\ldots|R|$.

Es gelte ferner $A_{ID}=A_0$. Wenn der tuple identifier für eine Operation Bedeutung hat, stimmt die Indizierung der Attribute also nicht mit der Indizierung der Komponenten überein.

Kardinalität

Die Kardinalität $|R|$ einer Relation R ist gleich der Anzahl der Tupel von R. Wenn nicht anders angegeben, steht *n* für $|R|$ (die Anzahl der Tupel von R) und *m* für $|S|$ (die Anzahl der Tupel von S)

Valenz

Die Valenz $val_{A[j]}(R)$ eines Attributes A_j in einer Relation R stellt die Anzahl unterschiedlicher Werte des Attributes A_j in R dar:

$$val_{A[j]}(R)=|\{t_i[A_j]|i=1...|R|\}|$$

Primärschlüsselattribut

Die Werte eines Primärschlüsselattributes sind paarweise verschieden. Daraus folgt, dass die Valenz eines Primärschlüsselattributes mit der Kardinalität der Relation des Attributes identisch ist.

Selektivität

Die Selektivität sel_{op} eines SQL Operators Op in Bezug auf eine Relation R ist wie folgt definiert:

$$sel_{op} = \frac{|Op(R)|}{|R|},$$

Es ist die Kardinalität der aus der Operation Op resultierend Ergebnisrelation E im Verhältnis zur Kardinalität der Relation R. Man beachte, dass ein kleinerer Wert der Selektivität als "höhere Selektivität" bezeichnet wird. Bei der Analyse der Laufzeitkomplexität sequentiell ausgeführter relationaler Operatoren spielt die Kardinalität der Ergebnisrelation ($E=|Op(R)|$) eines Operators auf der Relation R eine wichtige Rolle.

Da $|Op(R)|$ in den wenigsten Fällen genau berechnet, sondern nur abgeschätzt werden kann, kann folglich auch sel_{op} nur abgeschätzt werden.

Basisdomänen

Im Kontext der DBMS können anhand des *Datentyps* mehrere Basisdomänen unterschieden werden, von denen die wichtigsten hier aufgeführt werden sollen:

Basisdomäne	Definition
INTEGER	ganze Zahlen
DECIMAL	Festkommazahlen
FLOAT	Fließkommazahlen
STRING[L]	Zeichenfolgen mit max. Länge L
TEXT	Volltext
BOOLEAN	Logische Werte (true, false, null)
DATE	Kalenderdaten
TIME	Uhrzeit

Tabelle 1: Übersicht der Basisdomänen

Im Folgenden soll jede *Domäne* als Teilmenge dieser Basisdomänen verstanden werden. Auf der TEXT Domäne werden – bis auf die im Rahmen des vorgestellten Verfahrens gesondert beschriebenen Volltextoperationen – keine Relationalen Operatoren angewendet.

Besonderheiten der STRING-Domänen

Die STRING-Domäne besteht aus Zeichenfolgen mit einer maximalen Länge L. Ein Zeichen ist ein Element des Alphabetes "ASCII-Code[17]". Ein Alphabet $(\Sigma, <)$ ist eine endliche, nichtleere Menge Σ zusammen mit einer totalen Ordnung $<$ auf Σ. Ein Wort über dem Alphabet $(\Sigma, <)$ ist eine Folge über Σ, also eine endliche Folge von Zeichen aus Σ. Seien im Folgenden u, v, u_1, v_2 Worte über Σ und x, y Zeichen aus Σ. Die lexikographische Ordnung auf Worten des Alphabetes $X=\{x_1, x_2, \ldots, x_n\}$ ist dann wie folgt rekursiv definiert.

Seien zwei Worte $u=xu_1$ und $v=yv_1$ gegeben, dann gilt

$x<y \Rightarrow u<v$

$x=y \Rightarrow u<v \Leftrightarrow u_1<v_1$

Eine Nummerierung N(x), die eine lexikographische Metrik δ auf den Worten eines Alphabetes $(\Sigma, <)$ liefert, ist wie folgt definiert:

$$N(x)=\Sigma_{j=0\ldots|x|-1}\ 256^{L-j}*Z[x[j+1]]$$

$$\delta(x,y)=|N(x)-N(y)|$$

mit $Z[x]$ als sequenzieller Nummerierung der Zeichen des Alphabetes Σ anhand der Ordnung $<$.

Ein Beispiel für T-Werte verschiedener Worte mit L=9 wird im Folgenden angegeben:

Auf der STRING-Domäne wird eine solche lexikographische Ordnung beispielsweise beim Sortieren angewendet. Durch eine Nummerierung wie oben beschrieben sind auch Intervalle und Operationen, wie sie beispielsweise in 7.5.1.3 verwendet werden, möglich. Es ist aber zu beachten, dass die Nummer N(x) bei STRING-Domänen mit großer Zeichenzahl schnell wächst. Dazu ein Beispiel für Worte bis zu neun Zeichen:

Wort x	1.	2.	3.	4.	5.	6.	7.	8.	9.	Nummer N(x)
Test	84	101	115	116						398550224986907000000000
Thesaurus	84	104	101	115	97	117	114	117	115	398604556238493000000000
Testing	84	101	115	116	105	110	103			398550225102829000000000
Pyro	80	121	114	111						380029620471933000000000
Tests	84	101	115	116	115					398550225113350000000000

Tabelle 2: Beispiele für T-Werte, die als Basis einer Metrik auf einer STRING-Domäne dienen

7.2 Das Hauptproblem der theoretischen Analyse

Bei der theoretischen Analyse müssen SQL-Anfragen aufgrund einer gegebenen Datenstruktur samt Eigenschaften der Daten bezüglich der Laufzeitkomplexität beurteilt werden. Das Hauptproblem besteht daher darin, ein Transformationsverfahren zu entwickeln, das zu einer gegebenen SQL-Anfrage eine korrespondierende Zeitkomplexität ausgibt.

Abbildung 86: Das erforderliche Transformationsverfahren leitet aus einer gegebenen SQL-Anfrage die zugehörige Zeitkomplexität ab.

[17] Im Rahmen der Arbeit wurde ASCII-Code verwendet. Eine Unicode-Adaption wäre problemlos möglich.

7.3 Zeitkomplexität

Für die Zeitkomplexität einer SQL-Anfrage ist offensichtlich die Anfrageausführung im DBMS entscheidend. Letztere gilt es daher zu analysieren und auf eine Möglichkeit hin zu prüfen, die Laufzeitkomplexität theoretisch und in Abhängigkeit von den Parametern der zugrunde liegenden Datenstruktur (deren Eigenschaften und Mengengerüst) zu bestimmen.

Zwar beschäftigen sich zahlreiche Publikationen mit der Optimierung von SQL-Anfragen in DBMS, aber keine liefert ein umfassendes Verfahren zur Ableitung der Zeitkomplexität für eine komplette Anfrage[18]. Selbst zur Zeitkomplexität einzelner Operatoren lässt sich die benötigte Information in der Literatur nur bruchstückhaft und oft nicht mit dem nötigen Detailgrad finden.

Daher soll im Folgenden ein solches Verfahren zur Ermittlung der Laufzeitkomplexität einer gegebenen SQL-Anfrage entwickelt werden. Im Wesentlichen aufbauend auf [MIT1995] und [DAD1996] und mathematischen Quellen ([BRO1987] et cetera) wird ein Regelwerk zur Selektivitäts-[19] und Laufzeitabschätzung Relationaler Operatoren entwickelt (siehe Abschnitt 7.5).[20] Wir beschränken uns auf die obere Schranke, also auf die O-Klassen der Zeitkomplexität.

7.3.1 Bedeutung der Selektivität und Zeitkomplexität Relationaler Operatoren

Die Anfragenverarbeitung eines DBMS (siehe dazu Abschnitt 7.6) lässt sich in zwei wesentliche Phasen unterteilen. Die erste, die *Vorbereitungsphase* genannt werden soll, erstreckt sich von der Anwenderanfrage über die Transformation der SQL-Anfrage in Relationale Operatoren bis hin zur Ermittlung des optimalen Operatorbaumes. Die zweite Phase soll *Ausführungsphase* genannt werden. Sie bringt die zuvor ermittelte und aus DBMS Sicht optimale Operatorreihenfolge zur Ausführung und übergibt das Ergebnis der Anfrage an den Anwender.

7.3.2 Ermittlung der Zeitkomplexität und Selektivität

Da ein DBMS die aus der Vorbereitungsphase der Anfrageverarbeitung resultierenden Relationalen Operatoren sequentiell abarbeitet, genügt es, die Laufzeitkomplexität einzelner Relationaler Operatoren auf Basis der Eingabedaten[21] zu ermitteln.

Um eben diese Eingabedaten für den folgenden Operator zu erhalten, muss aber auch noch eine so genannte Selektivitätsabschätzung betrieben werden. Sie besagt, wie ein Relationaler Operator die Kardinalität einer Relation, auf die er angewendet wird, verändert.

Zur Abschätzung von Selektivität und Zeitkomplexität kann man sich die für DBMS-Optimizier entwickelten Abschätzungsverfahren zunutze machen. So bietet die Optimizer-Literatur bezüglich der Laufzeitkomplexität und Selektivität Relationaler Operatoren eine solide Grundlage. Der Grund für das geringere Literaturangebot bei der Selektivität dürfte in der Diskrepanz zwischen den Möglichkeiten eines Optimizers und eines menschlichen "Evaluators" liegen.

---

[18] Optimizer arbeiten aus Effizienzgründen nicht vollenumerativ auf den potenziellen Varianten der Operatorbäume. Um die optimale Ausführungsreihenfolge zu ermitteln, werden daher nicht alle möglichen Permutationen aller Operatoren vollständig berechnet, sondern Regeln zur Vorauswahl bestimmter Operatoren angewendet (zum Beispiel die Push-Regel für Selektionen, um diese aufgrund ihrer meist hohen Selektivität und relativ geringen Laufzeit möglichst früh auszuführen). Die Literatur zur statistischen Optimierung beschäftigt sich daher vorrangig mit der relativen Einordnung der Zeitkomplexität einzelner Relationaler Operatoren. Der optimale Ausführungsplan wird quasi schrittweise erzeugt.

[19] Statt der Selektivität ermitteln wir eigentlich die Kardinalität der Ergebnisrelation eines Operators, da diese die Laufzeit der nachfolgenden Operatoren maßgeblich beeinflusst (siehe dazu auch 7.1).

[20] Dieses Regelwerk kann potenziell der Evaluation beliebiger SQL-Anfragen dienen.

[21] Die Eingabedaten des Operators O_2 sind die Ausgabedaten des Operators O_1.

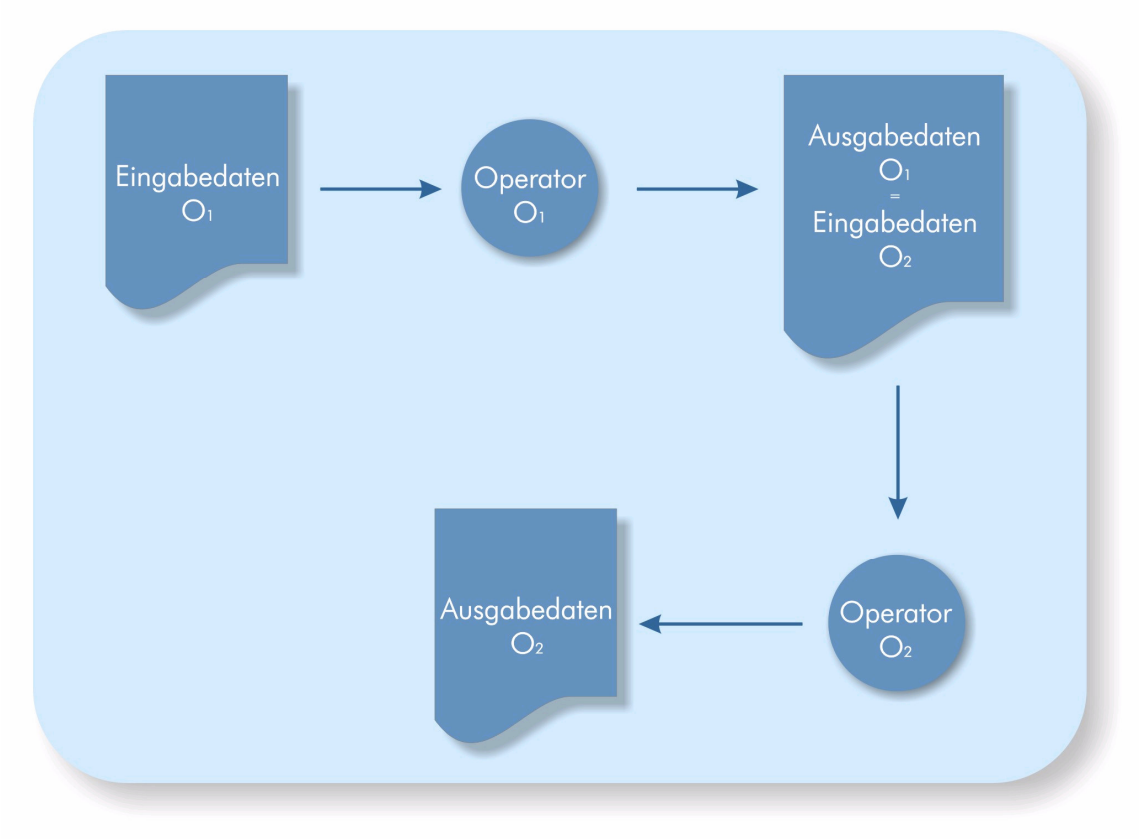


Abbildung 87: Die Ausgabedaten vorangegangener Operatoren sind von großer Bedeutung

7.3.3 Abhängigkeit der Selektivität von Datenausprägungen

Behalten die Laufzeitkomplexitäten der Relationalen Operatoren bei unterschiedlicher Datenausprägung ihre wesentlichen Eigenschaften, so gilt dies nicht für die Selektivität. Sie ist in hohem Maße von der Ausprägung der Daten abhängig.

Um dem Rechnung zu tragen wurde bei der Entwicklung besserer Optimierungstechniken zunächst versucht, die Selektivität auf Basis statistischer Informationen (im Wesentlichen handelt es sich dabei um die Kardinalitäten der Relationen und die Valenzen ihrer Attribute) abzuschätzen. Da sich aber sowohl die Kardinalität der Relationen, als auch die Valenzen ihrer Attribute im Laufe der Ausführung der Relationalen Operatoren drastisch verändern können, birgt dieser Ansatz Risiken der Fehleinschätzung.

Insbesondere die statistischen Selektivitätsabschätzungen bei Verbünden und anderen komplexen Operatoren sind sehr problematisch. Kommt dann noch eine Ungleichverteilung der zugrunde liegenden Daten hinzu, wird die statistische Selektivität auf Basis ungenauer Annahmen ermittelt, die sich im Laufe der Optimierungsanalyse noch verstärken. Einen sehr guten Überblick über diese Fehlerverstärkung bietet [IOA1995].

7.3.4 Andere Techniken zur Selektivitätsabschätzung

Aufgrund der Nachteile der statistischen Selektivitätsabschätzung ist in der Literatur ein Trend zu so genannten Sampling-Ansätzen und ähnlichen Verfahren zu verzeichnen ([GAN1996], [LIP1990b], [LIP1990], [CHE1994], [IOA1995], [SUN1993] et cetera).

7.3.5 Statistische Selektivitätsabschätzung auf Basis von Metainformationen

Die Situation für einen menschlichen "Evaluator" ist allerdings wesentlich günstiger. Im Rahmen Evaluation ist nicht nur die Datenstruktur, sondern auch die Datenbasis und die Semantik der Relationalen Operatoren bekannt. Aufgrund dieser Metainformation – die ein Optimizer nicht besitzt – kann eine exakte Selektivitätsabschätzung vorgenommen werden.

Diese "erweiterte statistische Selektivität" ist also bestens als Basis der theoretischen Evaluation geeignet, liefert sie doch die zur korrekten Berechnung der Zeitkomplexität der Relationalen Operatoren erforderlichen exakten Kardinalitäten und Valenzwerte.

7.3.6 Der Unterschied der Selektivitätsberechnung in den beiden Phasen der Anfrageverarbeitung

Um zu ermitteln, in welcher Reihenfolge das DBMS die Relationalen Operatoren einer Anfrage zur Ausführung bringen wird, muss man sich quasi in die Lage des Optimizers versetzen. In diesem Fall darf die Selektivität also nicht auf Basis der Metainformationen, sondern nur auf Basis der statistischen Daten (Kardinalitäten und Valenzen) ermittelt werden.

Im daraus folgenden "optimalen" Operatorbaum muss dann die Zeitkomplexität der einzelnen Operatoren berechnet und am Ende summiert werden. Bei dieser Berechnung der Zeitkomplexitäten der Operatorsequenz dürfen und müssen (zur Ermittlung korrekter Werte) dann allerdings die Metainformationen verwendet werden, da diese die realen Kardinalitäten und Valenzen bei Ausführung der Relationalen Operatoren liefern.[22]

7.4 Repräsentative Basisoperationen

Wie bereits erwähnt, konzentriert sich diese Arbeit bei den Untersuchungen auf die konkreten Datenstrukturen und SQL-Anfragen des vorgestellten Verfahrens. Die im Folgenden aufgeführten Operatoren wurden unter Berücksichtigung dieses Sachverhaltes ausgewählt.

7.4.1 Änderungsanfragen

Unter den *Änderungsanfragen* sollen die drei klassischen Anweisungen `INSERT`, `UPDATE` und `DELETE` zusammengefasst werden.

7.4.2 Leseanfragen

Leseanfragen bilden die Gruppe der SQL-Befehle, die keine Änderungen an den Relationen vornehmen. Hier werden auch jene berücksichtigt, die in früheren Varianten des vorgestellten Verfahrens zum Einsatz kamen.

7.4.2.1 Umfang der betrachteten Operatoren

Der Umfang der betrachteten Operatoren geht über das benötigte Maß im Rahmen des zu analysierenden Verfahrens hinaus. Dies ist aber nur in der in dieser Arbeit vorgestellten finalen Version der Fall. Frühere Versionen benötigten alle im Folgenden vorgestellten Operatoren. Erst aufgrund der daraus theoretisch resultierenden Laufzeit wurde das Verfahren optimiert. Daher hat sich der Autor entschlossen, die entwickelten theoretischen Laufzeiten in der ursprünglich erforderlichen Breite darzustellen.

[22] Sich eben diese Informationen – also die aktuellen Kardinalitäten und Valenzen – zunutze zu machen ist der Grundgedanke der Sampling-Techniken (siehe oben). Da die komplette Ermittlung der Kardinalitäten und Valenzen der betroffenen Relationen aber zu teurer wäre, werden beispielhaft Segmente der Relation untersucht und diese Werte auf die Gesamtrelation hochgerechnet.

7.5 Relationale Operatoren

Die Analyse soll in möglichst einheitlicher Form dargestellt werden. Aus diesem Grunde wird im Folgenden die Relationenalgebra als dafür geeignetes Werkzeug eingeführt.

7.5.1 Operatoren

Es werden nur die im Weiteren verwendeten Operatoren der Relationenalgebra berücksichtigt. Es handelt sich dabei grundsätzlich um ein- und zweistellige Operatoren, die eine (R) beziehungsweise zwei (R, S) Ausgangsrelationen in eine Ergebnisrelation E transformieren.

7.5.1.1 Konvention für die Attributbezeichnung

Im Folgenden wird zur Übersichtlichkeit bei der Angabe von Attributen A[i] oder A_i gleichbedeutend verwendet, um eine Doppelindizierung übersichtlicher darzustellen:

$$OP_{A_i} = OP_{A[i]}$$

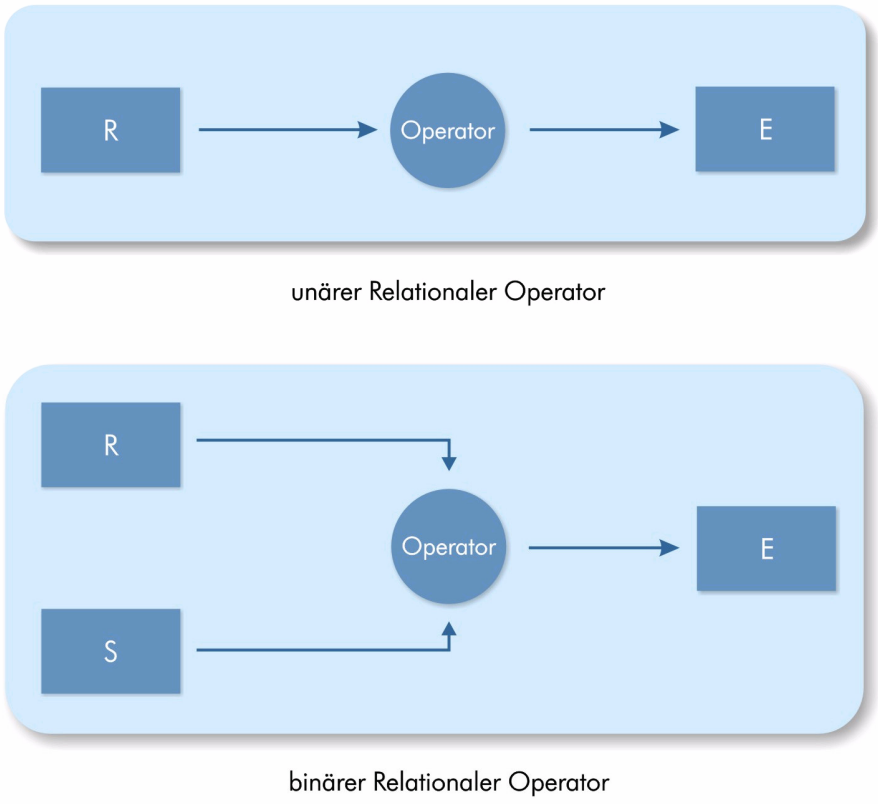


Abbildung 88: Unäre und binäre Relationale Operatoren liefern grundsätzlich eine Ergebnisrelation E

7.5.1.2 Projektion (SELECT) [unär]

Sei $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g mit $t=(d_1,d_2,\ldots,d_g)\in R$. Bezeichne nun analog zu d_j auch d[j] die j-te Komponente von t. Sei ferner $L=(i_1,\ldots,i_s)$ mit $i_k\in\{1,\ldots,g\}$ und $k=1,\ldots,s$ eine Liste von Attribut- beziehungsweise Komponentennummern. Sei ferner t[L] definiert als

$$t[L]=(d[i_1],\ldots,d[i_s])$$

Dann ist die Projektion PJ wie folgt definiert:

$$PJ_L R:=\{t[L]\mid t\in R\}$$

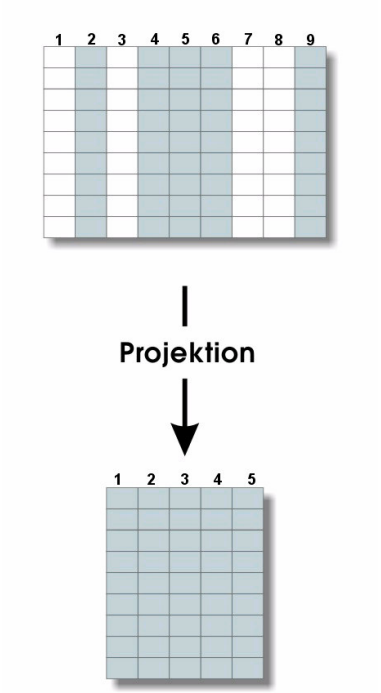


Abbildung 89: Relationaler Operator – Projektion; im Beispiel gilt L=2,4,5,6,9. Die dadurch adressierten Attribute werden in der Ergebnisrelation E zu den Attributen 1,2,3,4,5

Semantik

Die in der Projektion ausgewählten Attribute L bilden eine Teilmenge der Attribute von R. Nur diese werden in die Ergebnisrelation E aufgenommen. Das Attribut A_{ID} der Relation R ist immer auch Bestandteil der Ergebnisrelation E.

Kardinalität der Ergebnisrelation

Sei E die Ergebnisrelation $E:=PJ_LR$, dann gilt:

$|E|=|R|$

Die Projektion verändert die Anzahl der Attribute, nicht aber die Anzahl der Tupel der Ausgangsrelation R. Darin besteht ein Unterschied zur theoretischen Relationenalgebra, die identische Tupel auf der Relation ausschließt, womit die Projektion auch die Kardinalität von R verändern könnte. Wie bereits erwähnt, trifft dies auf Relationen (Tabellen) in konkreten DBMS aber nicht zu.

Zeitkomplexität

Die Projektion gehört zur Klasse

$O(n)$ für $n=|R|$

Die Projektion ist bezüglich der Zeitkomplexität als lineare, von der Tupelzahl abhängige Operation anzusehen. Es gilt daher $O(n)$ mit $|R|=n$.

7.5.1.3 Selektion (WHERE) [unär]

Sei $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g mit $t=(d_1,d_2,\ldots,d_g)\in R$. Dann ist die Selektion (auch als *Restriktion* bezeichnet) SL wie folgt definiert:

$SL_PR:=\{t|t\in R \wedge P(t)=TRUE\}$

mit P als Prädikat auf dem Relationenschema RS darstellt

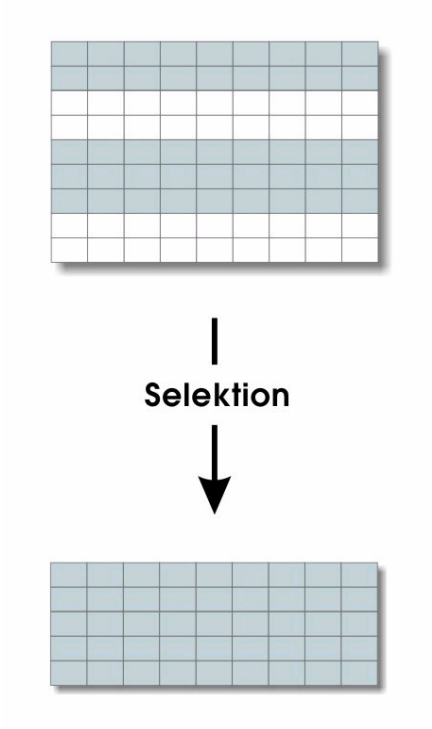


Abbildung 90: Relationaler Operator - Selektion

Semantik

Das in der Selektion angegebene Prädikat `P` bestimmt, welche Tupel der Relation `R` in der Ergebnisrelation aufgenommen werden. Hier einige Beispiele für Prädikate:

WHERE-Bedingung
`a=10 AND b>100`
`a+b<c OR a+b=0`
`a="test" AND NOT(LENGTH(b)<10)`

Tabelle 3: Beispiele für Prädikate

Im Folgenden wird zwischen primitiven und komplexen Prädikaten unterschieden. Erstere verwenden nur Vergleichsoperatoren ("=","<=",">=",">","<","<>"), letztere Operatoren wie "BETWEEN", "IN", "LIKE", et cetera.

Kardinalität der Ergebnisrelation

Sei E die Ergebnisrelation E:=SL_PR, dann gilt:

$$|E| = \prod_{i=1\ldots p} \xi_i * |R|$$

mit einer SQL-Anfrage der Form

```
SELECT ...
FROM ...
WHERE ... P1 AND P2 ... AND Pp
```

wobei "P_i" für ein Prädikat wie im Folgenden aufgeführt steht und ξ_i den im Folgenden definierten Einfluss von P_i auf die Kardinalität der Ergebnisrelation `E` bestimmt.

Eigentlich gilt

$$|E| \approx \prod_{i=1 \ldots p} \xi_i * |R| .$$

da hier der *average case* der Selektivität (angenommene Gleichverteilung der Werte eines Attributes) abgeschätzt wird. Im worst case (alle Werte eines Attributes außer einem kommen genau einmal vor) würde

$$|E| \leq |R| - (val_A(R) - 1)$$

gelten, was bei großen Relationen mit weitgehend ausgeglichener Werteverteilung auf den Attributen sehr nahe an `|R|` liegt und daher in der Praxis wenig brauchbar ist.

Da aber der nachfolgende Operator auf `E` arbeitet und zur weiteren Abschätzung ein konkretes `E` bzw. `|E|` benötigt wird, wird hier "=" im Sinne des *average case* verwendet. Bei den nachfolgend eingeführten Operatoren geben wir es ebenfalls an, wenn das Gleichheitszeichen nur symbolisch gemeint ist.

Prädikat "A = α"

Sei `A` ein Attribut von `R` und α ein Element der Domäne von `A`. Dann gilt für das Prädikat "A=α":

$$\xi = \frac{1}{val_A(R)} * f$$

mit `f` als Faktor laut Fallunterscheidung (siehe unten). Dabei steht $val_A(R)$ für die Valenz des Attributes `A`.

Da sowohl die Werte von `A` als auch α potenziell auf Basis von Metainformation auf eine Teilmenge der Domäne von `A` eingeschränkt werden können, müssen für die Selektivitätsberechnung die Überdeckungen der möglichen Werteintervalle W_α und W_A von α und `A` berücksichtigt werden.

Daher ist folgende Fallunterscheidung zu treffen:

1. Fall: Die Werteintervalle W_α und W_A sind identisch. Dies ist der triviale Fall, wenn keine weiteren Meta-Informationen vorliegen. Formal ausgedrückt gilt $W_\alpha = W_A$, was anschaulich in der folgenden Grafik illustriert wird:

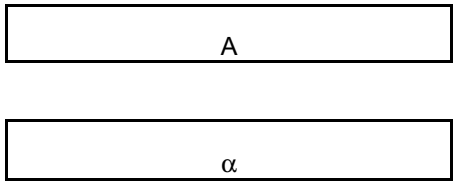


Abbildung 91: Deckungsgleiche Werteintervalle der potenziellen Werte des Attributes und der potenziellen Werte der Variable α

2. Fall: Werteintervalle W_α und W_A überlagern sich, aber nicht vollständig[23]. Formal ausgedrückt gilt $W_A \neq W_\alpha$, was mit folgender Grafik illustriert wird:

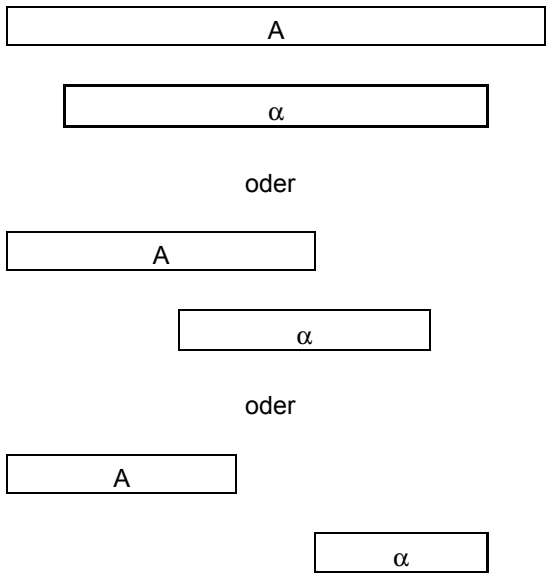


Abbildung 92: Intervalle von α und dem Attribut ohne Deckungsgleichheit

Im ersten Fall gilt f=1. Im zweiten Fall gilt hingegen:

f=p(A,α)*p(α,A) mit

$$p(A,\alpha) = \begin{cases} \dfrac{\min(\max(A),\max(\alpha)) - \max(\min(A),\min(\alpha))}{\max(A) - \min(A)} & falls >= 0 \\ 0, sonst \end{cases}$$

$$p(\alpha,A) = \begin{cases} \dfrac{\min(\max(A),\max(\alpha)) - \max(\min(A),\min(\alpha))}{\max(\alpha) - \min(\alpha)} & falls >= 0 \\ 0, sonst \end{cases}$$

In Abhängigkeit von der Domäne von A sind dann Fallunterscheidungen zu treffen.[24] Wichtig ist, dass eine Differenzierung nach der Mächtigkeit der Domänen nicht erforderlich ist.[25]

[23] In diesem Fall ist also zusätzliche Information vorhanden, mit der eine verfeinerte Abschätzung vorgenommen werden kann. Die Annahme der vollständigen Intervallüberdeckung muss oftmals getroffen werden, weil keine näheren Informationen vorliegen. Es wird dann postuliert, dass die Werte des Attributes und der Variable in einem identischen Intervall gleichverteilt sind.

[24] Die einschlägige Literatur [MIT1995], [DAD1996], [MAN1988] et cetera nimmt die Differenzierung nach den Datentypen der Domänen nicht vor. Gerade der Relationale Operator "Selektion" wird aber auch auf Attributen mit nicht numerischen Domänen ausgeführt. In diesem Falle kann unabhängig von individuellen Datenausprägungen auch statistisch keine Gleichverteilung angenommen werden, wenn natürliche Sprache gespeichert wird. Der Grund dafür sind die unterschiedlichen Wahrscheinlichkeiten des Auftretens der einzelnen Buchstaben und Worte; beispielsweise in der deutschen Sprache (siehe dazu [MEI1978]).

Es spielt also beispielsweise keine Rolle, ob es sich bei der Domäne um natürliche, ganze Zahlen, rationale oder reelle Zahlen handelt. Zu Besonderheiten von STRING-Domänen, siehe Seite 105.

Dabei stellt `p(A,α)` anschaulich den Korrekturfaktor für `A` dar. Es ist der prozentuale Anteil des Werteintervalls von `A`, der das Werteintervall von α überlappt. Analoges gilt für `p(α,A)`. Die folgende Grafik veranschaulicht dies:

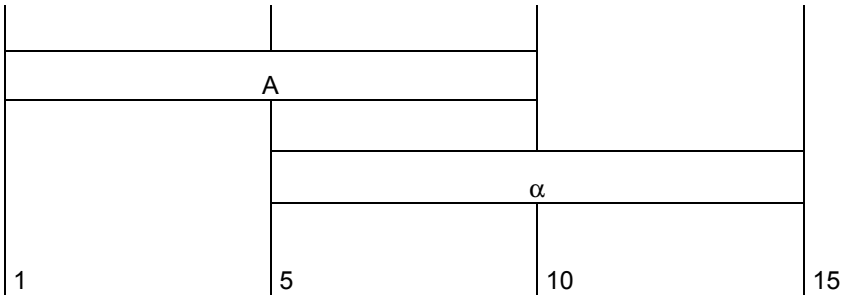


Abbildung 93: Anteilige Intervallüberdeckung zweier Wertebereiche

In den `p(...)` steht der Zähler für die Länge des Überschneidungsbereichs beider Intervalle. Der Nenner repräsentiert das komplette Intervall des jeweiligen Attributes. Im obigen Beispiel würde sich sowohl für α, als auch für A der Wert `5/10=0,5` ergeben.

Durch diese Fallunterscheidung wird dem Umstand Rechnung getragen, dass die Punktselektion (`A=α`) nicht zwingend im Wertebereich der realen Werte von `A` liegen muss. Wenn beispielsweise für die Domäne von `A` `dom(A)=[1001,2000]` mit `|R|=50` und $val_A(R)=50$, sowie `A=[1001,1100]` gilt, folgt mit der "einfachen" Formel `|E|=ξ*|R|=(1/50)*50=1`. Bedenkt man jedoch, dass α Werte zwischen `1001` und `2000` annehmen kann, müsste die Selektivität eigentlich entsprechend (etwa zehnfach) geringer sein. Eben dies berücksichtigt die erweiterte Formel:

$$|E| = \xi * |R| = \frac{1}{50} * f * 50$$

mit `f=p(A,α)*p(α,A)` mit

$$p(A,\alpha) = \frac{\min(1100,2000) - \max(1001,1001)}{1100 - 1001} = \frac{(1100) - (1001)}{1100 - 1001} = 1$$

$$p(\alpha,A) = \frac{\min(1100,2000) - \max(1001,1001)}{(2000) - (1001)} = \frac{(1100) - (1001)}{(2000) - (1001)} = \frac{99}{999} \approx \frac{1}{10}$$

Dies ist zwar ein Sonderfall, der aber in der Praxis durchaus bedeutsam ist.

Wichtig ist in diesem Zusammenhang, dass man als notwendige Voraussetzung zur Berechnung der Selektivität die Grenzen des Werteintervalls von `A` und α kennen muss. Dies kann für `A` natürlich auch eine begrenzte Basisdomäne sein.

Feinere Abschätzungen sind aber aufgrund der im konkreten Fall möglichen Ausprägungen der Werte in `A` möglich. Aus diesen lassen sich oftmals die obere und die untere Grenze des Intervalls ableiten.

[25] Eine Differenzierung nach endlichen, abzählbar unendlichen und überabzählbaren Datentypen (Mengen) ist auch anschaulich nicht erforderlich, da aufgrund der Annahme "endliche Anzahl der Werte in A" die Domäne eines Attributes A immer endlich ist.

Vereinfachung

Im ersten Fall (deckungsgleiche Werteintervalle) ergibt sich noch folgende algebraische Schlussfolgerung:

$$\xi = \frac{1}{\left|DT\left(PJ_i R\right)\right|}$$

mit A als i-tem Attribut (bildet i-te Komponente der Tupel im Sinne der PJ-Definition) da

$$val_A(R) = \left|DT\left(PJ_i R\right)\right|$$

gilt (zu "DT" (DISTINCT-Operator) siehe 7.5.1.5).

Falls A beispielsweise ein Primärschlüsselattribut ist (und α im Werteintervall von A liegt), gilt damit offensichtlich ξ=1/|R|. Und damit folgt |E|=(1/|R|)*|R|=1. Es gilt daher |E|=1, wie für ein Primärschlüsselattribut nicht anders zu erwarten.

Ist A kein Primärschlüsselattribut (α aber immer noch im Werteintervall von A), kann jeder Wert in A mehrfach vorkommen. Die Valenz gibt an, wie viele unterschiedliche Werte A enthält. Ist jeder Wert zweimal vorhanden, gilt beispielsweise $val_A(R)=|R|/2$. Für die Selektivität gilt folglich ξ=1/(|R|/2)=2/|R|. Es gilt damit |E|=(2/|R|)*|R|=2, was wieder offensichtlich ist.

Man kann die Selektivität alternativ wie folgt definieren: ω(A)=|R|/$val_A(R)$ mit ω(A)= mittlere Anzahl der Instanzen eines Wertes auf A. Damit ergibt sich für die Selektivität:

ξ=ω/|R|

Falls A ein Primärschlüsselattribut ist, gilt ξ=1/n, da jeder Wert genau einmal vorkommen kann. Ist jeder Wert zweimal vorhanden, gilt beispielsweise ω=2 und damit ξ=2/|R|. Es gilt damit |E|=(2/|R|)*|R|=2, was wieder offensichtlich ist.

Zeitkomplexität

Die Selektion gehört zur Klasse

O(n*p) für n=|R| und p=Anzahl der zu evaluierenden primitiven Prädikate

Wir betrachten im Folgenden nur den Fall per logischer Konjunktion verknüpfter Prädikate. Dass die Auswertung komplexer Prädikate durchaus entscheidenden Einfluss auf das Zeitverhalten einer SQL-Anfrage haben kann, liegt auf der Hand [HEL1993]. Wir analysieren im Folgenden nur die Zeitkomplexität von Sonderfällen der Selektion unter Einbeziehung primitiver Prädikate mit bestimmten Eigenschaften und deren logischen Junktoren.

Sonderfall 1: $O\left(\left(\log_2 n\right) + m * p\right)$, wenn P genau p primitive Prädikate enthält und mindestens eines der Attribute einen Wert höchstens m mal enthält, dieses in einem Gleichheitsprädikat ("=") verwendet wird und einen Index besitzt; ist die Domäne dieses Attributes zusätzlich begrenzt, ergibt sich sogar O(m*p). In beiden Fällen postulieren wir, dass der Optimizer des DBMS das ausgezeichnete Attribut zuerst auswertet und der Wert "m" für die Evaluation bekannt ist.

Sonderfall 2: $O\left(\left(\log_2 n\right) + p\right)$, wenn alle primitiven Prädikate in P logisch mit demselben Primärschlüsselattribut (das einen Index besitzen muss) per Konjunktion (AND) verbunden sind; ist die Domäne dieses Attributes zusätzlich begrenzt, ergibt sich sogar O(p).

Sonderfall 3: $$O\left(\sum_{i=1\ldots p}\left(\left(\log_2 n\right) + \left(\log_2\left|P_i\right|\right) * \left|P_i\right|\right)\right) + O\left(p * MAX_{j=1\ldots p}\left|P_j\right|\right)$$

mit p= Anzahl der Prädikate, wenn die Attribute aller Prädikate einen Index besitzen, konjunktiv verbunden (AND) sind und $|P_i|$ die Anzahl der Tupel darstellt, die das primitive Prädikat P_i erfüllen.

Die Zeitkomplexität der Selektion ist bei einem einfachen Durchsuchen linear von der Tupelzahl in R abhängig (O(n) mit |R|=n). Auch wenn Datenstrukturen für einen optimierten Zugriff (B-Baum, sortierte Liste et cetera) auf die in P angesprochenen Attribute existieren, ändert sich die obere Schranke für die Zeitkomplexität normalerweise nicht.

Dafür gibt es mehrere Gründe, von denen die wichtigsten im Folgenden aufgeführt werden:

(i) Alle Tupel aus R können potenziell P erfüllen, wenn sich P auch auf Nicht-Primärschlüsselattribute bezieht. In diesem Fall hat ein optimales Selektionsverfahren die Zeitkomplexität $O(1) + O(n) = O(n)$, da nach der Selektionsoperation auf der zugrunde liegenden Datenstruktur für den optimalen Zugriff ($O(1)$) noch alle Tupel t mit $P(t)=true$ selektiert (durchlaufen) werden müssen ($O(n)$). Das ist nötig, da die Selektion immer alle Tupel zurückliefert, die P erfüllen.

Erfüllen nicht alle, sondern nur *m* Tupel aus R das Prädikat P, so gilt allgemein $O(\log_2 n)+O(m)*O(p) = O((\log_2 n)+ m*p)$. Dies ergibt sich aus dem Indexzugriff ($O(\log_2 n)$) und dem folgenden "Scan" der identischen Werte ($O(m)$), bei denen dann noch die übrigen Prädikate[26] geprüft werden müssen (Faktor p). Ist also bekannt, dass ein Wert maximal m mal in einem Attribut vorkommen kann, so kann $O((\log_2 n)+ m*p)$ angenommen werden.

(ii) Das Prädikat P kann sich auf mehrere Attribute aus R beziehen; im besten Fall sind dann entsprechend viele optimierte Zugriffe nötig. Aus den Tupelmengen[27], welche die einzelnen primitiven Prädikate erfüllen, werden dann anschließend die Schnittmengen (Konjunktion; AND) gebildet[28]. Es ist offensichtlich, dass die Summe der Kardinalitäten der Tupelmengen der einzelnen primitiven Prädikate kleiner als n sein muss, damit insgesamt ein optimierter Zugriff sinnvoll ist[29].

Sonderfall 1

Eine Zeitkomplexität von $O((\log_2 n)+ m*p)$ ist anzunehmen, wenn P genau p primitive Prädikate enthält und mindestens eines der Attribute (A_x) einen Wert höchstens m mal enthält, sowie einen Index besitzt. In diesem Fall werden zunächst unter Verwendung des Index alle Tupel aus R selektiert, die das primitive Prädikat des Attributes A_x erfüllen.[30] Anschließend werden die so selektierten Tupel per Durchlauf (Scan) auf Erfüllung der anderen primitiver Prädikate geprüft.

Sonderfall 2

Eine Zeitkomplexität von $O((\log_2 n)+p)$ ist anzunehmen, wenn das Prädikat P mindestens ein Primärschlüsselattribut (keine doppelten Werte) verwendet. Zusätzlich müssen alle übrigen primitiven Prädikate aus P mit diesem konjunktiv (AND) verknüpft sein.

In diesem Fall kann zunächst ein optimierter Zugriff (binäre Suche) in $O(\log_2 n)$ über das Primärschlüsselattribut erfolgen. Voraussetzung ist natürlich die entsprechende Zugriffsstruktur (Index). Nach Auffinden des passenden Tupels[31], müssen zusätzlich lediglich noch die übrigen primitiven Prädikate ($O(p)$) geprüft werden.

---

[26] Von diesen nehmen wir in diesem Fall an, dass Sie keine optimierte Zugriffsstruktur bieten.

[27] Hier werden meist keine "echten" Zwischenrelationen gebildet. Vielmehr kommen Mengen von so genannten TIs zum Einsatz. Jedes Tupel wird mit einem eindeutigen tupel identifier (TI) ausgestattet, mit dem sich elegant Schnittmengen und Vereinigungsmengen von Tupelmengen bilden lassen.

[28] Im Rahmen dieser Arbeit ist die Disjunktion von Prädikaten nicht von Bedeutung.

[29] Für die Bildung der Schnittmengen wird eine Zeitkomplexität von $O(\Sigma_{i=1..p}(|P_i|*\log_2|P_i|))$ benötigt (mit p= Anzahl der primitiven Prädikate und mit $|P_i|$ = Kardinalität der Tupelmenge, die das primitive Prädikat P_i erfüllt). Zur effizienten Schnittmengenbildung müssen die Tis nämlich sortiert vorliegen. Und diese Summe ist spätestens dann größer als $O(n)$, wenn die Kardinalität der einzelnen Tupelmengen n übersteigt.

[30] Dies sind maximal m Tupel, da das ausgezeichnete Attribut einen Wert maximal sooft enthält und ein Gleichheitsprädikat ("=") vorausgesetzt wurde. Die übrigen Prädikate können dann mit einem "Scan" über alle gefundenen Tupel geprüft werden. Ist die Domäne des ausgezeichneten Attributes zusätzlich begrenzt, ergibt sich durch Einsatz eines Hashverfahrens (siehe auch Seite 124) sogar $O(m*p)$. Komplexe Prädikate sind besonders zu berücksichtigen.

[31] Es kann nur eines geben - das ist wesentlich. Kann für das Primärschlüsselattribut zusätzlich ein Hashverfahren mit injektiver Hashfunktion (siehe dazu Seite 124) eingesetzt werden, ergibt sich sogar $O(1)$.

Sonderfall 3

Eine Zeitkomplexität von $O\left(\sum_{i=1\ldots p}\left(\left(\log_2 n\right)+\left(\log_2\left|P_i\right|\right)*\left|P_i\right|\right)\right)+O\left(p*MAX_{j=1\ldots p}\left|P_j\right|\right)$

mit p=Anzahl der primitiven Prädikate, liegt vor, wenn die Attribute die von den primitiven Prädikaten verwendet werden alle einen Index besitzen, konjunktiv verbunden (AND) sind und $|P_i|$ die Anzahl der Tupel darstellt, die das primitive Prädikat P_i erfüllen. Die so entstandenen Tupelmengen werden dann sortiert[32] und anschließend über Merge-Join vereinigt[33].

Voraussetzungen für die Sonderfälle

Es wird die Möglichkeit des sequentiellen Durchlaufens in Sortierordnung vorausgesetzt[34]. Zugriffsstrukturen wie beispielsweise 2-3-Bäume [AHO1974] bieten dies ohne Änderung der sonstigen Zeitkomplexitäten. Auf diese Weise kann zunächst ein Element e selektiert und dann solange das in der Sortierung folgende Element gewählt werden, bis es von e abweicht. So sind effiziente Selektionen auf Nichtprimärschlüssel-Attributen möglich, da hier potenziell mehrere Tupel den gleichen Attributwert besitzen können.

7.5.1.4 Sortieren (ORDER) [unär]

Sei $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g mit $t=(d_1,d_2,\ldots,d_g)\in R$. Dann ist das Sortieren $OD_{A[i]}$ über dem Attribut A_i wie folgt definiert (A[i] wird als analoge Darstellung von A_i verwendet; die Sortierung nach mehreren Attributen oder Funktionen von Attributen ist im Rahmen dieser Arbeit irrelevant; A[i] muss eine lineare Ordnung besitzen, was für die betrachteten Datentypen der Fall ist):

$$OD_{A[i]}R := E = \left\{t \middle| t \in \mathrm{N} \times \left(PJ_{(1,\ldots,g)}R\right) \wedge \left(PJ_0 t_x < PJ_0 t_y \Rightarrow PJ_i t_x \leq PJ_i t_y\right)\right\}$$

mit g als der Anzahl der Attribute in R.

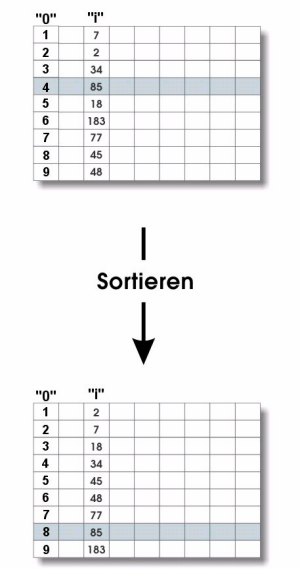


Abbildung 94: Relationaler Operator - Sortieren

[32] $(\log_2|P_i|*|P_i|)$

[33] $p*MAX(|P_i|)$, da die p Tupelmengen maximal soweit durchlaufen werden, bis alle Elemente der größten Menge geprüft sind. Unter "merge-join" ist hier natürlich keine Verbundbildung im Sinne der Relationalen Algebra zu verstehen. Damit ist lediglich eine Vereinigung mit Duplikatelimination im Sinne des Merge-Join-Algorithmus gemeint.

[34] Ein Hash-Index bietet diese Möglichkeit beispielsweise nicht.

Semantik

Das Sortieren einer Relation R wird aufgrund der Werte des angegebenen Attributes A_i durchgeführt.

Kardinalität der Ergebnisrelation

|E|=|R|

Die Sortierung verändert die Anzahl der Tupel der Relation R nicht.

Zeitkomplexität

Das Sortieren gehört zur Klasse

$O(n * \log_2 n)$ für n=|R|

Sonderfall: O(n), wenn sich die Werte des Attributes, nach dem sortiert werden soll, als k-adische Schlüssel interpretieren lassen und sich jede Stelle des Schlüssels maximal b Werte annehmen kann. Sind k und b konstant, folgt dann O((n+b)*k)=O(n).

Die schnellsten Sortierverfahren arbeiten in O(n). Dazu müssen allerdings die oben stehenden Werte k und b Konstant sein. Beispiele für zu sortierende Werte, bei denen dies gegeben ist, sind

(a) die Zahlen von 0 bis 9 (b=10; k=1)

(b) 16-stellige Binärzahlen (b=2;k=20)

(c) 40-stellige Zeichenfolgen von "a" bis "z" (b=26;k=40)

Sortierverfahren, die in solchen Szenarien in O(n) sortieren, sind Bucketsort und Radixsort. Bucketsort [AHO1974; S.77 ff.] arbeitet wie ein Hash-Verfahren, benutzt als Hashfunktion aber die Identität [h(a)=a] und bildet jeden Wert genau auf einen "bucket" ab. Damit lässt sich (a) abdecken. Identische Elemente werden dabei auf den gleichen Bucket abgebildet.

Bei Radixsort wird beginnend mit der letzten Stelle für alle k Stellen der zu sortierenden Zeichenfolgen eine Bucketsort-Verteilung auf die b Buckets (engl., Eimer) vorgenommen. Da die einzelnen Bucketsort-Schritte durch das Einfügen der einsortierten Elemente an das Ende der Bucket-Listen stabil sind, können Zeichenfolgen durch iteratives Bucketsort sortiert werden. Die Anzahl der Buckets bei Zeichenketten über dem natürlichen Alphabet beträgt in der Regel 256 (ASCII-Zeichen).

Die Werte "AA_", "AB_", "CA**C**", "AA**B**" würden mit der ersten Iteration in die Buckets

```
"_" = "AA_", "AB_"
"B" = "AAB"
"C"="CAC"
```

sortiert und zur Liste "A**A**_", "A**B**_", "A**A**B", "C**A**C" die in der nächsten Iteration zu

```
"A"="AA_", "AAB", "CAC"
"B"="AB_"
```

wird und damit zur Liste "**A**A_", "**A**AB", "**C**AC", "**A**B_" die in der finalen Iteration (k=3) zu

```
"A"="AA_", "AAB", "AB_"
"C"="CAC"
```

Wird, was die sortierte Ausgabe "AA_", "AAB", "AB_", "CAC" liefert.

Übertragen auf DBMS kommen solche Verfahren daher nur für Attribute in Frage, die (a), (b) und (c) erfüllen.

Sortierverfahren, die nicht auf begrenzte Domänen angewiesen sind (beispielsweise Heapsort; siehe [AHO1974], [SED1995] arbeiten in $O(n*\log_2 n)$.

7.5.1.5 Duplikatelimination (DISTINCT) [unär]

Sei $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g mit $t=(d_1,d_2,\ldots,d_g) \in R$. Dann ist die Duplikatelimination DT wie folgt definiert:

$DT\ R:=E=\{t \mid t=(A_{ID},A_1,\ldots,A_g) \in R\}$ und für alle Tupel $t_x, t_y \in E$ mit $x \neq y$ gelte $t_x[j] \neq t_y[j]$ für mindestens ein j, $j \neq 0$.

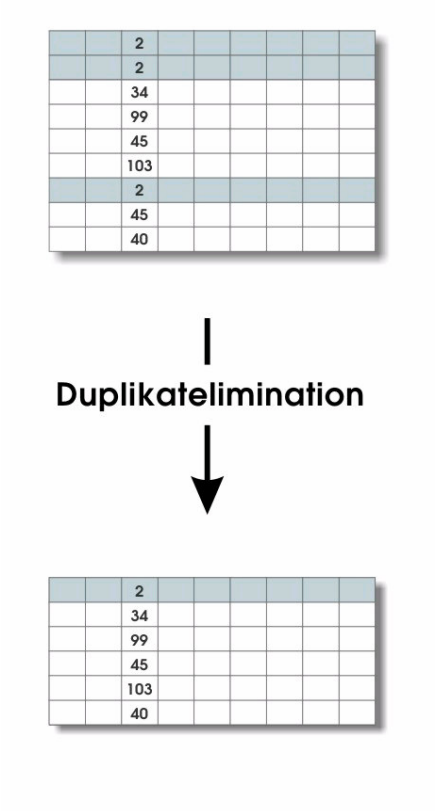


Abbildung 95: Relationaler Operator - Duplikatelimination

Semantik

Die Duplikatelimination auf einer Relation R wird aufgrund aller Attribute der Relation R durchgeführt. Bei k identischen Tupeln werden $k-1$ dieser Tupel nicht in die Ergebnisrelation übernommen.

Kardinalität der Ergebnisrelation

$$|E| <= \min\left(\prod_{i=1\ldots n} val_{A[i]}(R), |R|\right)$$

Hier wird das "<=" verwendet, da es sich um eine Abschätzung der oberen Schranke handelt, indem das Produkt der Valenzen der Attribute aus R gebildet wird.

Ist $val_{A[i]}(R)$ nicht bekannt, aber mit ω_i die durchschnittliche Anzahl der Instanzen eines Wertes auf $A[i]$, so kann $val_{A[i]}(R)$ aus ω_i abgeleitet werden: $val_{A[i]}(R)=|R|/\omega_i$. Damit ergibt sich dann für die Kardinalität der Ergebnisrelation:

$$|E| <= \min\left(\prod_{i=1\ldots n}\left(\frac{|R|}{\omega_i}\right), |R|\right)$$

Zeitkomplexität

Die Duplikatelimination gehört zur Klasse

$O(n * \log_2 n)$ für n=|R|

Sonderfall 1: O(1), falls die Relation R ein Primärschlüsselattribut enthält.

Sonderfall 1

Wenn in Relation R mindestens ein Primärschlüsselattribut enthalten ist, kann es per Definition keine identischen Tupel geben. Daraus folgt die Zeitkomplexität O(1).

7.5.1.6 Gleichverbund (EQUIJOIN) [binär]

Seien $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g und $S \subseteq RS\{B_1, \ldots, B_h\}$ eine Relationen vom Grad h. Dann ist der Gleichverbund JN wie folgt definiert:

$JN_{A[i]=B[j]}(R,S) := E(A_1, \ldots, A_g, B_1, \ldots, B_h)$

mit $E=\{t=(A_1,\ldots,A_g,B_1,\ldots,B_h) \mid t \in RxS \wedge A_i=B_j\}$ wobei der tuple identifier A_0 in E neu durchnummeriert wird.

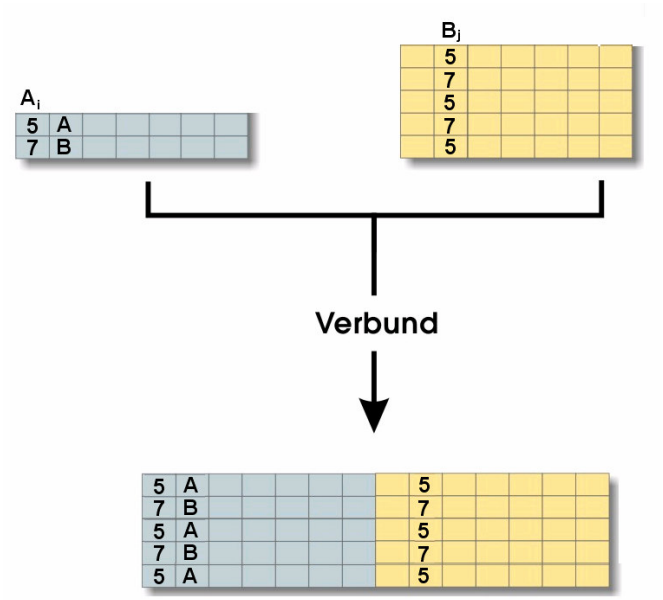


Abbildung 96: Relationaler Operator – Gleichverbund

Semantik

Die Bildung des Gleichverbundes zweier Relationen R und S erfolgt in zwei Schritten. Zunächst wird das kartesische Produkt (Kreuzprodukt) beider Relationen gebildet; dann wird dieses durch das Gleichheitsprädikat $A_i=B_j$ eingeschränkt. Die Ergebnisrelation E enthält alle Attribute beider Relationen und den tuple identifier A_0 der neu durchnummeriert wird.

Die im Gleichheitsprädikat $A_i=B_j$ verwendeten Attribute müssen bezüglich der zugrunde liegenden Domänen die gleichen Basisdomänen (siehe Seite 104) besitzen. Man kann also beispielsweise nicht Attribute auf den Basisdomänen STRING und INTEGER vergleichen. Ferner wird ein verlustfreier Verbund angenommen, der beim Einsatz von Primär- und Fremdschlüsseln durch das DBMS als so genannte referentielle Integrität sichergestellt wird.

Kardinalität der Ergebnisrelation

`|E|= |S|*(ξ*|R|)` mit

$$\xi = \frac{1}{val_{A[i]}(R)} = \frac{1}{val_{B[j]}(S)}$$

Dies lässt sich aus der (*average case*) Selektivität des Relationalen Operators *Selektion* ableiten. Aus

$$\xi = \frac{1}{val_{A[i]}(R)} * f$$

wird

$$\xi = \frac{1}{val_{A[i]}(R)}$$

Denn der Faktor `f` der *Selektion* muss nicht berücksichtigt werden, da ein verlustfreier Verbund angenommen wird, was zu identischen Werteintervallen der Verbundattribute führt.

Das kartesische Produkt kann (bezüglich der Selektion auf dem Verbundattribut) als `|S|`-fache Konkatenation der Relation `R` betrachtet werden. Auf dem Kreuzprodukt wird dann ebenfalls `|S|`-fach ein einfaches Selektionsprädikat (Gleichheitsprädikat der Verbundattribute) ausgeführt. Denn beim Verbund liefern `k` identisch Werte auf dem Verbundattribut auch mindestens `k` Tupel in der Ergebnisrelation (die Valenz spielt hier also keine Rolle).

Sonderfall 1:N

`|E|=max(|R|,|S|)`

Im Falle einer 1:N-Beziehung zwischen den Relationen R und S über deren Attribute A_i und B_j (mit einem der beiden als Primärschlüsselattribut) vereinfacht sich die Ableitungsformel offensichtlich wie angegeben. Denn da für eine Relation (die mit dem Primärschlüsselattribut als Verbundattribut) die Kardinalität der Relation gleich der Valenz des Verbundattributes ist, werden alle Tupel der anderen Relation genau einmal in die Verbundrelation aufgenommen. Oder formal:

`|E|= |S|*(ξ*|R|)` mit

$$\xi = \frac{1}{val_{A[i]}(R)} = \frac{1}{val_{B[j]}(S)}$$

ist

$$|E| = |S| * \frac{1}{val_{A[i]}(R)} * |R|$$

Mit `R` als Relation deren Verbundattribut Primärschlüsseleigenschaft besitzt (analog zeigt man dies für `S` als Relation mit Primärschlüsselattribut) folgt daraus

$$|E| = |S| * \frac{1}{val_{A[i]}(R)} * val_{A[i]}(R) = |S|$$

Und wegen |S|>=|R| folgt allgemein `|E|=max(|R|,|S|)`.

Zeitkomplexität[35]

Der Verbund gehört zur Klasse

$O(n * \log_2 n) + (m * \log_2 m)$ mit n=|R| und m=|S|, falls die Verbundattribute A_i und B_j nicht hash-geeignet sind und die Verbundattribute nicht sortiert vorliegen, was wir als Standardfall annehmen wollen.

Sonderfall 1: O(n+m) mit n=|R| und m=|S|, falls die Verbundattribute A_i und B_j nicht hash-geeignet sind, die Verbundattribute aber sortiert vorliegen.

Sonderfall 2: O(n+m) mit n=|R| und m=|S| falls die Domäne eines der Verbundattribute A_i und B_j klein genug ist, um eine injektive Hashfunktion einzusetzen; Sortierung der Verbundattribute ist in diesem Falle nicht erforderlich

Zum Aufbau eines Gleichverbundes sind in der Literatur zahlreiche Ansätze zu finden. Im Folgenden sollen nur die effizientesten vorgestellt werden, aus denen sich auch die angegebenen Zeitkomplexitäten ableiten.

Der Mischverbund basiert auf dem parallelen Durchlauf der zuvor zu sortierenden Verbundattribute (zwingende Voraussetzung) A_i und B_j der Relationen R und S. Der zugrunde liegende Algorithmus arbeitet wie folgt (wobei x und y Indizes für die *tupel identifier* seien):

```
Setze x=1, y=1
Sortiere Verbundattribut Ai
Sortiere Verbundattribut Bj
solange x<=|R| und y<=|S|
      solange Ai[x]<=Bj[y]  (R-Durchlauf, S "halten")
            wenn Ai[x]=Bj[y]  (Gleichheit der Verbundattribute)
                  Füge (tR[x],tS[y]) als Tupel zur
                  Ergebnisrelation hinzu
            sonst
                  x:=x+1  (nächster Wert aus Relation R)
      solange Bj[y]<=Ai[x]  (S-Durchlauf, "R" halten)
            wenn Bj[y]=Ai[x]  (Gleichheit der Verbundattribute)
                  Füge (tR[x],tS[y]) als Tupel zur
                  Ergebnisrelation hinzu
            sonst
                  y:=y+1  (nächster Wert aus Relation S)
```

Algorithmus 3: Erzeugen eines Mischverbundes

Da grundsätzlich immer einer der beiden Indizes (x,y) um eins erhöht wird, diese aber die Maximalwerte |R| beziehungsweise |S| annehmen, ergibt sich O(n+m) mit n=|R| und m=|S|, falls R und S bereits bezüglich der Verbundattribute A_i und B_j sortiert vorliegen.

[35] Die Zeitkomplexität des Gleichverbundes wurde im Wesentlichen aufbauend auf folgenden Quellen abgeleitet: [SHA1986], [RAM1988], [SHE1990], [JOH1993], [AHO1974], [SED1995]. Ein Hash-Verbund ohne injektive Hashfunktion kann in der Praxis dem Mischverbund überlegen sein. Es wurde dennoch der Mischverbund als Verbundalgorithmus angenommen, da nur er eine Laufzeit von $O(n * \log_2 m)$ garantieren kann. In Abhängigkeit von der Datenausprägung kann ein Hashverfahren immer eine Zeitkomplexität von $O(n^2)$ annehmen. Wie in [SHA1986] gezeigt, ist ein Hash-Join ab einer bestimmten Hauptspeichergröße (ermöglicht große Hash-Tabellen und reduziert Kollisionswahrscheinlichkeit) im Mischverbund im average-case weit überlegen (O(n+m) statt $O(n*\log_2 m)$).

Ist letzteres nicht der Fall, erhöht sich der Aufwand um

`O(n*log2n)` für `R` und

`O(m * log2m)` für `S` auf insgesamt

`O(n*log2n) + O(m*log2m) + O(n) + O(m)`

`= O(p*log2p)`

mit `p=max(n,m)`.

Sonderfall 1

Wenn die Verbundattribute A_i und B_j bereits in sortierter Form vorliegen reduziert sich die Zeitkomplexität des Algorithmus auf `O(n+m)`.

Sonderfall 2

Beim Hash-Verbund wird die Relation mit der kleineren Kardinalität in einer Hash-Tabelle abgelegt. Die kleine Kardinalität ist bedeutsam, da Hash-Verfahren normalerweise die Zeitkomplexität `O(n)` besitzen, wenn die Hashfunktion nicht injektiv ist. Um Robert Sedgewick [SED1995] zu zitieren: "Hashing ist ein gutes Beispiel für einen Kompromiss zwischen Zeit- und Platzbedarf. Wenn es keine Beschränkung des Speichers gäbe, könnten wir jede beliebige Suche mit nur einem Zugriff auf den Speicher ausführen, indem wir einfach den Schlüssel als Speicheradresse verwenden". Die Identität ("Schlüssel gleich Speicheradresse") ist eine injektive Funktion.

Sobald eine nicht-injektive Hashfunktion zum Einsatz kommt, sind Kollisionen möglich. Unter einer Kollision versteht man die Abbildung verschiedener Werte auf den gleichen Platz der Hash-Tabelle.

In diesem Kontext wird oft das so genannte *Geburtstags Paradoxon* angeführt. Es besagt, dass bereits bei 23 Personen die Wahrscheinlichkeit, dass zwei dieser Personen am gleichen Tag Geburtstag haben, bei über 50% liegt. Bei 50 Personen liegt diese Wahrscheinlichkeit bereits bei 97%.

Die Wahrscheinlichkeit einer Kollision beträgt bei einer 365 Behälter umfassenden Hash-Tabelle und einer Relation, deren Verbundattribut 50 Werte umfasst bereits 0,97. Allgemein gilt für die Wahrscheinlichkeit einer Kollision von `n` Werten auf eine `m` Behälter umfassende Hash-Tabelle [GUE1990]:

$$P(Kollision(n,m)) = 1 - \frac{m!}{m^n * (m-n)!}$$

Die kollisionsfreie Zuordnung eines Wertes `w` auf einen Platz einer leeren Hash-Tabelle hat eine Wahrscheinlichkeit `P(kf(1,m))=m/m=1`. Für die kollisionsfreie Zuordnung eines weiteren Wertes gilt dann offensichtlich

$$P(kf(2,m)) = \frac{m-1}{m}$$

Für den i-ten Wert gilt

$$P(kf(i,m)) = \frac{m-(i+1)}{m}$$

Damit folgt für eine kollisionsfreie Zuordnung von `n` Elementen:

$$P(keineKollision(n,m)) = \frac{m}{m} * \frac{m-1}{m} * \ldots * \frac{m-(n+1)}{m}$$
$$= \frac{m*(m-1)*\ldots*(m-(n+1))}{m^n}$$
$$= \frac{m!}{(m-n)!*m^n}$$

Und damit folgt wegen `P(Kollision(n,m))=1- P(keineKollision(n,m))` für `P(Kollision(n,m))` die angegebene Wahrscheinlichkeit. Dies zeigt, dass Kollisionen nicht vermeidbar sind, es sei denn, man setzt ein bucket-Verfahren ein. Im worst case können sonst sogar alle Werte auf einem Platz der Hash-Tabelle abgebildet werden, was zu `O(n)` führt.

Hash-Verfahren mit injektiver Hashfunktion

Ein Hash-Verfahren garantiert die Selektion eines Wertes in $O(1)$ nur dann, wenn eine injektive Hashfunktion zum Einsatz kommt (siehe [RAM1988];[SHA1986], S.250 ff.;[AHO1974], S. 111 ff.). Daraus folgt aber, dass die Anzahl der Behälter der Hash-Tabelle größer oder gleich der Kardinalität der Relation sein muss. Bei sehr großen Domänen kann folglich kein effizientes Hash-Verfahren eingesetzt werden. Jedes "normale" Hash-Verfahren zur Verbundbildung hat als obere Schranke der Zeitkomplexität von $O(n^2)$, da potenziell alle Werte auf einen Platz der Hash-Tabelle abgebildet werden können.

Ist jedoch eine endliche Domäne mit injektiver Hashfunktion gegeben, kann folgendes Verfahren eingesetzt werden (T_i ist das Verbundattribut aus der Relation, die nicht in der Hash-Tabelle abgelegt wird):

```
Setze x=1
Wenn S die Relation mit der kleineren Kardinalität
      Setze H:=S (S wird in der Hash-Tabelle abgelegt)
      Erzeuge die Hash-Tabelle H mit h(Bj)=Bj für alle t∈S
      Setze T:=R (R wird durchlaufen)
Sonst
      Setze H:=R (R wird in der Hash-Tabelle abgelegt)
      Erzeuge die Hash-Tabelle H mit h(Ai)=Ai für alle t∈R
      Setze T:=S (S wird durchlaufen)
solange x<=|T|
      wenn h(Ti[x]) nicht NULL (Gleichheit der Verbundattribute)
            Füge (tR[x],tS[y]) als Tupel zur
            Ergebnisrelation hinzu
      x:=x+1 (nächster Wert aus Relation R)
```

Algorithmus 4: Gleichverbund mit O(1) Hash-Verfahren

Endliche Domänen

Normalerweise sind alle Basisdomänen (siehe Seite 104) bis auf `BOOLEAN` unendlich, was den Einsatz eines Hash-Verfahrens problematisch macht. In der Praxis werden aber oftmals nur Teilmengen der Basisdomänen benötigt, weshalb kommerzielle Systeme diese auch in Form eigener Datentypen für Attribute anbieten.

Hier einige Beispiele:

Domäne	**Teilmenge**	**Systeme**	**Beschreibung**
STRING	CHAR[n]	DB2, Informix, Oracle, SQLBase, SQLServer, MySQL	Zeichenkette mit maximaler Länge (n)
INTEGER	SMALLINT	DB2, Informix, Oracle, SQLBase, SQLServer, MySQL	Ganzzahlige Werte im Wertebereich von –32768 bis +32767
INTEGER	TINYINT	SQLServer, MySQL	Ganzzahlige Werte im Wertebereich von 0 bis 255

Tabelle 4: Beispiele für endliche Teilmengen von Basisdomänen

Endliche Domäne bei endlicher Wertemenge

Zudem können beim Aufbau einer Hash-Tabelle Minimum (MIN) und Maximum (MAX) der Werte des Schlüsselattributes ermittelt werden (ohne zusätzliche Kosten). Das Intervall [Min,Max] enthält demnach die benötigte Teilmenge der Basisdomäne und ist endlich. Für die Basisdomäne INTEGER kann man daraus beispielsweise mit Max–Min die Anzahl der maximal benötigten Hash-Behälter bestimmen.

7.5.1.7 Vereinigung und Tupel einfügen (UNION/INSERT) [binär]

Seien $R, S \subseteq RS\{A_1, \ldots, A_g\}$ Relationen vom Grad g. Dann ist die Vereinigung UN wie folgt definiert:

$UN(R,S) := R \cup S$

Ferner $A_{ID} := A_{ID} + |R|$ für alle Tupel aus S, wodurch identische Tupel aus R und S dennoch als (im tuple identifier) unterschiedliche Tupel in UN(R,S) aufgenommen werden.

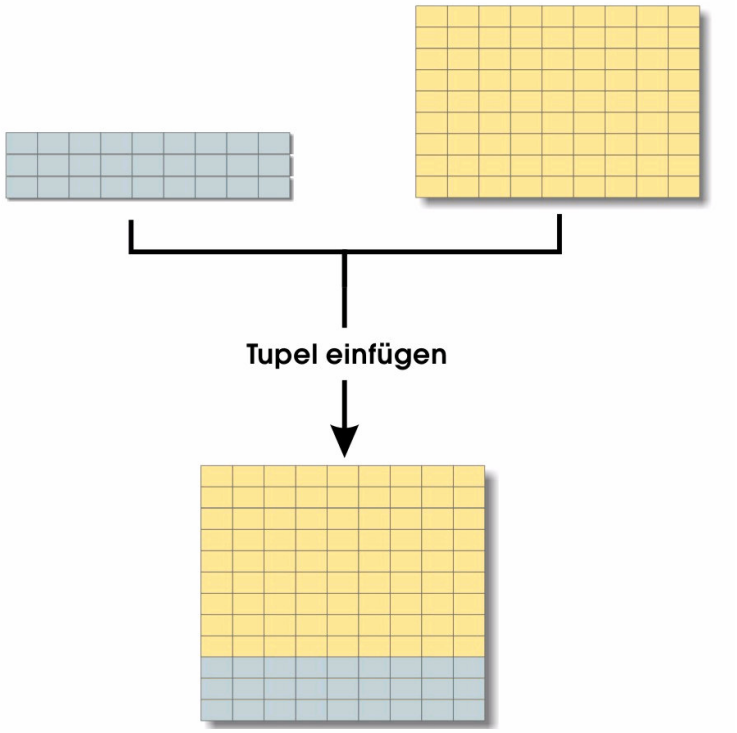


Abbildung 97: Relationaler Operator – Vereinigung beziehungsweise Tupel einfügen

Semantik

Die Vereinigung zweier Relationen R und S mit identischem Relationenschema führt zu einer Ergebnisrelation E, die aus allen Tupeln der Relation R und S besteht, wobei die Komponente B_{ID} von S von den Werten des Attributes A_{ID} im Rahmen des Aufbaus von E disjunkt gehalten wird.

Das Einfügen eines einzelnen Tupels t (dessen Komponenten bezüglich der Domänen mit den Domänen der Attribute der Relation R übereinstimmen müssen) kann man als Sonderfall der Vereinigung betrachten:

$INS(R,t) := UN(R,S)$ mit $S=\{t\}$.

Das Einfügen von m neuen Tupeln in R entspricht ebenfalls der Vereinigung mit einer Relation S. Der einzige Unterschied besteht darin, dass die Ergebnisrelation auf Basis der bereits bestehenden Relation R aufgebaut werden kann, was bei einer aufrecht zu erhaltenden Sortierung zu einer deutlich besseren Laufzeit als bei der Vereinigung führt.

Das Einfügen eines Tupels t in eine Relation R ist nur möglich, wenn die Werte des Tupels und die Attribute der Relation bezüglich der zugrunde liegenden Domänen die gleichen Basisdomänen (siehe Seite 104) besitzen. Es wird eine Ergebnisrelation E erzeugt, die mit R übereinstimmt, zusätzlich aber noch t enthält. Wir betrachten im Folgenden direkt den allgemeinen Fall, in dem auch eine Tupelmenge $\{t_1, \ldots, t_T\}$ eingefügt werden kann.

Kardinalität der Ergebnisrelation

`|E|=|R|+|S|`

Zeitkomplexität

Die Vereinigung beziehungsweise das Tupel-Einfügen gehört zur Klasse

`O(m+n)` für die Vereinigung der Relationen `R` und `S`

`O(m)` für das Einfügen von m neuen Datensätzen in `R`

Die Vereinigung zweier Relationen erfordert offensichtlich[36] `O(n+m)` Operationen, da beide Relationen durchlaufen werden müssen, um die Ergebnisrelation zu erzeugen.

Sollen `m` neue Tupel in die Relation `R` eingefügt werden, so müssen diese `m` Tupel durchlaufen werden. Beim Einfügen neuer Tupel in eine bestehende Relation `R` ist folgender Sonderfall interessant:

Sonderfall: $$O\left(\log_2\left(\frac{(n+(m-1))!}{(n-1)!}\right)\right),$$

falls `m` neue Tupel in die Relation `R` einzufügen sind und eine Sortierordnung auf einem Attribut von `R` aufrechterhalten werden muss.

Soll eine Sortierordnung auf einem Attribut von `R` aufrechterhalten werden, so lässt sich die Operation nicht mehr in `O(n+m)` durchführen. Die Kosten für die Aufrechterhaltung der Sortierordnung hängen von der verwendeten Datenstruktur ab. Bei einem effizienten Verfahren wie beispielsweise den 2-3-Bäumen [AHO1974] entstehen für das Einfügen eines Tupels auf einer Relation mit `n` Tupeln Kosten in Höhe von $O(\log_2 n)$.

Sonderfall: Bulk-Insert in eine bereits befüllte Relation

Meist wird beim Einfügen mehrerer Datensätze ("bulk insert") vom "maximalen" Zeitverhalten ausgegangen [MIT1995, DAD1996 et cetera]. Werden `m` Tupel in eine Relation R mit bereits `n` enthaltenen Tupel eingefügt, folgt daraus:

$$O(m*\log_2(m+n))$$

Diese Abschätzung ist allerdings recht grob, da beim Einfügen sehr großer Tupelmengen in Relationen mit aufrecht zu erhaltendem Sortierindex signifikante Abweichungen zu der feineren Abschätzung entstehen:

$$O(\log_2(n)+\log_2(n+1)+\ldots+\log_2(n+m-1)).$$

Dies lässt sich wie folgt umformen:

$$\begin{aligned}
&O(\log_2(n)+\log_2(n+1)+\ldots+\log_2(n+(m-1))\\
&=O(\log_2((n)*(n+1)*\ldots*(n+(m-1))))\\
&=O(\log_2((n)*(n+1)*\ldots*(n+(m-1))))\\
&=O(\log_2((n+(m-1))*(n+(m-2))*\ldots*(n+(m-m))))\\
&=O\left(\log_2\frac{(n+(m-1))!}{(n-1)!}\right)
\end{aligned}$$

[36] Theoretisch könnte ein Datenbanksystem eine interne (physische) Datenstruktur verwenden, die das Zusammenführen von zwei Relationen `R` und `S` in konstanter Zeit ermöglicht. Voraussetzung dafür wäre aber, dass auch alle Ergebnisrelationen der Relationalen Operatoren physisch aufgebaut werden. Zur Wahrung der Unabhängigkeit von einem konkreten DBMS wird hier aber anstelle von `O(1)` die Zeitkomplexität `O(n+m)` angenommen.

7.5.1.8 Tupel löschen (DELETE) [unär]

Sei $R \subseteq RS\{A_1, ..., A_g\}$ eine Relation vom Grad g und $S=SL_PR$ eine durch das Selektionsprädikat P definierte Teilmenge der Tupel von R. Dann ist das Tupel-Löschen DEL wie folgt definiert:

$DEL_P(R) := \{t \mid t \in R \wedge t \notin S\}$

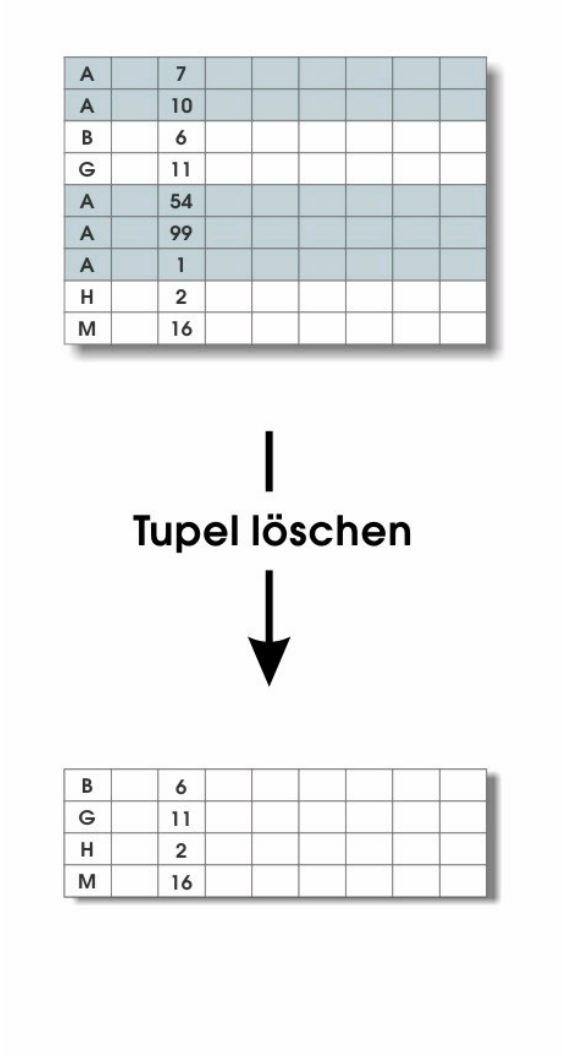


Abbildung 98: Relationaler Operator – Tupel löschen mit Prädikat A_1="A"

Semantik

Das Löschen einer Tupelmenge $S=SL_PR$ (definiert durch ein Prädikat P) aus einer Relation R kann in zwei Schritte zerlegt werden (sei im Folgenden m die Anzahl der Tupel in S). Zunächst erfolgt eine Selektion anhand der definierten Selektionsfunktion[37] (siehe Seite 110), dann werden die selektierten Tupel aus R entfernt.

Kardinalität der Ergebnisrelation

$|E| = |R| - |S|$ mit $S=SL_PP$

[37] In klassischen DBMS darf hier nur auf einer Relation selektiert werden; insbesondere wird ein Löschen aus Verbünden nicht unterstützt. Diese Einschränkung hat allerdings keine Auswirkung auf die folgenden Untersuchungen.

Zeitkomplexität

Das Tupel-Löschen gehört zur Klasse

$O(n*p)$ für $n=|R|$ und p=Anzahl der zu evaluierenden primitiven Prädikate, falls keine Sortierung auf einem der zu ändernden Attribute aufrechterhalten werden muss.

In $O(n*p)$ werden die durch das Prädikat P spezifizierten Tupel selektiert. Integriert man die Lösch-Operation in den Selektionsalgorithmus[38] gilt daher $O(n*p)$ für das Tupel-Löschen.

Sonderfall: $$O\left(\log_2 \frac{n!}{(n-m)!}\right) + O(n*p),$$

falls eine Sortierordnung auf einem Attribut $A_{i1},\ldots,A_{ig}$ der Relation R aufrechterhalten werden muss. Die Sortierung wird zwar auf der Relation durch die Löschoperation erhalten, das DBMS muss aber die zugehörige Index-Struktur aktualisieren.

Besitzt R eine Sortierordnung auf einem beliebigen Attribut, die aufrechterhalten werden soll, so lässt sich die Lösch-Operation eines einzelnen Tupels nicht mehr in konstanter Zeit durchführen. Die Kosten für die Aufrechterhaltung der Sortierordnung hängen von der verwendeten Datenstruktur ab. Bei einem effizienten Verfahren entstehen für das Löschen eines Tupels Kosten in Höhe von $O(\log_2 n)$. Insgesamt ergibt sich damit eine beim Löschen von m Tupeln eine Zeitkomplexität von

$$O(\log_2(n)+\log_2(n-1)+\ldots+\log_2(n-m+1)).$$

Dies lässt sich wie folgt umformen:

$$O(\log_2(n)+\log_2(n-1)+\ldots+\log_2(n-(m+1)))$$
$$=O(\log_2((n)*(n-1)*\ldots*(n-(m+1)))$$
$$=O\left(\log_2 \frac{n!}{(n-m)!}\right)$$

Und damit ergibt sich insgesamt eine obere Schranke von

$$O\left(\log_2 \frac{n!}{(n-m)!}\right) + O(n*p)$$

wenn die Aufrechterhaltung der Sortierung in die Selektion eingearbeitet wird.

7.5.1.9 Tupel ändern (UPDATE) [unär]

Sei $R \subseteq RS\{A_1, \ldots, A_g\}$ eine Relation vom Grad g und $S=SL_PR$.

Das Ändern einer Tupelmenge S (definiert durch eine Selektion mit dem Prädikat P) auf einer Relation R kann analog zum Löschen einer Tupelmenge in zwei Schritte zerlegt werden (sei auch hier im Folgenden m die Anzahl der Tupel in S). Zunächst erfolgt eine Selektion, dann werden die selektierten Tupel aus R aktualisiert:

$UPD_P(R,A_i=w_i) := \{t \mid t\in R \wedge P(t)=false\} \cup \{t[t_i/w_i \mid i\in L] \mid t\in R \wedge P(t)=true\}$

mit $i\in L$ und $L\subseteq\{1,\ldots,g\}$. Dabei ist w_i der Wert, welcher der i-ten Komponente aller Tupel t zugeordnet wird, die das Attribut P erfüllen.

[38] Beim Aufbau der "letzten" Ergebnisrelation im Operatorbaum (siehe Abschnitt 7.6, Seite 135 ff.) kann offensichtlich jedes zugehörige Tupel entfernt werden.

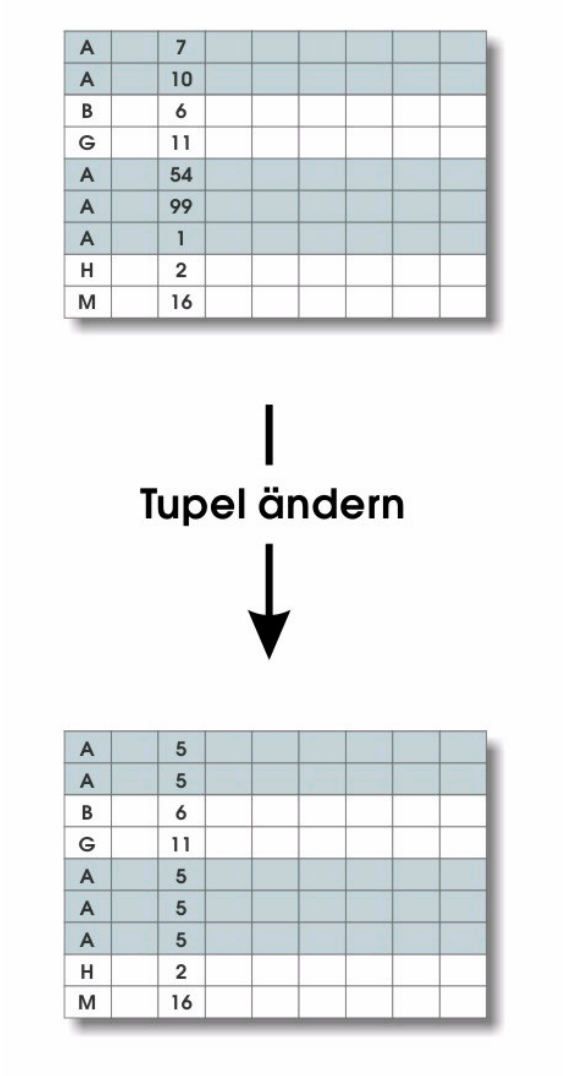


Abbildung 99: Relationaler Operator – Tupel ändern

Semantik

Das Ändern eines Tupels `t` in einer Relation `R` kann anschaulich in zwei Schritte zerlegt werden. Zunächst erfolgt eine Selektion anhand der definierten Selektionsfunktion (siehe Seite 110), dann werden die im Operator spezifizierten Attribute der selektierten Tupel auf die angegebenen Werte gesetzt.

Kardinalität der Ergebnisrelation

`|E|=|R|`

Zeitkomplexität

Das Tupel-Ändern gehört zur Klasse

`O(n*p)` für `n=|R|`

falls keine Sortierung auf einem der zu ändernden Attribute aufrechterhalten werden muss.

Das Ändern eines Tupels ist bezüglich der Zeitkomplexität als konstante Operation anzusehen. Zuvor müssen allerdings die durch das Prädikat `P` spezifizierten Tupel selektiert werden (`O(n*p)`). Integriert man die Update-Operation in den Selektionsalgorithmus gilt daher `O(1)+ O(n*p)= O(n*p)` mit `|R|=n`.

Sonderfall: $O(2*m*\log_2 n)+O(n*p)$, falls eine Sortierordnung auf einem Attribut der Relation `R` aufrechterhalten werden muss.

Besitzt R eine Sortierordnung (auf einem beliebigen Attribut), die aufrechterhalten werden soll, so lässt sich die Änderungs-Operation eines einzelnen Tupels nicht mehr in konstanter Zeit durchführen. Die Kosten für die Aufrechterhaltung der Sortierordnung hängen auch hier wieder von der verwendeten Datenstruktur ab. Anschaulich wird das Tupel quasi aus der Sortierordnung entfernt und anschließend – mit dem geänderten Wert – wieder in die Sortierordnung integriert. Bei einem effizienten Verfahren entstehen dafür Kosten in Höhe von $2*O(\log_2 n)$. Insgesamt ergibt sich damit eine Zeitkomplexität von $O(2*m*\log_2 n)+O(SL_P R)$[39], wenn die Aufrechterhaltung der Sortierung in die Selektion eingearbeitet wird.

7.5.2 Selektivität[40]

Die Zeitkomplexitäten der im Abschnitt 7.5.1 vorgestellten Relationalen Operatoren hängen fast ausschließlich von den Kardinalitäten der Relationen ab, auf welche die Operatoren angewendet werden. Da aber nur in seltenen Fällen die Kardinalität einer Relation bei Anwendung eines Operators unverändert bleibt (|OP(R)|=|R|), muss ein Weg gefunden werden, |OP(R)| zu berechnen.

Vereinfacht ausgedrückt geht es darum, die Anzahl der Tupel der Ergebnisrelation abzuschätzen. Abhängig ist die Kardinalität der Ergebnisrelation von der Selektivität des Relationalen Operators. Im Folgenden wird zu den einzelnen Relationalen Operatoren die Selektivität teilweise nur indirekt durch die Kardinalität der Ergebnisrelation E angegeben.

7.5.2.1 Bedeutung der Selektivität

Aufgrund der sequentiellen Ausführung der Relationalen Operatoren des Operatorbaums (siehe Seite 139) kommt der Selektivitätsabschätzung eine entscheidende Rolle bei der Evaluation der betrachteten Verfahren zu. Ob ein Sortierverfahren mit der Zeitkomplexität $O(n*\log_2 n)$ auf 10.000 oder auf 1.000 Datensätzen durchgeführt wird, ist offensichtlich ein signifikanter Unterschied.

Letztendlich basiert auch das in Abschnitt 7.6.1.7 (Seite 139) beschriebene Verfahren zur Optimierung des Operatorbaumes nicht nur auf Kosten-, sondern auch auf Selektivitätsabschätzungen. Operatoren, die eine hohe Selektivität (und niedrige Kosten) besitzen, sollten zuerst ausgeführt werden. Dieser Ansatz führt zu dem in [HEL1993] (Seite 271, ff.) beschriebenen *Rank*-Verfahren. Danach berechnet man für alle Operatoren Op des Operatorbaumes den Rang wie folgt:

$$Rang(Op) = \frac{sel_{op} - 1}{cpt}$$

mit cpt= Kosten des Operators pro Tupel und sel_{op}= Selektivität des Operators Op. Der cpt Wert wird dabei durch eine Approximation (siehe [HEL1993], Seite 269) berechnet.

7.5.2.2 Verfahren zur Abschätzung der Selektivität

Die Verfahren zur Abschätzung der Selektivität eines Operators lassen sich im Wesentlichen in drei Gruppen aufteilen. Diese sollen nun kurz vorgestellt werden.

Testende Methoden

Die testenden Verfahren (Sampling-Verfahren; siehe [LIP1990], [LIP1990b], [HAA1992]) nutzen einen Bruchteil der Tupel der an einer Operation beteiligten Relation(en), um die Selektivität des Operators abzuschätzen. Dazu wird der Operator auf die gewählte Testmenge T der Tupel angewendet und so |E| durch die Näherung |Op(T)| berechnet. Dieser Ansatz soll hier nicht weiter verfolgt werden, da er keine allgemeingültigen Aussagen ermöglicht.

[39] Hier gilt die klassische Abschätzung (siehe weiter oben) da sich die Kardinalität der Grundmenge zur Ermittlung von $O(\log_2 n)$ im Gegensatz zu den Operatoren *Einfügen* und *Löschen* nicht verändert.

[40] Als mathematische Basis für die Berechnungen im Kontext der Selektivität dienten vor allem [BRO1987] sowie [JAC1992]

Theoretische Methoden

Theoretische Methoden verwenden Funktionen, die auf allgemeinen statistischen Annahmen beruhen, um die Selektivität eines Operators abzuschätzen. Solch eine allgemeine Annahme stellt beispielsweise die Anzahl unterschiedlicher Werte in einem Attribut A dar. Bei 100 Tupeln und der aus der Basisdomäne INTEGER (siehe Seite 104) abgeleiteten Domäne [0,200] wird A 100 verschiedene Werte enthalten. Bei 400 Tupeln sind es hingegen maximal 200 verschiedene Werte (aufgrund der Domänenbegrenzung).

Statistische Methoden

Statistische Methoden (Histogramm-Verfahren; siehe [IOA1995]) sind den weiter oben beschriebenen theoretischen Methoden sehr ähnlich. Sie nutzen aber anstatt allgemeiner Annahmen über die Werteverteilung einer Relation statistische Informationen aus der Datenbank selbst. Ein theoretisches Verfahren wird bei 100 Tupeln für das Attribut A mit der aus der Basisdomäne INTEGER (siehe Seite 104) abgeleiteten Domäne [0;200] annehmen, dass jeder Wert der Domäne mit der Wahrscheinlichkeit 0,5 (100/200) auftritt.

Ein Histogramm-Verfahren "weiß" aber beispielsweise, dass nur 10 verschiedene Werte in A vorkommen, die Wahrscheinlichkeit also nur 0,05 beträgt. Voraussetzung dafür ist, dass das DBMS dem Verfahren die benötigten statistischen Informationen zur Verfügung stellt. Diese Informationen können vom DBMS entweder laufend (dynamischer Ansatz) oder auf Anfrage (statischer Ansatz) aktualisiert werden.

7.5.2.3 Das in der Lösung angewendete Verfahren

Zur Berechnung der Selektivität der Relationalen Operatoren wird eine Hybridform der theoretischen und statistischen Methode verwendet. Grundsätzlich wird die Selektivität auf Basis des theoretischen Ansatzes abgeleitet. Statistische Informationen werden nur insofern benutzt, wie sie sich - unabhängig von konkreten Daten - aus der Datenstruktur der jeweiligen Verfahren ableiten lassen. Ist letzteres möglich, wird die abgeleitete Selektivität aufgrund dieser Informationen angepasst.

Statistische Gleichverteilung

Wir nehmen für alle Basisdomänen eine statistische Gleichverteilung der Werte an. Das heißt, für alle Werte w einer Domäne D ist es gleich wahrscheinlich, dass sie auf einem Attribut angenommen werden, es sei denn, dieses Attribut besitzt Merkmale (zum Beispiel "nimmt in 80% aller Fälle durch 2 teilbare Werte an"), mit denen man diese Annahme verfeinern kann.

7.5.2.4 Selektivität der Operatoren im Überblick

Im Folgenden werden die im vorangegangenen Abschnitt 7.5.1 definierten Relationalen Operatoren mit den zugehörigen Selektivitäten in Form der Kardinalität der Ergebnisrelation E in tabellarischer Form wiedergegeben. Die Selektivität der Operatoren wurde weitgehend auf Basis von [MAN1988] und [DAD1996] abgeleitet.

Operator	**Kürzel**	**Kardinalität von E**	**Anmerkungen**
Projektion	PJ	|E| = |R|	
Selektion	SL	$\|E\| = \prod_{i=1\ldots p} \xi_i * \|R\|$	
p("A=α")		$\xi = \frac{1}{val_A(R)} * f$	mit ξ=1/val_A(R)*f und f (entfällt, wenn sich die Intervalle von α und A vollständig überdecken) wie folgt definiert: f=p(A,α)*p(α,A) mit $p(A,\alpha) = \begin{cases} \frac{\min(\max(A),\max(\alpha)) - \max(\min(A),\min(\alpha))}{\max(A) - \min(A)} & falls >= 0 \\ 0, sonst \end{cases}$ $p(\alpha,A) = \begin{cases} \frac{\min(\max(A),\max(\alpha)) - \max(\min(A),\min(\alpha))}{\max(\alpha) - \min(\alpha)} & falls >= 0 \\ 0, sonst \end{cases}$
Sortierung	OD	|E| = |R|	
Duplikat-elimination	DT	$\|E\| <= \min\left(\prod_{i=1\ldots n} val_{A[i]}(R), \|R\|\right)$	
Gleichverbund	JN	|E| = |R| * |S| *ξ	mit $\xi = \frac{1}{val_{A[i]}(R)} = \frac{1}{val_{B[j]}(S)}$ Wenn A[i] und B[j] die Verbundattribute sind.
Sonderfall 1:N		|E|=max(|R|,|S|)	
Vereinigung	UN	|E| = |R| + |S|	
Tupel einfügen	INS	|E| = |R| +1	
Tupel ändern	UPD	|E| = |R|	
Tupel löschen	DEL	|E| = |R| −1	

Tabelle 5: Übersicht der Selektivität der Relationalen Operatoren

7.5.2.5 Zeitkomplexität

Im Folgenden werden die im vorangegangenen Abschnitt 7.5.1 definierten Relationalen Operatoren mit den zugehörigen Zeitkomplexitäten in tabellarischer Form wiedergegeben. Die Zeitkomplexitäten der Operatoren wurden weitgehend aus den Komplexitäten der in [AHO1974] und [SED1995] analysierten Algorithmen abgeleitet. Grundsätzlich ist `n=|R|` und `m=|S|` anzunehmen.

Operator	**Kürzel**	**Komplexität**	**Anmerkungen**
Projektion	PJ	`O(n)`	
Selektion	SL	`O(n*p)`	Einfaches Durchsuchen bei `p` Prädikaten. Teure Prädikate (Semijoin, et cetera) sind besonders zu berücksichtigen.
(Sonderfall 1)		$O((\log_2 n)+m*p)$ bzw. O(m*p)	Wenn `F` genau `p` Prädikate enthält und mindestens eines der Attribute einen Wert höchstens `m` mal enthält, sowie ein Gleichheitsprädikat ("=") und einen Index besitzt; ist die Domäne des ausgezeichneten Attributes zusätzlich hash-geeignet, ergibt sich sogar `O(m*p)`. Teure Prädikate (Semijoin, et cetera) sind besonders zu berücksichtigen.
(Sonderfall 2)		$O((\log_2 n)+p)$ bzw. `O(p)`	Wenn alle Prädikate in `F` logisch mit demselben Primärschlüsselattribut (das ein Gleichheitsprädikat ("=") und einen Index besitzen muss) per Konjunktion (`AND`) verbunden sind; ist die Domäne dieses Attributes zusätzlich hash-geeignet, ergibt sich sogar `O(p)`. Teure Prädikate (Semijoin, et cetera) sind besonders zu berücksichtigen.
(Sonderfall 3)		$O\left(\sum_{i=1\ldots p}((\log_2 n)+(\log_2\lvert P_i\rvert)*\lvert P_i\rvert)\right)+O(p*MAX_{j=1\ldots p}\lvert P_j\rvert)$ = mit `p` Anzahl der Prädikate	Wenn alle Prädikate einen Index besitzen, konjunktiv verbunden (`AND`) sind und $\lvert P_i\rvert$ die Anzahl der Tupel darstellt, die Prädikat P_i erfüllen.
Sortierung	OD	$O(n*\log_2 n)$	Allgemeingültige Verfahren
(Sonderfall)		`O(n)`	Verfahren mit begrenzter Domäne
Duplikat-elimination	DT	$O(n*\log_2 n)$	
(Sonderfall 1)		`O(1)`	Falls die Relation `R` ein Primärschlüsselattribut enthält.
Gleichverbund	JN	$O(n*\log_2 n)+(m*\log_2 m)$	Falls die Verbundattribute nicht hash-geeignet sind und die Verbundattribute nicht sortiert vorliegen, was wir als Standardfall annehmen wollen. Als Algorithmus kann der Mischverbund eingesetzt werden.
(Sonderfall 1)		`O(n+m)`	Falls die Verbundattribute nicht hash-geeignet sind, aber bereits in

Operator	Kürzel	Komplexität	Anmerkungen
			sortierter Form vorliegen reduziert sich die Zeitkomplexität auf O(n+m). Als Algorithmus kann der Mischverbund eingesetzt werden.
(Sonderfall 2)		O(n+m)	Falls die Domäne eines der Verbundattribute A_i und B_j klein genug ist, um eine injektive Hashfunktion einzusetzen; Sortierung der Verbundattribute ist in diesem Falle nicht erforderlich.
Vereinigung	UN	O(n+m)	Für die Vereinigung der Relationen R und S, falls keine Sortierordnung aufrechterhalten werden muss.
Tupel einfügen	INS	O(1)	Falls keine Sortierordnung aufrechterhalten werden muss.
(Sonderfall)		$O\left(\log_2 \frac{(n+(m-1))!}{(n-1)!}\right)$	Mit n=|R| und m = Anzahl der einzufügenden Tupel. Falls eine Sortierordnung auf einem Attribut $A_{i1},\ldots,A_{in}$ der Relation R aufrechterhalten werden muss.
Tupel löschen	DEL	O(n*p)	Mit n=|R|, falls keine Sortierung auf einem der zu ändernden Attribute aufrechterhalten werden muss.
(Sonderfall)		$O\left(\log_2 \frac{n!}{(n-m)!}\right)+O(n*p)$	Mit n=|R| und m = Anzahl der zu löschenden Tupel. Falls eine Sortierordnung auf einem Attribut $A_{i1},\ldots,A_{in}$ der Relation R aufrechterhalten werden muss.
Tupel ändern	UPD	$O(SL_FR)$	Falls keine Sortierung auf einem der zu ändernden Attribute aufrechterhalten werden muss.
(Sonderfall)		$O(2*m*\log_2 n)+O(SL_pR)$	Mit n=|R| und T = Anzahl der zu ändernden Tupel. Falls eine Sortierordnung auf einem Attribut $A_{i1},\ldots,A_{in}$ der Relation R aufrechterhalten werden muss.

Tabelle 6: Übersicht der Zeitkomplexität der Relationalen Operatoren

7.6 Anfragen-Verarbeitung in DBMS

Die Anfragenverarbeitung eines DBMS lässt sich in zwei wesentliche Phasen unterteilen. Die erste, die Vorbereitungsphase genannt werden soll, erstreckt sich von der Anwenderanfrage über die Transformation der SQL-Anfrage in Relationale Operatoren bis hin zur Ermittlung des optimalen Operatorbaumes.

Die zweite Phase soll Ausführungsphase genannt werden. Sie bringt dann die zuvor ermittelte optimale Operatorreihenfolge zur Ausführung und übergibt das Ergebnis der Anfrage an den Anwender.

Eine detaillierte Aufteilung findet sich in der folgenden Abbildung. Dabei wird die Vorbereitungsphase in sechs und die Ausführungsphase in vier Teilschritte zerlegt.

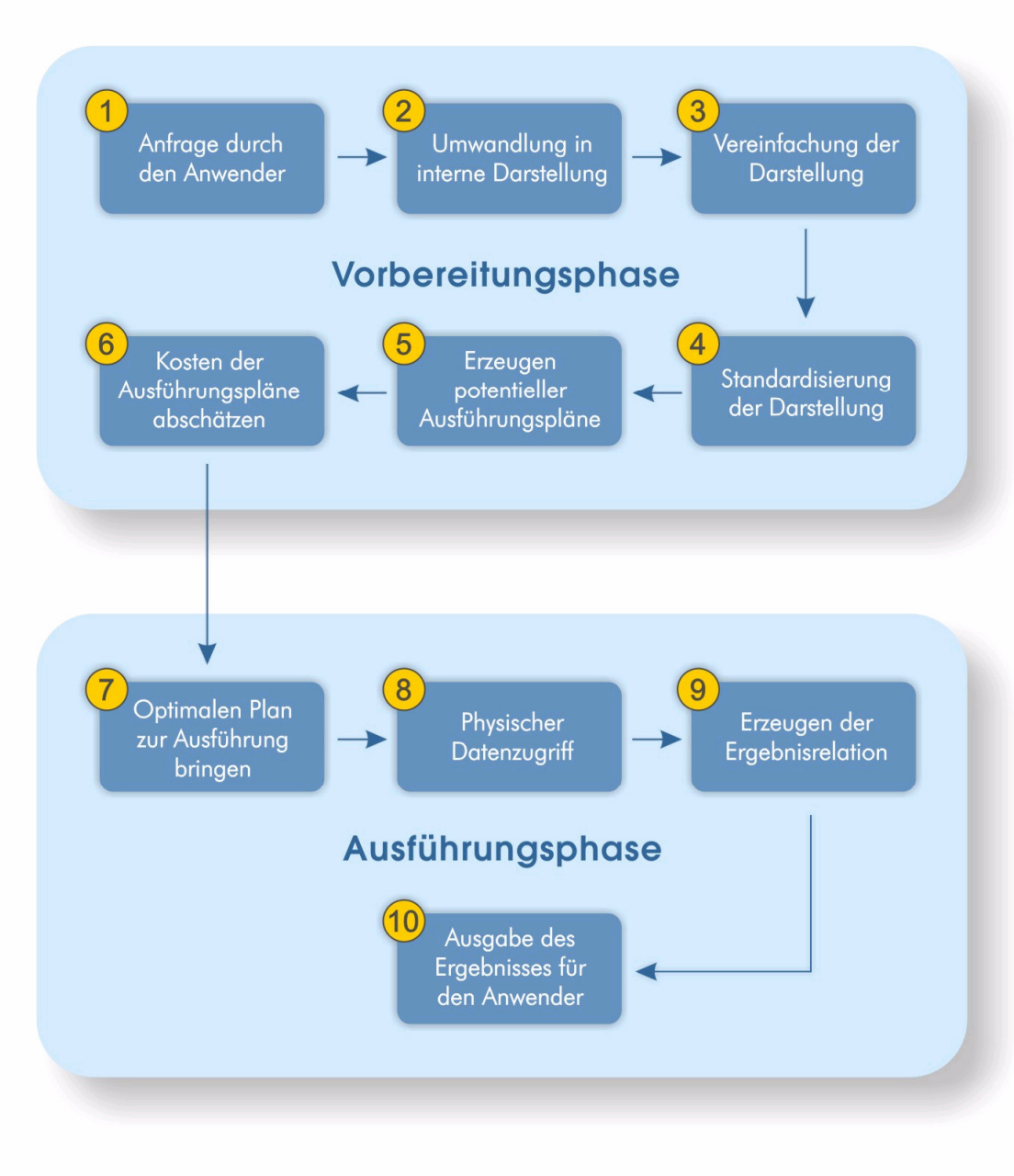


Abbildung 100: Die in zwei wesentlichen Phasen geteilten 10 Schritte der Anfragenausführung

7.6.1 Die Vorbereitungsphase

Diese Phase erstreckt sich von der Anwenderanfrage über die Transformation der SQL-Anfrage in Relationale Operatoren bis hin zur Ermittlung des optimalen Operatorbaumes. Es werden alle Aufgaben vor der eigentlichen Ausführung der Anfrage erledigt. Die zu dieser Phase gehörenden Teilschritte werden im Folgenden kurz erläutert. Zuvor soll aber noch die Zeitkomplexität der Vorbereitungsphase betrachtet werden.

7.6.1.1 Zeitkomplexität der Vorbereitungsphase

Bei komplexen Anfragen spielt auch die Laufzeit der Vorbereitungsphase selbst eine Rolle. Im Rahmen des zu analysierenden Verfahrens ist dies in der finalen Version allerdings nicht der Fall. Frühere Versionen besaßen komplexere Leseanfragen, so dass die Vorbereitungsphase nicht irrelevant war. Daher wird die Laufzeitkomplexität der Vorbereitungsphase ebenfalls kurz dargestellt.

Eine SQL-Anfrage enthält potenziell eine Vielzahl von Operatoren. In welcher Reihenfolge diese Operatoren zur Verarbeitung kommen, ist im wesentlichen Aufgabe des so genannten Optimizers. Als Ergebnis liefert dieser einen detaillierten *Ausführungsplan*, anhand dessen die weitere Verarbeitung erfolgt.

Zur objektiven Bewertung soll der optimale Ausführungsplan zugrunde gelegt werden. Es muss also die Arbeitsweise moderner Optimizer simuliert werden, um den zugrunde optimalen Ausführungsplan bei gegebener SQL-Anfrage zu ermitteln. Daher sind auch die im Rahmen dieser "Simulation" anfallenden Laufzeitkosten für die Vorbereitungsphase zu veranschlagen.

Der dominierende Summand bei der Laufzeitkomplexität der Vorbereitungsphase ist die Ermittlung der optimalen Ausführungsreihenfolge der einzelnen Operatoren (Abschnitt 7.6.1.7 auf Seite 139 ff.). Gleichzeitig ist dies der am besten abschätzbare Teil der Anfrageverarbeitung, da sich hier standardisierte Verfahren etabliert haben. Wir greifen an dieser Stelle bereits etwas vor und geben hier die Zeitkomplexität der Optimierung des Operatorbaumes an:

Zeitkomplexität zur Bildung der optimalen Verbundreihenfolge

Gehört zur Klasse

$$O\left(\left(\left|Ops_v\right| * 2^{\left|Ops_v\right|-1}\right) * C_{OPT}\right)$$

mit $|Ops_V|$=Anzahl der Verbundoperationen in der Anfrage und C_{OPT}=Kosten zur Bestimmung der Zeitkomplexität und Selektivität

Zeitkomplexität zur Bildung der optimalen Reihenfolge der sonstigen Operatoren

Gehört zur Klasse

$$O\left(2 * \log_2\left|Ops_S\right| * \left|Ops_S\right|\right)$$

mit $|Ops_S|$=Anzahl der sonstigen Operatoren. Eine detailliertere Ausführung dazu finden sich in Abschnitt 7.6.1.7 auf Seite 139 ff..

Durch die oben angeführten Zeitkomplexitäten werden die Kosten des Optimizers abgeschätzt. Die übrigen Laufzeitkosten der Vorbereitungsphase sollen unter der Konstanten $C_{BASISKOSTEN}$ zusammengefasst werden. Diese wird zur Approximation der realen Laufzeit mit der Gesamtzahl der Operatoren der Anfrage multipliziert, da sie offensichtlich von diesen abhängig ist[41]:

$$O\left(C_{BASISKOSTEN} * \left(\left|Ops_s\right| + \left|Ops_v\right|\right)\right)$$

mit $|Ops_S|$=Anzahl der sonstigen Operatoren und $|Ops_V|$=Anzahl der Verbundoperationen.

[41] Wie teuer diese Anfragevereinfachung wird, hängt stark vom betrachteten DBMS ab. Im Gegensatz zur "reinen" Optimierung finden sich in der Literatur keine Standardalgorithmen für diese Aufgabe. Im Rahmen dieser Arbeit treten zwar nur Anfragen auf, die nicht mehr vereinfacht werden können, die Kosten für die Analyse zur Anfrageverarbeitung müssen dennoch berücksichtigt werden.

Daraus ergibt sich folgende Gesamtzeitkomplexität pro SQL-Anfrage:

$$O\left(\left(\left|Ops_v\right| * 2^{\left|Ops_v\right|-1}\right) * C_{OPT} + \left(2 * \log_2\left|Ops_s\right| * \left|Ops_s\right|\right) * C_{OPT} + C_{BASISKOSTEN} * \left(\left|Ops_s\right| * \left|Ops_v\right|\right)\right)$$

7.6.1.2 Anfrage in SQL (Schritt 1)

Es wird postuliert, dass alle Anfragen in SQL (einen guten Überblick geben [LAD1997] und [FRE1998]) gestellt werden.

7.6.1.3 Transformation zu Relationaler Algebra (Schritt 2)

Die im zweiten Schritt vorgenommene Umwandlung in eine interne Darstellung kann zur Vereinfachung als Übersetzung in Relationen Algebra interpretiert werden. Letztere besitzt prozeduralen Charakter und ist daher für die weitere Verarbeitung gut geeignet. Daher wird zunächst eine Transformation gegebener SQL-Anweisungen in Relationen Algebra durchgeführt.

Zeitkomplexität für die Transformation zu Relationaler Algebra

Die Zeitkomplexität wird wie folgt abgeschätzt: $O(C_{BASISKOSTEN(A)})$. Wie oben beschrieben, wird sie unter der Konstante $C_{BASISKOSTEN}$ subsumiert.

7.6.1.4 Transformation zu Normalform (Schritt 3)

Die Standardisierung im dritten Schritt wird durch Umformung in eine Normalform erreicht. Dabei sind zwei Abstraktionsebenen innerhalb der Anfrage zu unterscheiden.

Die logischen (booleschen) Operatoren im WHERE-Teil einer SQL-Anweisung lassen sich in die konjunktive oder die disjunktive Normalform bringen.

Als Beispiel soll der logische Ausdruck `A=10 AND B>16 OR (C=3 AND (D>77 OR E=15))` dienen.

In konjunktiver Normalform (KNF), die durch eine Konjunktion von Disjunktionsprädikaten gekennzeichnet ist, würde dieser Ausdruck wie folgt lauten: `(A=10) AND (B>16 OR C=3) AND (B>16 OR D>77 OR E=15)`

In disjunktiver Normalform, die durch eine Disjunktion von Konjunktionsprädikaten gekennzeichnet ist, würde dieser Ausdruck wie folgt lauten: `(A=10 AND B>16) OR (C=3 AND D>77) OR (C=3 AND E=15)`

Wesentlich komplizierter ist die Standardisierung auf der übergelagerten Abstraktionsebene der Anfragestruktur. Hier geht es um eine Normalform für den Aufbau der Anfrage mit potenziellen Unterabfragen, et cetera. Die meisten DBMS führen diese Standardisierung aber in direktem Zusammenhang mit der Optimierung durch. Wir schließen uns dieser Vorgehensweise an und führen dies daher hier nicht näher aus.

7.6.1.5 Vereinfachung der Anfragedarstellung (Schritt 4)

In diesem Schritt geht es vorrangig um die Elimination von redundanten Prädikaten. Letztere können beispielsweise durch *Views* oder *ineffiziente Anfragen* von Anwenderseite entstehen. Aus

```
SELECT * FROM A,B WHERE A.PK=B.FK AND A.Beispiel<10 AND A.Beispiel<20
```

kann durch Entfernung des redundanten Prädikates folgende semantisch äquivalente Abfrage erzeugt werden:

```
SELECT * FROM A,B WHERE A.PK=B.FK AND A.Beispiel<10
```

Zeitkomplexität zur Vereinfachung der Anfragedarstellung

Die Zeitkomplexität wird wie folgt abgeschätzt: $O(C_{BASISKOSTEN(c)})$. Wie oben beschrieben, wird sie unter der Konstante $C_{BASISKOSTEN}$ subsumiert.

7.6.1.6 Erzeugen des Operatorbaums (Schritt 5)

Ein Operatorbaum entsteht durch Transformation einer SQL-Anfrage in eine Sequenz Relationaler Operatoren. Aus der Abfrage (bei der die Attribute `R.A,R.B,R.C,S.M,S.N` und `T.X` der Ausgangsrelationen `R,S` und `T` in der Relation, auf welcher der PJ-Operator aufsetzt, die Komponenten `k1,k2,k3,k4,k5` und `k6` bilden)

```
SELECT R.A,R.B,R.C,S.M,S.N,T.X
FROM R,S,T
WHERE R.A=S.M AND R.B=T.X
AND R.C=199 AND S.N<100
```

wird dadurch

$PJ_{k1,k2,k3,k4,k5,k6}(SL_{R.C=199}(SL_{S.N<100}(JN_{R.B=T.X}(JN_{R.A=S.M}(R,S),T))))$

was dem folgenden Operatorbaum entspricht:

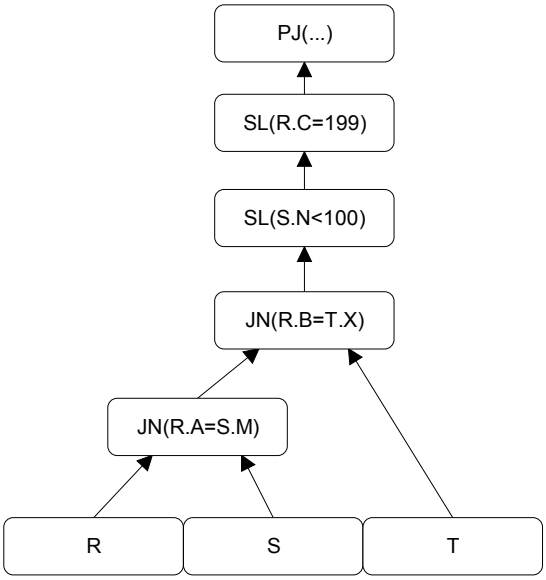


Abbildung 101: Ein aus einer SQL-Abfrage resultierender Operatorbaum

Dabei werden gleichzeitig die im Folgenden beschriebenen Anpassungen vorgenommen, die auf den folgenden zulässigen Äquivalenzumformungen basieren [CHA1995], [DAD1996]. Bevor auf die Äquivalenzumformungen eingegangen wird, soll noch eine kompakte Übersicht der in Abschnitt 7.5 eingeführten Relationalen Operatoren gegeben werden.

Übersicht der Relationalen Operatoren und der Umformungen darauf

Bezeichnung	**SQL-Syntax**	**Relationaler Operator**
Projektion	SELECT	$PJ_L R$
Selektion	`WHERE ...[F]...`	$SL_F R$
Sortierung	`ORDER`	$OD\ R_{Attr}$
Duplikatelimination	`DISTINCT`	`DT R`
Gleichverbund	`WHERE ...` $A_i=B_j$`...`	$JN_{Ai=Bj}(R,S)$

Tabelle 7: Übersicht der Relationalen Operatoren

Auf diesen Operatoren sind unter anderem die im Folgenden aufgeführten Äquivalenzumformungen möglich:

Kommutative Operatoren (binäre)

$JN_{Ai=Bj}(R,S) \Leftrightarrow S\ JN_{Bj=Ai}(S,R)$

Kommutative Operatoren (unäre)

$SL_{F1}(SL_{F2}R) \Leftrightarrow SL_{F2}(SL_{F1}R)$

$PJ_{L1}(PJ_{L2}R) \Leftrightarrow PJ_{L2}(PJ_{L1}R)$; *unter der Annahme, dass bei geschachteltem PJ das "äußere"* L *neu indiziert wird, so dass die gleichen Attribute angesprochen werden, wie vor der Anwendung des "inneren"* PJ.

$SL_F(PJ_LR) \Leftrightarrow PJ_L(SL_FR)$; *Selektionsattribute müssen erhalten bleiben (in* L *vorkommen)*

Assoziative Operatoren

$JN_{Bj=Ck}(JN_{Ai=Bj}(R,S),T) \Leftrightarrow JN_{Ai=Bj}\ (R,(J_{NBj=Ck}(S,T))$; *Verbund-Attribute müssen "passen"*

Distributive Operatoren

$SL_F(JN_{Ai=Bj}(R,S) \Leftrightarrow JN_{Ai=Bj}(SL_FR,SL_FS)$; *Selektionszerlegung muss "passen"*

$PJ_L(JN_{Ai=Bj}(R,S) \Leftrightarrow JN_{Ai=Bj}(PJ_LR\ ,PJ_LS)$; *Projektionszerlegung muss "passen"*

Zusammenfassung von Operatoren

$PJ_{L1}R,\ PJ_{L2}R \Leftrightarrow PJ_{L1\&L2}R$; *wobei "&" für die Zusammenfassung der Listen* L1 *und* L2 *steht*

$SL_{F1}R,\ SL_{F2}R \Leftrightarrow SL_{(F1,F2)}R$

Überführe n-äre Verbünde in n-1 binäre Verbünde

Die Ausführung des Verbundoperators ist assoziativ und kommutativ. Daraus folgt, dass sich ein n-ärer Verbund in eine Sequenz von n-1 binären Verbünden zerlegen lässt.

$JN_{Bj=Ck}(JN_{Ai=Bj}(R,S),T) \Leftrightarrow JN_{Ai=Bj}(R,S),\ JN_{Bj=Ck}(S,T)$

Zerlege Selektionen mit p primitiven Prädikaten in p Selektionen mit einem primitiven Prädikat

Eine Selektion mit einer aus p primitiven Prädikaten bestehenden Selektionsfunktion kann in p Selektionen mit jeweils einem primitiven Prädikat transformiert werden:

$SL_{(p1\ AND\ p2)}R \Leftrightarrow SL_{p1}(SL_{p2}R)$

7.6.1.7 Optimieren des Operatorbaums (Schritt 5 und 6)[42]

Die Optimierung eines Operatorbaums basiert im Wesentlichen auf der Veränderung der Ausführungsreihenfolge der enthaltenen Relationalen Operatoren [LOC1987], [SCH1995]. Weitere Optimierungstechniken sollen nur vorgestellt werden, soweit sie für diese Arbeit von Bedeutung sind.

Ausführungsreihenfolge der Operatoren optimieren

Die Bestimmung der optimalen Ausführungsreihenfolge der Operatoren ist die wesentliche Aufgabe des Optimizers. Sie basiert auf den Abschätzungen zur Selektivität und der Zeitkomplexität (siehe Abschnitt 7.5 auf Seite 109 ff.). Grob vereinfacht ausgedrückt werden für alle Operatoren aufgrund der betrachteten Relation (dabei handelt es sich um die Basisrelation oder die Ergebnisrelation des vorausgegangenen Operators) die Selektivität und Zeitkomplexität ermittelt. Es wird dann der Operator mit maximaler Selektivität und minimaler Zeitkomplexität in der Ausführungsreihenfolge an die nächste Stelle gesetzt. Auf diese Weise "wächst" der Operatorbaum quasi von den Blättern zur Wurzel.

[42] Einen sehr guten Überblick über dieses Themengebiet gibt [CHEN1998]. Interessante Aspekte werden in [SIM1996] (Sortierordnung und Optimierung), [POU1996] (Algebraische Optimierung), [CHA1995] (Optimierung bei Aggregatfunktionen) und [HEL1993] (Berücksichtigung teurer Prädikate in Selektionen) behandelt.

Dabei kann aufgrund der Separation von Verbünden und anderen Operationen zwischen der Optimierung der Verbundreihenfolge und der Optimierung der Reihenfolge der übrigen Operationen unterschieden werden.[43]

Zeitkomplexität zur Bildung der optimalen Verbundreihenfolge

Gehört zur Klasse

$$O\left(\left(\left|Ops_v\right| * 2^{|Ops_v|-1}\right) * C_{OPT}\right)$$

mit $|Ops_v|$=Anzahl der Verbundoperationen in der Anfrage und C_{OPT}=Kosten zur Bestimmung der Zeitkomplexität und Selektivität

Für die Bestimmung der optimalen Ausführungsreihenfolge der Verbundoperatoren benötigt der Optimizer selbst natürlich auch eine bestimmte Laufzeit. Der Zeitaufwand zur Berechnung der optimalen Reihenfolge der Operationen mit einem vollständigen Enumerationsverfahren[44] beträgt bei einem Verbund von n Relationen bereits $O(n!)$.

Bei einem Verbund über acht Relationen entspricht dies bereits 40.320 Operationen (zur Berechnung der Kosten), wenn alle Permutationen geprüft werden sollen.

Durch dynamische Programmierung kann die Laufzeitkomplexität zwar auf $O(n*2^{n-1})$ reduziert werden[45] (im Beispiel 1024 Operationen), der Optimizer darf aber natürlich auch dann bei den Kostenbetrachtungen nicht unberücksichtigt bleiben. Dies insbesondere deshalb nicht, weil ad-hoc SQL-Anfragen - wie sie in der Praxis meist zum Einsatz kommen - eine Optimierung für jede SQL-Anfrage erfordern[46].

Werden SQL-Anfragen in Schleifen ausgeführt, können sich die Optimizer-Kosten zu einem wesentlichen Faktor entwickeln.

Zeitkomplexität zur Bildung der optimalen Reihenfolge der sonstigen Operatoren

Gehört zur Klasse

$$O\left(\left(2 * \log_2\left|Ops_s\right| * \left|Ops_s\right|\right) * C_{OPT}\right)$$

mit $|Ops_N|$=Anzahl der sonstigen Operatoren und C_{OPT}=Kosten zur Bestimmung der Zeitkomplexität und Selektivität

Für die Bestimmung der optimalen Ausführungsreihenfolge der sonstigen Operatoren (im Rahmen dieser Arbeit handelt es sich hier um Selektionen und Projektionen) sind ebenfalls vorrangig die Zeitkosten von Interesse. Unter Berücksichtigung der Push-Regel kann vereinfachend eine Sortierung der sonstigen Operatoren nach Selektivität und Zeitkomplexität angenommen werden.

Damit ergibt sich die oben angeführte Laufzeitkomplexität zur Optimierung der Reihenfolge der sonstigen Operatoren.

---

[43] Diese Vereinfachung wird für die Abschätzung der Zeitkomplexität der Optimierung einer SQL-Anfrage eingeführt.

[44] Dabei werden alle möglichen Reihenfolgen der Verbünde betrachtet. Ein Verfahren, das offensichtlich bei einer großen Anzahl von Verbünden sehr teuer wird.

[45] In [CHEN1998] (Seite 35) und [SES1994] (Seite 439) wird die Laufzeit für die Bildung der optimalen Verbundreihenfolge mit $O(n*2^{n-1})$ angegeben. In [MIT1995] (Seite 260) findet sich eine Abschätzung von $O(2^n)$. Alle Quellen geben die zugrunde liegenden Algorithmen nur ansatzweise wieder. Aufgrund eines eigenen Algorithmus kommt der Autor zu einem Ergebnis, das nahe an [CHEN1998] und [SES1994] liegt. Da offensichtlich $O(2^n) < O(n*2^{n-1})$ für $n>2$ gilt, wird im Folgenden diese Abschätzung verwendet.

[46] Statische SQL-Anfragen, die immer wieder ausgeführt werden, können vorverarbeitet (precompiled) werden. In diesem Falle ist bei der eigentlichen Ausführung dann keine Analyse mehr notwendig, da bereits der komplette Ausführungsplan vorliegt. Damit diese vorverarbeiteten SQL-Anfragen dauerhaft optimal bleiben, müssen diese allerdings regelmäßig an die sich verändernden statistischen Daten angepasst werden. Im Rahmen dieser Arbeit werden grundsätzliche alle Anfragen als nicht-vorverarbeitet angenommen.

Selektionen auf identischen Relationen zusammenfassen

Mehrere Selektionen auf derselben Relation werden zu einer Selektion zusammengefasst. Hier einige Beispiele:

$SL_{R.A=10}(SL_{R.C<100}R) \Leftrightarrow SL_{(R.A=10\ AND\ R.C<100)}$

$SL_{R.C=77}(JN_{R.A=S.M}(SL_{R.B=8}R, SL_{S.N=10}S)$

$\Leftrightarrow (JN_{R.A=S.M}(SL_{R.C=77}(SL_{R.B=8\ AND}\ R), SL_{S.N=10}S))$

$\Leftrightarrow (JN_{R.A=S.M}(SL_{(R.B=8\ AND\ R.C=77)}R), SL_{S.N=10}S))$

Selektionen zu den Blättern verschieben

Diese "Push"-Regel soll dafür sorgen, dass die meist kostengünstigen und stark selektiven ($|SL_FR|$ deutlich kleiner als $|R|$) Selektionen frühzeitig ausgeführt werden, um die Kardinalität der Relationen, auf denen kostenintensive Operatoren (wie zum Beispiel der Verbund) ausgeführt werden müssen, so klein wie möglich zu halten.

Diese Regel darf allerdings nicht "blind" angewendet werden. Oft sind Selektionen selbst sehr kostenintensiv – eventuell sogar kostenintensiver als Verbünde. In [HEL1993] wird ein Verfahren dargestellt mit dem eine Sequenz von Verbünden und Selektionen ermittelt werden kann, die tatsächlich kostenminimal ist. Im Rahmen dieser Arbeit treten allerdings keine derart kostenintensiven Selektionen (Prädikate) auf, daher soll an dieser Stelle nicht weiter darauf eingegangen werden.

Optimieren der Auswertung der Prädikate einer Selektion

Innerhalb einer Selektionsoperation können sowohl sehr einfache, als auch sehr aufwendige Prädikate vorkommen. Der Definition "rank" aus [HEL1993] folgend, kann eine kostenoptimale Sequenz ermittelt werden, indem man alle Prädikate nach aufsteigender `Rank`-Reihenfolge evaluiert (siehe auch 7.5.2.1). Die Selektivität der einzelnen Relationalen Operatoren wurde weiter oben behandelt (siehe Seite 132 ff.). Die Kosten dieser `Rank`-Bildung wurden bereits ab Seite 139 ff. berücksichtigt.

7.6.2 Ausführungsphase

In den Schritten 7, 8, 9 und 10 der Anfrageverarbeitung wird der zuvor ermittelte Operatorbaum zur Ausführung gebracht und das Ergebnis an den Anwender übergeben. Die Laufzeitkomplexität dieser Schritte kann durch das eingeführte Transformationsverfahren abgeschätzt werden.

7.7 Das Transformationsverfahren

Im Folgenden wird das angewendete Verfahren zur Transformation zu skalarer Zeitkomplexität vorgestellt. Es beruht im Wesentlichen auf folgenden Schritten:

1. Erstellen der nötigen SQL-Anfragen für die Basisoperationen (siehe Seite 108)
2. Transformation der SQL-Anfrage in Relationale Algebra (siehe Seite 137)
3. Optimieren des Operatorbaums (siehe Seite 139)
4. Abschätzen der Kosten des Operatorbaums
5. Zusammenfassen der Transformation

8 Das theoretische Laufzeitverhalten

Die theoretische Laufzeitkomplexität des Verfahrens ist bedeutsam, um abschätzen zu können, wie sich die Performance bei großer und insbesondere wachsender Textbasis C entwickelt. Es werden dazu die Ergebnisse aus "7. Laufzeitkomplexität" verwendet.

8.1 Aufbau und Wartung der Datenstrukturen

Im Folgenden soll die Laufzeitkomplexität zum Aufbau und Aktualisieren der in 6.1.1.10 beschriebenen Datenstrukturen abgeleitet werden.

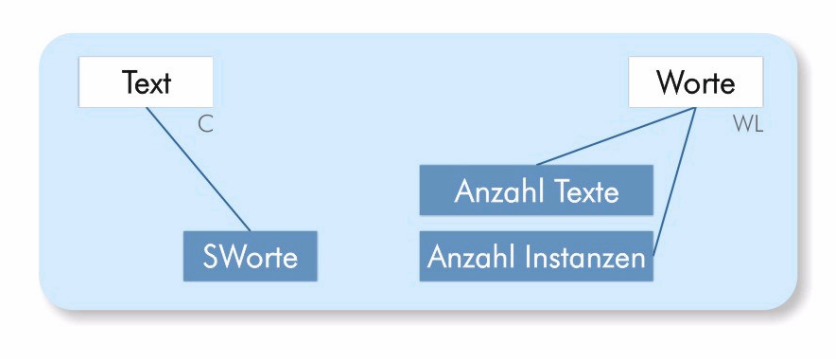


Abbildung 102:Die CRIC Daten müssen bei neuen, geänderten und gelöschten Texten adaptiert werden

Die Datenstrukturen WL und C müssen aktualisiert werden, wenn

(i) Ein neuer Text T eingefügt wird

(ii) Ein bestehender Text T entfernt wird

(iii) Ein bestehender Text T verändert wird

Die Wortliste WL enthält alle Worte, die in den Texten T der Textbasis C vorkommen können.

8.1.1 Neuer Text

8.1.1.1 Neuer Text in C

Das Einfügen von Text T in Relation C (Attribut „volltext“ der Tabelle „text“) erfolgt durch ein INSERT mit Erhalt des Index. Dabei wird das Primärschlüsselattribut text.id durch ein AUTO_INCREMENT seitens des DBMS mit Kosten O(1) gesetzt:

```
      INSERT INTO text (volltext)
      VALUES (T)
=     INS(text,T)
```

Damit ergibt sich mit 7.5.1.7 die Laufzeitkomplexität

$$O(\log_2|C|) \qquad (1)$$

8.1.1.2 Textanalyse

Nachdem ein neuer Text T eingefügt wurde, wird dieser zunächst in Worte zerlegt:

$O(|T|)$ mit $|T|$=Anzahl Zeichen in T (2)

Jedes Wort wird in WL aufgenommen. Dabei wird auch gespeichert, in wie vielen Texten das Wort vorkommt (Inkrementieren/Dekrementieren), um eine Basis für die dynamische Stoppwortbewertung und die Wortgewichtung (siehe 6.1.1.5und 6.1.1.6) im Text zu haben.

8.1.1.3 Wortliste WL

Ein Text hat im Mittel |T|/len Worte mit len=durchschnittlichen Länge eines Wortes. Daraus folgt, dass für einen Text T auf WL (Tabelle „worte“) im Mittel |T|/len Operationen (wenn man die Verarbeitung eines Wortes als *Operation* bezeichnet) erforderlich sind. Ein Wort W wird in WL nur einmal gespeichert. Die Operationen auf WL beschränken sich danach auf das Aktualisieren der Werte $N_c(W)$ („Anzahl Instanzen“) und $I_c(W)$ („Anzahl Texte“) des Wortes. Aufgrund der Stoppwortliste SWL wird die Operation auf WL in der Regel (Zipf´s Gesetz [ZIP1949]) sogar deutlich weniger als |T|/len mal ausgeführt. Wir verwenden als worst-case aber diese obere Grenze.

Für jedes Wort W aus T (also |T|/len mal) muss mit $N_T(W)$ = „Instanzen des Wortes W im Text T“ Folgendes ausgeführt werden:

```
UPDATE worte
SET   anzahl_instanzen=anzahl_instanzen+N_T(W),
      anzahl_texte=anzahl_texte+1
WHERE wort=W
```

Existiert das Wort W noch nicht in WL (Tabelle „worte“), so gibt UPDATE einen Fehler zurück. Dann muss zusätzlich Folgendes ausgeführt werden:

```
INSERT INTO worte (wort,anzahl_instanzen,anzahl_texte)
VALUES (W,N_T(W),1)
```

Da im worst-case alle Worte eines Textes T noch nicht in WL enthalten sind, müssen für die O-Notation immer beide Befehle berücksichtigt werden. In Relationenalgebra dargestellt gilt also:

(i) $UPD_{wort=W}$(anzahl_instanzen=anzahl_instanzen+$N_T(W)$, anzahl_texte=anzahl_texte+1)

(ii) INS(worte,(wort,anzahl_instanzen,anzahl_texte))

Diese Operationen werden auf WL dann |T|/len mal (für jedes Wort aus T) ausgeführt. Damit folgt für die Laufzeitkomplexität (die Selektivität spielt keine Rolle, da beim UPD die Kardinalität der Ergebnisrelation identisch zur Eingangsrelation ist):

$$(|T|/len)*[O(\log_2|WL|)] + (|T|/len)*[O(\log_2|WL|)] \qquad (3)$$

Denn es müssen |T|/len Datensätze in WL unter Erhaltung der Indexstruktur aktualisiert (siehe 7.5.1.9) und ein neuer Datensatz unter Erhalt der Indexstruktur eingefügt (siehe 7.5.1.7) werden:

UPDATE: $O(2*1*\log_2 n)+O(n*p)$

INSERT: $O\left(\log_2\left(\frac{(n)!}{(n-1)!}\right)\right)=O(\log_2 n)$

8.1.1.4 CRIC Gesamtlaufzeit

Insgesamt ergibt sich:

$$O(\log_2|C|) \quad (1)$$
$$+O(|T|) \quad (2)$$
$$+2*(|T|/len)*O(\log_2|WL|) \quad (3)$$

mit $|T|$=Anzahl Zeichen in T. Da $|T|$ nicht mit der Zeitachse wächst und begrenzt ist (maximale Textlänge) kann es als Konstante betrachtet werden, was folgende Vereinfachung ermöglicht:

$$O(\log_2|C|)$$
$$+2*(\log_2|WL|)$$

$$= \quad O(\log_2|C|)$$
$$+O(\log_2|WL|)$$

$$= \quad O(\log_2|C|+\log_2|WL|)$$
$$= \quad O(\log_2(|C|*|WL|)) \quad (4)$$

8.1.1.5 Kosten des Volltext-Index des Datenbanksystems

Beim Einfügen eines neuen Textes müssen nicht nur die CRIC Datenstrukturen, sondern auch der Volltext-Index des Datenbanksystems aktualisiert werden. Das Datenbanksystem (*MySQL*) hält Strukturen analog zu `WL` vor, verwendet aber als vollständigen inversen Index auch die Wort-Text-Relation `WT`.

Wort-Text Relation WT

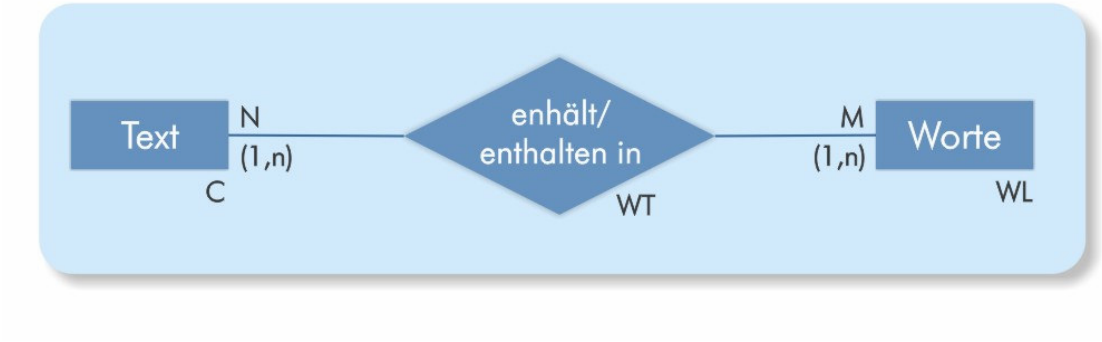


Abbildung 103:Das DBMS nutzt einen vollständigen inversen Index

Da alle Worte bei der Textanalyse "gesammelt" und en bloc in die Wort-Text Relation `WT` übertragen werden und dort das gleiche Wort mehrfach (in unterschiedlichen) Texten vorkommen kann (kein Primärschlüsselattribut) gilt

$$(|T|/len)*O(\log_2((|T|/len)+2*|WT|)/2) \quad (5)$$

Laufzeit

Technisch sind diese Datenstrukturen durch einen doppelten (kaskadierten) B-Tree und nicht auf Basis von Relationen implementiert. Ferner wird als Gewicht eines Wortes nur der lokale Anteil (TF) des TF-IDF-Derivates gespeichert. Der Grund dafür ist die Tatsache, dass das Datenbanksystem die Wortgewichtungen für alle Texte sonst laufend neu berechnen müsste, da sich mit jedem neuen, geänderten oder entferntem Text der globale Anteil (IDF) aller Gewichtungen aller Texte ändert. Und ein Neuberechnen aller Gewichtungen in Echtzeit wäre zu langsam. Der globale Anteil wird in Form der $I_C(W)$ in einer Primärspeicherstruktur vorgehalten und dann beim Ermitteln der "echten" Wortgewichte mit dem lokalen Wortgewicht verbunden.

Die Laufzeitkomplexität ergibt sich allerdings bis auf den Anteil von WT analog zu CRIC [GOL2003],[ZAW2004]. Da die feste Stoppwortliste des DBMS deaktiviert wurde, stimmt WL zwischen CRIC und dem inversen Index des DBMS überein. Daher folgt mit (4) und (5):

$$
\begin{aligned}
& O(\log_2(|C|*|WL|) \\
& +(|T|/len)*O(\log_2((|T|/len)+2*|WT|)/2) \\
= \quad & O(\log_2(|C|*|WL|) \\
& +O(\log_2(|WT|)) \\
= \quad & O(\log_2(|C|*|WL|*|WT|) \qquad (6)
\end{aligned}
$$

8.1.1.6 Suche der verwandten Texte

Zur Ermittlung der verwandten Texte werden die n Worte mit dem höchsten Gewicht verwendet. Diese werden im Rahmen der Textanalyse bestimmt. Diese n Worte werden dann als Suchanfrage A für die Volltextsuche auf dem Datenbanksystem genutzt.

Die Volltextsuche macht sich bei der Bearbeitung der Suchanfrage A den erweiterten inversen Index zu nutze, der neben der Wort-Text-Relation auch die Gewichtung des Wortes in einem Text vorhält. Es wird für alle Texte, die die Suchworte enthalten die Summe der Gewichtungen der Worte gebildet und daraus die Confidence abgeleitet.

Anfrageverarbeitung

Der inverse Index liefert zunächst für eine Anfrage A mit den Worten $a_1,\ldots,a_n$ die lokalen Wort-Gewichtungen des Wortes a_i in den m_{ai} Texten $T_{ai,1},\ldots,T_{ai,mai}$ in denen a_i vorkommt. Dazu muss für jedes Wort a_i aus A eine binäre Suche auf der Wort-Text Relation WT durchgeführt werden, was in $O(\log_2|WT|)$ erfolgt. Anschließend müssen (auf dem B-Tree) die Gewichtungen aller Texte m die a_i enthalten gelesen werden: $O(m_{ai})$ für Wort a_i. Es ergibt sich also für die Selektion aller Worte $a_{i=1\ldots n}$ der Suchanfrage aus dem inversen Index:

$$O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}\, m_{ai})$$

Dabei ist m_{ai} auf $|C|/2$ begrenzt, da das Datenbanksystem Worte, die in mehr als 50 Prozent aller Texte vorkommen, als Stoppworte behandelt und nicht in den inversen Index aufnimmt:

$$O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}\, |C|/2)$$

Dann müssen alle Texte (nicht mehr beschränkt, da bereits zwei Suchworte durch überdeckungsfreies Vorkommen alle Texte aus C adressieren können) nach der Confidence sortiert werden:

$$O(|C|*\log_2|C|)$$

Insgesamt ergibt sich eine Laufzeitkomplexität von:

$$O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}\, |C|/2)+O(|C|*\log_2|C|) \qquad (7)$$

mit n=Anzahl der Worte der Anfrage

Da die Volltextsuche nicht in Sequenz mit anderen Relationalen Operatoren zum Einsatz kommt, kann auf eine Selektivitätsabschätzung verzichtet werden.

Speichern in der CRIC Distanz-Matrix

Das Speichern der verwandten Texte in der CRIC "Text-Text-Matrix" ist bezüglich der Laufzeit irrelevant, da eine konstante Anzahl an verwandten Texten (in der Evaluation "15") gespeichert werden. Eine größere Anzahl macht nur dann Sinn, wenn durch Zugriffsbeschränkung nicht alle vorberechneten verwandten Texte für alle Benutzer sichtbar sein dürften. Andernfalls sind 15 Empfehlungen in der Praxis ausreichend.

8.1.1.7 Gesamtlaufzeit

Diese ergibt sich durch (4), (6) und (7) und hat daher die folgende Zeitkomplexität:

$$
\begin{aligned}
& O(\log_2(|C|*|WL|) \\
& + O(\log_2(|C|*|WL|*|WT|) \\
& + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}\,|C|/2)+O(|C|*\log_2|C|) \\
= \quad & O(\log_2(|C|^2*|WL|^2*|WT|) + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}\,|C|/2)+O(|C|*\log_2|C|)
\end{aligned}
$$

Da CRIC für `n` den konstanten Wert "`10`" verwendet folgt die vereinfachte Laufzeitkomplexität:

$$
\begin{aligned}
= \quad & O(\log_2(|C|^2*|WL|^2*|WT|) + O((\log_2|WT|)+10*|C|/2)+O(|C|*\log_2|C|) \\
= \quad & O(\log_2(|C|^2*|WL|^2*|WT|^2) + O(|C|)+O(|C|*\log_2|C|)
\end{aligned}
$$

8.1.1.8 Eine frühere Variante von CRIC

In einer früheren Variante von CRIC wurde der vollständige inverse Index auf Basis von Relationen des DBMS gebildet und nicht die Volltextsuche des DBMS verwendet. Die Laufzeit (7) wurde in diesem Falle durch folgende SQL-Anfrage bestimmt:

```
SELECT text.*
FROM text,worte,enthaelt
WHERE text.id=enthaelt.text (mit enthaelt.text als Fremdschlüssel auf text.id)
AND enthaelt.worte=worte.id (mit enthaelt.worte als Fremdschlüssel auf worte.id)
AND (
      worte.wort=a1 OR
      (…) OR
      worte.wort=aq
      )
```

Das ergibt in Relationenalgebra dargestellt (wobei die Sequenz der mit `OR` verknüpften Prädikate zunächst nur für ein Prädikat betrachtet wird):

(i) $E_1=SL_{[worte.wort=a1]}(worte)$

(ii) $E_2=JN_{[enthaelt.worte=E1.id]}(enthaelt,E_1)$

(iii) $E_3=JN_{[text.id=E2.text]}(text,E_2)$

(iv) $PJ_{(k1,k2)}(E_3)$

Mit `text.id` und `text.volltext` als `k1`-tes und `k2`-tes Attribut in E_3.

Daraus ergibt sich folgende Laufzeitkomplexität:

(i) $O(\log_2|worte|)$

Mit der Selektivität der Selektion und `wort` als Primärschlüssel von `worte` folgt $|E_1|=1$ und damit weiter

(ii) $O(1*\log_2 1)+O(|enthaelt|*\log_2|enthaelt|)$

Mit dem Sonderfall der Selektivität der 1:N Verbundbildung folgt $|E_2|=|enthaelt|$. Da enthaelt aber mindestens ein Tupel für jedes Wort W aus worte enthält, kann man mit dieser Metainformation die Abschätzung auf $|E_2|=(|enthaelt|-|worte|)$ verfeinern. Damit folgt weiter

(iii) $O((|enthaelt|-|worte|)*\log_2(|enthaelt|-|worte|)$
$+ O(|text|*\log_2|text|)$

Mit dem Sonderfall der Selektivität der 1:N Verbundbildung folgt $|E_3|=|E_2|=(|enthaelt|-|worte|)$. Damit gilt für die Laufzeitkomplexität der Projektion:

(iv) $O(|enthaelt|-|worte|)$

Insgesamt folgt eine Laufzeitkomplexität von

(v) $O(\log_2|worte|)$
$+ O(1*\log_2 1)+O(|enthaelt|*\log_2|enthaelt|)$
$+ O((|enthaelt|-|worte|)*\log_2(|enthaelt|-|worte|)$
$+ O(|text|*\log_2|text|)$
$+ O(|enthaelt|-|worte|)$

Betrachtet man statt einem die q mit OR verknüpften Prädikate, so kann die Anfrage auf q Vereinigungen mit einfachem Prädikat (wie oben betrachtet) umgestellt werden. Damit ergibt sich für die Laufzeit-Komplexität (da jeder Text alle q Worte enthalten kann) neben (v) noch $O(q*|text|)$.

8.1.2 Text löschen

Analog zum Einfügen eines neuen Textes soll das entfernen eines bestehenden Textes analysiert werden.

8.1.2.1 Text aus C entfernen

Das Löschen von Text T (mit „nr“ als Primärschlüsselwert) in Relation C erfolgt durch ein DELETE mit Index-Erhalt:

```
      DELETE FROM text
      WHERE id=nr
=     DEL[id=nr](text)
```

damit ergibt sich mit 7.5.1.8 die Laufzeitkomplexität

$2*O(\log_2|C|)$ (1)

8.1.2.2 Textanalyse

Eine Textanalyse entfällt, da man über WT die Worte des Textes erhält.

8.1.2.3 Wortliste WL

Die Worte der Wortliste WL werden beim Löschen eines Textes nicht gelöscht, sondern immer nur dekrementiert (UPDATE). Es ergibt sich für die Operationen auf WT, die (|T|/len) mal (für jedes Wort aus T) ausgeführt werden analog zu 8.1.1.3 allerdings ohne den INSERT-Anteil:

$(|T|/len)*O(\log_2|WL|)$ (2)

8.1.2.4 CRIC Gesamtlaufzeit

Insgesamt ergibt sich:

$$2*O(\log_2|C|) \quad (1)$$
$$+(|T|/len)*O(\log_2|WL|) \quad (2)$$

mit $|T|$=Anzahl Zeichen in T. Da $|T|$ auch hier nicht mit der Zeitachse wächst und begrenzt ist (maximale Textlänge) kann es wieder als Konstante betrachtet werden, was folgende Vereinfachung ermöglicht:

$$2*O(\log_2|C|)$$
$$+ O(\log_2|WL|)$$

$$= \quad 2*O(\log_2|C|)+O(\log_2|WL|)$$
$$= \quad O(2*(\log_2|C|)+\log_2|WL|)$$
$$=^{47} \quad O(\log_2|C|+\log_2|WL|)$$

$$= \quad O(\log_2(|C|*|WL|)) \quad (3)$$

8.1.2.5 Kosten des Volltext-Index des Datenbanksystems

Beim Löschen eines Textes muss wieder der Volltext-Index des Datenbanksystems aktualisiert werden. Hier ergibt sich analog zum Einfügen eines Textes wieder die analoge Laufzeit wie bei CRIC zuzüglich der Pflege von WT:

Wort-Text Relation WT

Das Löschen des Wort-Text Tupels aus WT hat folgende Laufzeit:

$$(|T|/len)*[O(\log_2(2*|WT|-(|T|/len))/2)+O(\log_2|WT|)]$$

Laufzeit

$$O(\log_2(|C|*|WL|))$$
$$+(|T|/len)*[O(\log_2(2*|WT|-(|T|/len))/2)+O(\log_2|WT|)] \quad (4)$$

8.1.2.6 Gesamtlaufzeit

Diese ergibt sich aus (3) und (4) und hat daher die folgende Zeitkomplexität:

$$O(\log_2(|C|*|WL|))$$
$$+ O(\log_2(|C|*|WL|))$$
$$+ (|T|/len)*[O(\log_2(2*|WT|-(|T|/len))/2)+O(\log_2|WT|)]$$

[47] Entfernen der Konstanten; Beschränkung auf Variable.

Zusammengefasst und mit |T| als Konstante gilt weiter:

$$
\begin{aligned}
& O(\log_2(|C|^2*|WL|^2)) \\
& + [O(\log_2|WT|)+O(\log_2|WT|)] \\
= \quad & O(\log_2(|C|^2*|WL|^2)) \\
& + O(\log_2|WT|^2) \\
= \quad & O(\log_2(|C|^2*|WL|^2*|WT|^2))
\end{aligned}
$$

8.1.3 Text ändern

Ein geänderter Text wird durch eine Konkatenation der Operationen "Text löschen" und "Neuer Text" erzielt und hat daher die summierte Laufzeit von:

$$
\begin{aligned}
& O(\log_2(|C|^2*|WL|^2*|WT|^2)) \\
& +O(\log_2(|C|^2*|WL|^2*|WT|^2) + O(|C|)+O(|C|*\log_2|C|) \\
= \quad & O(\log_2(|C|^4*|WL|^4*|WT|^4)) + O(|C|)+O(|C|*\log_2|C|)
\end{aligned}
$$

8.1.4 Reduktion der Zeitkomplexität durch eine Näherung

Betrachtet man die Suche nach verwandten Texten (8.1.1.6) als Funktion f_n so ist diese offensichtlich nicht symmetrisch. Daher gibt es auch keine Umkehrfunktion f_n^{-1}. Daher muss f_n für jeden neuen Text T_{new} in C für alle T_i aus C neu berechnet werden, weil sich der Funktionswert durch den neuen Text T_{new} geändert haben könnte.

Die Laufzeitkomplexität der Operation $f_n(T_{neu})$ für alle Texte $T_{i=1\ldots|C|}$ aus C neu zu berechnen ist

$$O(|C|*[(n*\log_2|WT|+\Sigma_{i=1\ldots n}|C|/2)+(|C|*\log_2|C|)])$$

mit n=Anzahl der Worte der Anfrage; da dies eine Konstante ist, ergibt sich:

$$
\begin{aligned}
& O(|C|*[(\log_2|WT|+|C|/2)+(|C|*\log_2|C|)]) \\
= \quad & O(|C|*\log_2|WT|+|C|^2+(|C|^2*\log_2|C|)) \\
= \quad & O(|C|*\log_2|WT|+|C|^2+|C|^2*\log_2|C|) \\
> \quad & O(|C|^2)
\end{aligned}
$$

Um dieses quadratische Wachstum der Laufzeitkomplexität zu vermeiden ist eine Näherung für f_n für alle $T_1,\ldots,T_{|C|}$ in C erforderlich. Für diese Näherung wird eine Symmetrie von f_n angenommen und lediglich für die $T_1,\ldots,T_n$ des Ergebnisses von $f_n(T_{neu})$ wird eine Ersatzfunktion f_{sub} der fehlenden inversen Funktion f_n^{-1} angewendet, die wie folgt definiert ist:

$$
\begin{aligned}
& \forall\ T_{i=1\ldots|C|},\ T_i \in f_n(T_{neu}): \\
& f_{sub}(T_i) = f_n(T_i) \cup \{T_{neu}\}
\end{aligned}
$$

Veranschaulicht wird T selbst den dazu verwandten Texten aller Ti als inhaltlich verwandten Text hinzugefügt. Die korrekte Funktion $f_n(T_{i=1\ldots|C|})$ wird in Paketen von t Texten berechnet wenn das System Leerlauf hat. Auf diese Weise „kalibriert“ sich das semantische Netz ständig selbst nach.

8.2 Ein Konzept für Hochlastszenarien

Wenn zu einem Text T die "verwandten" Texte angezeigt werden sollen, muss eine Suche für die verwandten Texte in der Hilfstabelle V durchgeführt werden. Die Zeitkomplexität dieser Operation ist (mit Sonderfall 1 der Selektion, da die eine Spalte von V einen Text maximal 15-mal enthält; die Kardinalität von V ist $15*|C|$):

$$O(\log_2(15*|C|)+15)$$
$$= O(\log_2(15*|C|))$$

Angesichts der zu erwartenden niedrigen Änderungsrate in C (im Vergleich zur Leserate) und in Anbetracht der geringen Wahrscheinlichkeit, dass sich $f_n(T)$ für einen Text T dadurch ändert, ist es nicht sinnvoll, diese Suche bei jedem Aufruf der verwandten Texte eines Textes T durchzuführen.

Durch ein Caching des Suchergebnisses für einen Text T kann die Zeitkomplexität deutlich reduziert werden. Mit p als Cache-Periode, A_w als der Anzahl der wiederholten Zugriffe auf T_W (deren verwandte Texte) innerhalb von p und A_n als der Anzahl der neuen Zugriffe auf T_e (deren verwandte Texte) in p resultiert:

$$O(\log_2(15*|C|))*(A_n)$$

anstelle von

$$O(\log_2(15*|C|))*(A_n+A_w)$$

Es ist offensichtlich, dass Caching besonders dann sinnvoll ist, wenn A_w sehr viel größer ist als A_n.

9 Das reale Laufzeitverhalten

Die Reaktionszeit und Lastverträglichkeit des Systems wurde auf einem großen deutschen Internet-Portal getestet. Mit rund 332.000 Texten und durchschnittlich über 50.000.000 Seitenabrufen pro Monat kann es als repräsentativ gelten. Insbesondere sind deutliche Nutzungsspitzen mit bis zu 322 Seitenabrufen pro Sekunde aufgrund aktueller Nachrichten oder Promotion (Newsletter) zu verzeichnen.

9.1 Testaufbau

Als Testumgebung kam ein Server mit folgender Ausstattung zum Einsatz:

- CPU: Intel Doppelprozessor Xeon 700
- Primärspeicher: 2 GB
- Sekundärspeicher: 100 GB RAID5

Zunächst wurden alle Texte aus einem Content Management System in einem initialen Lauf in Form von Paketen mit je 1.000 Texten an CRIC übergeben und in die Datenbasis aufgenommen:

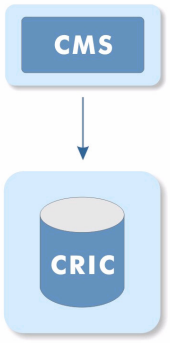


Abbildung 104:Die Daten werden aus einem Content Management System übernommen. Dazu wurde ein "Plugin" entwickelt, dass bei Veröffentlichung von Inhalten im CMS diese auch immer an CRIC weiter gibt.

In allen Texten des Portals wurden für den Benutzer unsichtbar (weiß auf weiß) die Links zu fünf inhaltlich verwandten Texten ausgegeben.

9.2 Ergebnisse

In 32,68% aller Seitenabrufe (durchschnittlich 16.340.000 Seitenabrufe pro Monat; die übrigen Seitenabrufe waren solche von Übersichtsseiten) war diese Ausgabe über einen Zeitraum von 12 Monaten aktiv. Bei den Nutzungsspitzen waren es 47,31% der Seitenabrufe, was einem Maximum von 152,34 Aufrufen mit Empfehlung verwandter Texte pro Sekunde entspricht. Bei den Lastspitzen zeigte wie erwartet das Caching Wirkung. Hier wurden in Spitzenzeiten bis zu 41,45% der Aufrufe aus dem Cache geliefert (mit Cache-Periode p = 5 Minuten). Daraus lässt sich eine maximale Berechnungsrate von 89,19 Ausgaben pro Sekunde ableiten. Die CPU-Auslastung bei dieser Berechnungsrate lag bei rund 40 Prozent. Durch einfache Umrechnung erhält man bis zu 40.850.000 Empfehlungen pro Monat als Obergrenze für die gegebene Hardware.

9.3 Fazit

Mit rund 40 Millionen Seitenabrufen pro Monat (was durch Hardware weiter skalierbar wäre) ist das Verfahren auch auf großen Portalen einsetzbar und die Prämisse (ii) (siehe dazu Seite 4) offensichtlich erfüllt.

10 Qualitative Evaluation

Eine objektive qualitative Bewertung der Empfehlungen kann nur durch einen Vergleich erfolgen. Klassischer Weise wird ein Vergleich der von Benutzern als "verwandt" eingestuften Texten und den von einem Empfehlungssystem selektierten Texten auf einer Testbasis von Texten gezogen. Hier kommt allerdings bei großen Textzahlen das Problem ins Spiel, dass der Benutzer selbst keine ausreichende "Recall"-Quote erreicht (er kann nicht hunderttausende von Texten auf Verwandtschaft prüfen). Lässt man den Benutzer die empfohlenen Texte explizit bewerten, ist es schwierig eine ausreichend große Grundgesamtheit mit kontinuierlicher Teilnahme zu realisieren.

Mit den Cranfield Experimenten [CLE1967] sollte eine Testumgebung zum objektiven Vergleich verschiedener Textklassifikationsverfahren geschaffen werden. Dazu wurde ein fester Korpus (aus dem Themenfeld *Luft- und Raumfahrt*) von Experten manuell durchgängig klassifiziert.

Der Vergleich zu bereits vorgegebenen Test-Korpora (TREC-Daten, TDT-Korpora et cetera) ermöglicht zwar eine objektiven Bewertung mit ebenfalls darauf getesteten anderen Verfahren, birgt aber die Gefahr, dass die Bewertung der Relevanz von Texten in den Test-Korpora nur für eben diese Test-Korpora gelten. Selbst das kann nicht als gegeben betrachtet werden, da die menschlichen Bewertungen keine sichere Basis bieten. So hat die Relevanzbewertung unter verschiedenen Experten meist kein einstimmiges Ergebnis. Solche Vergleiche verschiedener Verfahren auf Test-Korpora sind demnach nicht hinreichend für die allgemeine Qualität eines Verfahrens (siehe dazu auch [HAN1997]).

Voorhees [VOO2002] weißt darauf hin, dass selbst die Test-Korpora nicht mit ausreichender Güte klassifiziert werden können, da es bei angenommenen 30 Sekunden pro Dokument und rund 800.000 Dokumenten im TREC-Test-Korpus rund neun Monate dauern würde, eine einzige Klasse (Thema) von Dokumenten zu bestimmen.

Und schließlich haben Turpin und Hersh in Feldstudien gezeigt, dass die Bewertungen auf Basis der TREC Test-Korpora nicht mit einem Vergleich mit menschlichen Testpersonen übereinstimmten [TUR2001].

Was benötigt wird, ist ein "objektiver" Maßstab für die Relevanz eines Textes in Bezug auf ein Thema, eine Suchanfrage oder – wie bei CRIC – einen anderen Text. Einen sehr guten Überblick über diese Problemstellung gibt Geisdorf [GRE2000], der auch aufzeigt, dass es einen solchen objektiven Maßstab – zumindest noch - nicht gibt.

Der gewählte Evaluationsansatz

Um dieses Dilemma des fehlenden Maßstabes zu umgehen, wurde als messbare Größe die Akzeptanz von ausgesprochenen Empfehlungen über einen langen Zeitraum bei geringer Fluktuation der Benutzer ausgemacht. Nun entsteht dadurch allerdings das Problem, dass man die Akzeptanz mit etwas vergleichen muss, um eine Aussage zu treffen.

Diese Vergleichsmöglichkeit wurde in Form manueller Empfehlungen durch Fachleute gefunden. Das heißt jeder Text bietet neben den durch Fachleute ausgesprochenen Empfehlungen auch solche, die durch CRIC ermittelt wurden. Da den Benutzern beide Empfehlungsquellen in gleicher Gestaltung zur Verfügung gestellt werden, wird ein Vergleich des automatischen Verfahrens zu Empfehlungen von Fachleuten möglich. Damit die Ergebnisse belastbar sind, muss der Test folgende Bedingungen erfüllen:

- große Anzahl Endbenutzer
- geringe Fluktuationsrate der Endbenutzer (damit neben "ansprechenden Titeln" die "Einschätzung der Güte" der Empfehlungen zum Tragen kommt)
- hohe Nutzungsfrequenz durch Benutzer
- statistische Messbarkeit der Akzeptanz
- langer Zeitraum
- große Textbasis
- regelmäßig neue Texte

All dies wurde im Rahmen des Testszenarios erfüllt. Daher sollen die Rahmendaten der Untersuchung nun im Detail vorgestellt werden

10.1 Rahmendaten

Die Untersuchung lief im Jahr 2005 über einen Zeitraum von 12 Monaten. Es wurde eine Textbasis von 95.000 Textobjekten mit einer durchschnittlichen Länge von 389 Worten in einem Intervall von [62,472] Worten verwendet. Im Durchschnitt wurden rund 4.000 Empfehlungen pro Monat ausgesprochen.

Die Qualität des Verfahrens wurde durch Integration in ein Internet-Portal für Elektrotechnik- und Life-Science-Themen evaluiert. Hinter dem Portal steht ein Fachverlag mit einer professionellen Redaktion. Die Nutzer des Portals weisen eine hohe Wiederkehrrate auf. Die Disposition dafür schafft die Tatsache, dass ausschließlich Leser des Printmediums Zugang zu der für die Untersuchung relevanten Funktion des Portals haben.

Zu jedem Text erhält der Benutzer zwischen ein bis fünf redaktionell erstellte Links zu verwandten Texten. Zur Evaluation wurden zusätzlich mit CRIC ermittelte Textempfehlungen in gleichem Umfang angeboten. Schließlich wurde noch ein dritter Link-Block mit ebenfalls automatisch ermittelten Textempfehlungen auf Basis von Metadaten (hierarchische, manuelle Strukturierung) angeboten. Dem Benutzer ist die Herkunft der einzelnen Empfehlungen nicht bekannt.

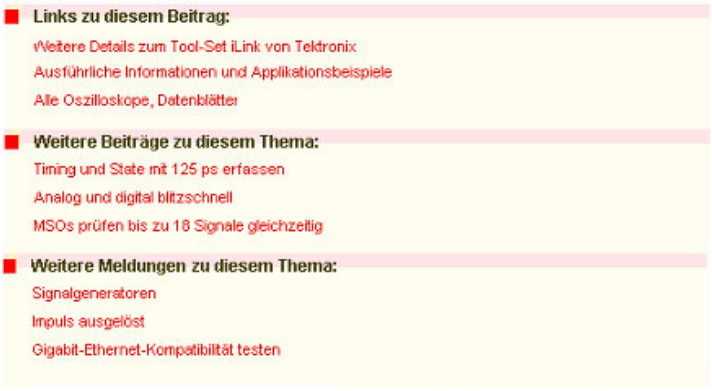


Abbildung 105:Die Empfehlungen aus Benutzersicht

Die drei Linkblöcke werden in der Reihenfolge:

- Redaktion
- CRIC
- Metadaten

angeboten und für den Benutzer in visuell voneinander abgesetzten Listen präsentiert.

10.2 Der Vergleich zu redaktioneller Selektion

Die drei Linkblöcke werden in der Reihenfolge "Redaktion", "CRIC", "Metadaten" angeboten und für den Benutzer in visuell voneinander abgesetzten Listen präsentiert. Dem Benutzer ist die Herkunft der einzelnen Empfehlungen nicht bekannt. Über einen Zeitraum von zwölf Monaten wurden folgende Link-Nutzungsraten (angezeigte Links pro genutzte Links als Durchschnittswerte mit Standardabweichung [S]) erzielt:

- Redaktion : 2,03 (S: 0,289491708)
- CRIC : 2,18 (S: 0,267472775)
- Metadaten : 6,26 (S: 0,744574389)

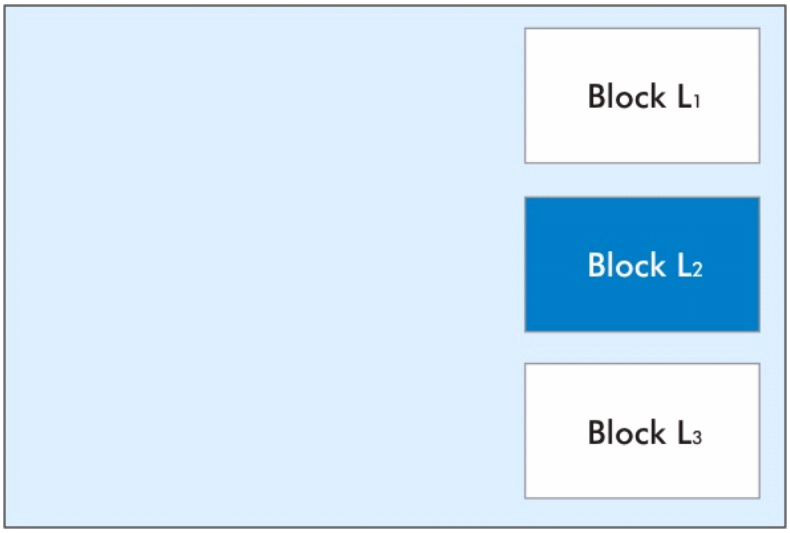


Abbildung 106:Abstrakte Aufteilung der Empfehlungen in Blöcke

Berücksichtigt man die Bevorzugung von weiter oben stehenden Links, ist der Wert von 2,18 für die CRIC Empfehlungen ein Wert, der die Qualität der Textempfehlungen in die Nähe redaktionell gepflegter Textempfehlungen rückt. Der deutliche Abfall der auf Basis der Metadaten erzeugten Links bestärkt dies nochmals.

10.2.1 Details

Im Folgenden werden die verdichteten Ergebnisse der Untersuchung aufgeführt.

	JAN	FEB	MRZ	APR	MAI	JUN	JUL	AUG	SEP	OKT	NOV	DEZ
CRIC	1,20	1,97	2,64	1,51	1,92	1,83	2,63	2,05	2,89	2,47	2,43	2,64
Redaktion	1,61	1,90	1,83	1,82	1,96	1,91	1,68	1,53	1,52	3,21	2,70	2,68
Struktur	5,64	6,50	5,66	5,30	5,60	6,59	6,59	5,62	5,78	8,46	6,62	6,74

Tabelle 8: Die Untersuchungsergebnisse nach Durchschnittswerten pro Monat

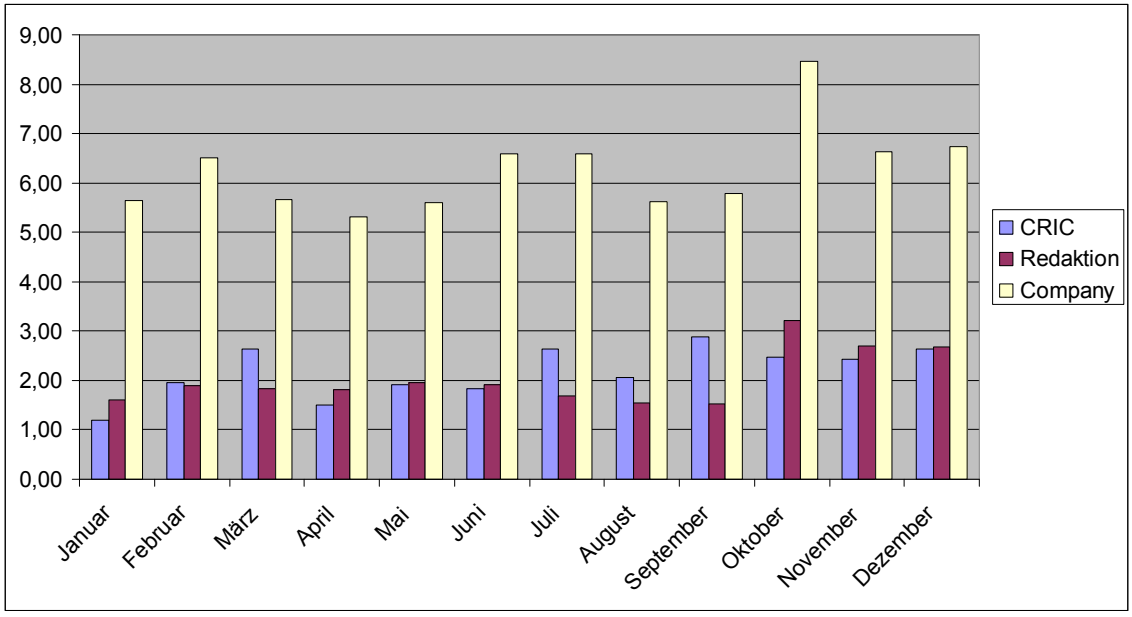


Abbildung 107:Die Untersuchungsergebnisse im Überblick. Die y-Achse gibt die mittlere Anzahl angezeigter Empfehlungen im Verhältnis zu einer genutzten Empfehlung.

10.3 Visualisierung der semantischen Proximität

Zur Visualisierung der semantischen Distanzen der Texte wurde ein java-basiertes Werkzeug verwendet. Es kann Texte als Kästen anzeigen und durch gerichtete Linien mit einem bestimmten Abstand verbinden. Damit Texte umso näher beieinander stehen, je ähnlicher sie sich sind, wird als "Abstand" der Confidence-Wert normiert[48].

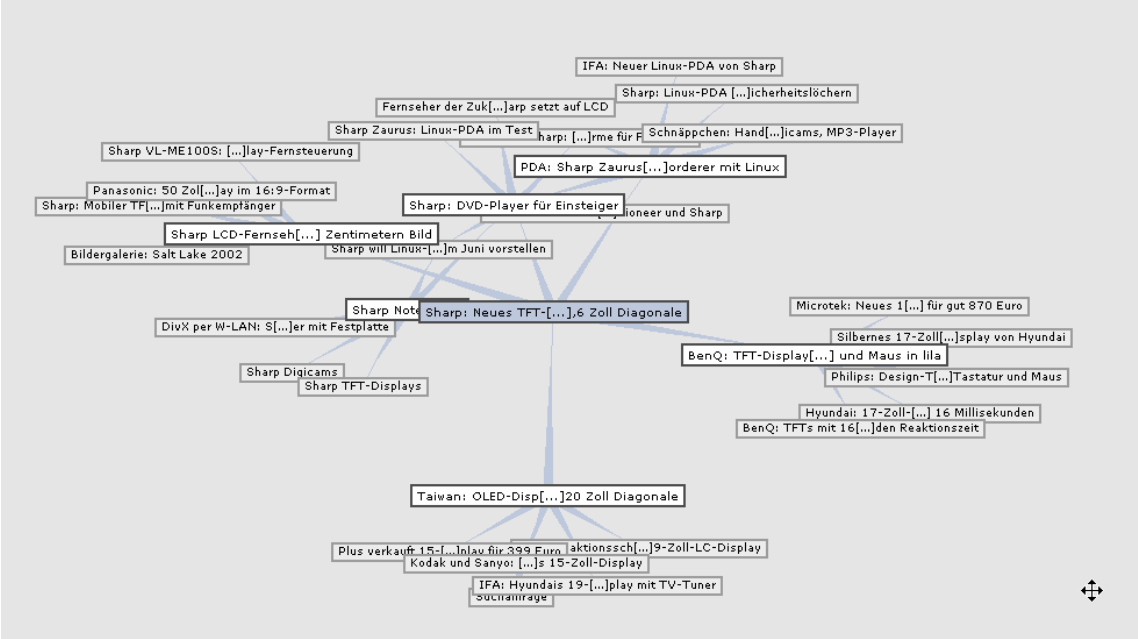


Abbildung 108: Visualisierung der semantischen Proximität im Browser.

Es steht immer ein Text im Zentrum der Darstellung. Um ihn herum werden die direkt verwandten Texte, sowie wiederum deren verwandte angezeigt. Durch Anklicken eines beliebigen Textes kann dieser zum neuen Zentrum gemacht werden. Außerdem lässt sich zu einem Text der in CRIC gespeicherte Text einblenden.

[48] Die folgende Aufstellung gibt die vorgenommene Anpassung wider:

```
Confidence < 10      : Abstand = 15
10 <= Confidence < 20: Abstand = 13
20 <= Confidence < 30: Abstand = 11
30 <= Confidence < 40: Abstand = 9
40 <= Confidence < 50: Abstand = 7
50 <= Confidence < 60: Abstand = 5
60 <= Confidence < 70: Abstand = 3
70 <= Confidence     : Abstand = 1
```

TEIL IV - RESUMÉ UND AUSBLICK

Zusammenfassung – Teil V
Zusammenfassung der Arbeit in deutscher Sprache.
Zusammenfassung der Arbeit in englischer Sprache.
Ausblick auf Weiterentwicklungen.

11 Zusammenfassung

Bereits seit Beginn des 20. Jahrhunderts wird die Problematik der "Informationsflut" in der Wissenschaft diskutiert (exemplarisch sei auf [OPP1927] verwiesen). Der amerikanische Trendforscher John Naisbitt prägte den Satz "Wir ertrinken in Informationen, aber hungern nach Wissen" [NAI1982]. Dass die Informationsmenge mittlerweile nicht mehr exponentiell wächst, ist einzig dem erreichten Maß an Informationsausstoß zu verdanken [MAR2002]. Aus einer unüberschaubaren Informationsmenge die "passenden" Texte selektieren zu können, ist daher wichtiger denn je.

Die Lösungsansätze des Problems "passende" Texte zu finden, lassen sich in zwei große Gruppen unterteilen: die "push"- und die "pull"-Ansätze. Zu ersteren zählt man Empfehlungssysteme (Recommender Systems) aller Art, zu letzteren die unterschiedlichen Suchverfahren.

Im Rahmen dieser Arbeit wird ein neues Verfahren für ein Empfehlungssystem vorgestellt. Es unterscheidet sich von ähnlichen Verfahren im Wesentlichen dadurch, dass eine Heuristik zum Einsatz kommt, die ein TF-IDF-Derivat mit Eigenschaften der Textstruktur verbindet und darauf sprachunabhängig (für romanische und indogermanische Sprachen) eine asymmetrische vorberechnete Distanzmatrix aufbaut. Als Basis können beliebige unstrukturierte Texte verwendet werden. Insbesondere werden keine Metadaten, aber auch keine Thesauri oder Korpora benötigt. Dabei bestanden die im Folgenden aufgeführten Prämissen (i) – (iv):

(i) Akzeptanz beim Benutzer
  a. Hohe Qualität der Empfehlungen

(ii) Garantiert schnelle Antwortzeiten
  a. Auch bei großen Textmengen
  b. Auch in Hochlastumgebungen

(iii) Erschließung aller vorhandenen Quellen
  a. Unstrukturierte Texte ohne Metadaten
  b. Unterschiedliche Dateiformate

(iv) Kein manueller Eingriff durch Autor oder Benutzer
  a. Automatisierte Verarbeitung

Die bekanntesten Vertreter der Empfehlungssysteme basieren auf dem Content-based Filtering (CBF), dem Collaborative Filtering (CF) oder hybriden Varianten. Diese Ansätze werden in der Literatur ausgiebig behandelt ([RES1997], [SAR2000] et cetera).

Content-based filtering (CBF)

Beim CBF werden die Daten und/oder Metadaten eines Textes verwendet, um verwandte Texte zu finden. Dem Benutzer werden dann die Texte angeboten, die den von ihm implizit oder explizit bewerteten Texten oder dem gerade angezeigten Text am ähnlichsten sind. Ein Verfahren, das diesen Ansatz wählt, ist beispielsweise "The information lens" [MAL1986]. Es setzt allerdings beim Autor der Texte eine explizite Strukturierung derselben voraus. Dem Nutzer werden dann über strukturierte Filter passende Texte zugespielt. Ein anderes Beispiel ist INFOSCOPE [FIS1991], das keine Vorarbeit beim Autor voraussetzt und den Nutzer beim Aufbau und Anpassen der Filter unterstützt. Weitere prominente Vertreter sind Newsweeder [LAN1995], InfoFinder [KRU1997], News Dude [BIL1999] und LIBRA [MOO2000].

Collaborative filtering (CF)

Das CF nutzt anstelle der Verwandtschaft von Texten die Ähnlichkeit von Benutzerprofilen. Empfehlungen für einen Benutzer werden aus dem Verhalten anderer Benutzer mit ähnlichem Profil gewonnen. Ausgehend von Hey [HEY1989] existieren viele Beispiele für CF. Einige davon sind Tapestry [GOL1992], GroupLens [RES1994], Ringo [SHA1995], Siteseer [RUC1997] und Jester [GOL2001].

Hybrid-Systeme

In der Praxis sind auch Hybrid-Systeme zu finden, die Content-based Filtering und Collaborative Filtering verbinden. Bekannte Vertreter dieser Gruppe sind INFOS [MOC1996], Fab [BAL1997] und Tango [CLA1999].

Verwandte Texte finden

Es gibt zahlreiche Ansätze semantisch verwandte Texte für einen gegebenen Text T zu berechnen. Der Vorgang lässt sich in zwei Schritte zerlegen. Zunächst werden die Texte in eine vergleichbare Form transformiert. Erst dann kann ein Distanzmaß den semantischen Abstand berechnen.

Die bekanntesten Transformationsvarianten ermitteln meist Fragmente (Worte, Sätze et cetera) eines Textes. Auf Basis von rund 50 untersuchten Empfehlungssystemen war TF-IDF das meist genutzte Verfahren. Bei der Distanzermittlung werden vorrangig symmetrische vektorbasierte Verfahren verwendet.

Viele Verfahren setzen den manuellen Eingriff seitens der Textautoren und/oder der Benutzer voraus und erfüllen nicht die Zielsetzung auf unstrukturierten Daten ohne Metadaten zu arbeiten. Ein Verfahren, das diese Vorgaben erfüllt, ist beispielsweise der Remembrance Agent [RHO1996], der verwandte Texte auf Basis des Vektorabstandes als ad hoc Anfrage ermittelt. Watson [BUD1999] und Margin Notes [RHO2000] ermitteln die wichtigsten Begriffe eines Textes und nutzen diese dann zur ad hoc Suche nach verwandten Texten. Es findet aber keine Vorberechnung statt, was in Hochlastszenarien problematisch ist.

Bestimmende Worte eines Textes ermitteln

Es existieren Standardverfahren, um die Schlüsselworte eines unstrukturierten Textes zu ermitteln. Diese basieren in der Regel darauf die relative Bedeutung eines Wortes für einen gegebenen Text zu bestimmen und die wichtigsten Worte als Schlüsselworte abzuleiten. Das bekannteste Verfahren ist sicherlich der TF-IDF Ansatz [SAL1981]. Es handelt sich um ein einfaches Distanzmaß, welches das häufige Auftreten eines Wortes im konkreten Text "belohnt" und ein häufiges Vorkommen über alle Texte "bestraft".

Das CRIC Verfahren zur Schlüsselwortselektion stellt ein TF-IDF-Derivat dar, das die Bedeutung des Wortes W eines gegebenen Textes T der Bestandteil der Textbasis C ist, aus folgenden Parametern ableitet:

- $N_T(W)$: Anzahl der Instanzen des Wortes W in Text T
- $N_C(W)$: Anzahl der Instanzen von W in C mit $C=\{T_1, \ldots, T_n\}$
- $P_T(W)$: Faktor der von Position P des Wortes W im Text T abhängt
- $I_C(W)$: Anzahl der Texte mit mindestens einer Instanz des Wortes W
- W_i: i-tes Vorkommen des Wortes W in Text T
- I: obere Schranke für $I_C(W)$
- S: obere Schranke für $N_C(W)$

Es wird eine Worterkennung (Token-Bildung) auf Text T durchgeführt, um dies anwenden zu können.

Textstruktur und Häufigkeit

Wenn ein Wort am Anfang oder Ende eines Textes T lokalisiert ist, wird es höher bewertet. Diese Gewichtung trägt der Bedeutungsdynamik in Texten Rechnung nach der wichtige Inhalte zumeist am Anfang (Titel, Einleitung) und Ende (Zusammenfassung, Fazit) eines Textes zu finden sind.

Nach zahlreichern Tests hat sich die folgende Aufteilung als besonders effizient erwiesen:

$$\sum_{i=1\ldots N_T(W)} P_T(W_i)$$

als Gewichtung des Wortes W über alle Vorkommen von W in Text T und mit

$$P_T(W_i) = \begin{cases} 4{,}5 & falls \quad Pos_T(W) \le |T| * 0{,}20 \\ 3 & falls \quad Pos_T(W) > |T| * 0{,}90 \\ 2 & sonst \end{cases}$$

wobei $Pos_T(W)$ die Position von W in T und $|T|$ die Anzahl der Zeichen in Text T sind.

Ist der Text sehr kurz (kleiner als 100 Zeichen), wird keine Aufgliederung in Bewertungsbereiche vorgenommen, sondern der komplette Text wie ein Textanfang (Gewichtung "4,5") gewertet. Ist der Text sehr groß (größer als 10.000 Zeichen) wird der Textanfang fix auf 2.000 Zeichen und das Ende fix auf 1.000 Zeichen gesetzt.

Stoppwortliste (SWL)

Ein Wort W ist kein Stoppwort der Stoppwortliste SWL, wenn die Anzahl der Instanzen von W in Texten aus C ($N_C(W)$) kleiner als ein bestimmter Prozentsatz "S" ist oder die Anzahl der Texte aus C in denen W vorkommt ($I_C(W)$) kleiner als ein bestimmter Prozentsatz "I" ist. Andernfalls ist W ein Stoppwort und wird sofern noch nicht vorhanden zur Stoppwortliste SWL hinzugefügt. Es gilt:

$$w(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq S \right) \wedge \left(\frac{I_C(W)}{|C|} \geq I \right) \\ \sum_{i=1\ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

Die Stoppwortliste SWL wird um W erweitert, wenn w(W)=0 gilt. Der "Stoppwert" S wird in Prozent der maximalen Anzahl der Instanzen eines Wortes in C im Verhältnis zur Anzahl aller Worte W_C von C definiert. Als geeigneter Wert für S hat sich 1,0% erwiesen. Für I, den maximalen Prozentsatz aller Texte aus C in denen ein Wort vorkommen darf, um kein Stoppwort zu sein, wird 3,0% verwendet.

Die Stoppwortliste SWL wird dynamisch erzeugt, wenn ein oder mehrere Texte in die Textbasis C eingefügt werden. Eine individuelle Stoppwortliste berücksichtigt die Ausprägung der Textbasis C. So ist beispielsweise der Name eines Unternehmens ("Miller & Co") in der Textbasis eben dieses Unternehmens sehr wahrscheinlich ein Stoppwort, das in einem Großteil der Texte vorkommt (E-Mail Signatur, Fußzeile in Dokumenten et cetera), in einer generischen Stoppwortliste aber nicht vertreten wäre.

Verwandte Texte ermitteln

Nachdem die Schlüsselworte eines Textes T ermittelt und bezüglich ihrer Gewichtung absteigend sortiert sind, werden die ersten m Schlüsselworte $W_1, \ldots, W_m$ als Anfrage für eine Volltextsuche auf einem Datenbanksystem verwendet. Dazu muss die Textbasis C als Volltext im betreffenden Datenbanksystem gehalten werden. Weil es ein zweischneidiges Schwert darstellt (Ambiguierung durch Wortstammreduktion; [KAN2000][HUL1996]) und Suchalgorithmen in Datenbanksystemen das Stemming ohnehin berücksichtigen, wurde es bewusst nicht in das CRIC Verfahren zur Erzeugung der Schlüsselworte integriert.

Tests haben gezeigt, dass der optimale Wert für m sehr stark vom verwendeten Datenbanksystem abhängt. Allerdings lagen alle Werte in einem Intervall von 5 bis 10 Schlüsselworten. Bei dem in der Evaluation eingesetzten Datenbanksystem MySQL wurde der Wert auf "10" gesetzt. Tendenziell steigert ein großes m die Precision, senkt aber die Recall Quote.

Das Ergebnis der Suche ist eine Menge von Texten T (Resultset), die nach der Confidence (Con_T) sortiert sind. Das Resultset besteht also aus Tupeln (T,Con). Die ersten n Texte $T_{i=1\ldots n}$ mit einer Confidence $Con(T_i)$ größer dem Schwellwert ConMin werden selektiert.

Linguistische Motivation

Der gewählte Lösungsansatz von CRIC basiert auf der Annahme, dass die Bedeutung eines ausreichend kurzen Textes auf wenige Worte zurückgeführt werden kann. Ferner disambiguieren sich diese Worte wechselseitig. Letzteres hat eine breite linguistische und philosophische Basis.

- Firth: „You shall know a word by the company it keeps" [FIR1957]
- Wittgenstein: „Man kann für eine große Klasse von Fällen der Benützung des Wortes ›Bedeutung‹ - wenn auch nicht für alle Fälle seiner Benützung – dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.“ [WIT1968]

Laufzeitkomplexität

Die Laufzeiten wurden in O-Notation durch Analyse der benötigten DBMS-Operationen und anderen Algorithmen bestimmt.

Neuer Text

Die Laufzeitkomplexität zum Verarbeiten eines neuen Textes ist durch folgende Bestandteile geprägt:

(i) Speichern eines Textes, Bestimmen und Speichern der relevanten Worte
(ii) Aufnahme des Textes in den inversen Index des DBMS
(iii) Suche verwandter Texte auf dem DBMS

Mit C als Textbasis, sowie WL und WT als Datenstrukturen für die Worte und die Wort-Text-Beziehungen in CRIC sowie |C|, |WL| und |WT| als deren Kardinalitäten ergibt sich folgende Zeitkomplexität:

$$(i) \quad O(\log_2(|C|*|WL|)$$
$$(ii) \quad +O(\log_2(|C|*|WL|*|WT|)$$
$$(iii) \quad + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}|C|/2)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^2*|WL|^2*|WT|) + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}|C|/2)+O(|C|*\log_2|C|)$$

Da CRIC für n den konstanten Wert "10" verwendet folgt die vereinfachte Laufzeitkomplexität:

$$= O(\log_2(|C|^2*|WL|^2*|WT|) + O((\log_2|WT|)+10*|C|/2)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^2*|WL|^2*|WT|^2) + O(|C|)+O(|C|*\log_2|C|)$$

Text löschen

Die Laufzeitkomplexität für das Löschen eines Textes ist durch folgende Bestandteile geprägt:

(i) Entfernen des Textes, dekrementieren der Anzahl der Worte im Wort-Index
(ii) Entfernen des Textes aus dem inversen Index des DBMS

Mit C als Textbasis, sowie WL und WT als Datenstrukturen für die Worte und die Wort-Text-Beziehungen in CRIC sowie |C|, |WL| und |WT| als deren Kardinalitäten ergibt sich folgende Zeitkomplexität:

$$(i) \quad O(\log_2(|C|*|WL|))$$
$$(ii) \quad + O(\log_2(|C|*|WL|))+ O(\log_2|WT|^2)$$

Zusammengefasst ergibt dies:

$$O(\log_2(|C|^2*|WL|^2 *|WT|^2))$$

Text ändern

Ein geänderter Text wird durch eine Konkatenation der Operationen "Text löschen" und "Neuer Text" erzielt und hat daher die summierte Laufzeit von:

$$O(\log_2(|C|^2*|WL|^2*|WT|^2)) +O(\log_2(|C|^2*|WL|^2*|WT|^2) +O(|C|)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^4*|WL|^4 *|WT|^4)) + O(|C|)+O(|C|*\log_2|C|)$$

Reduktion der Zeitkomplexität durch eine Näherung

Betrachtet man die CRIC Suche nach verwandten Texten als Funktion f_n so ist diese offensichtlich nicht symmetrisch. Daher gibt es auch keine Umkehrfunktion f_n^{-1}. Daher muss f_n für jeden neuen Text T_{new} in C für alle T_i aus C neu berechnet werden, weil sich der Funktionswert durch den neuen Text T_{new} geändert haben könnte. Die Laufzeitkomplexität der Operation $f_n(T_{neu})$ für alle Texte $T_{i=1\ldots|C|}$ aus C neu zu berechnen ist

$$O(|C|*[(n*\log_2|WT|+\Sigma_{i=1\ldots n}|C|/2)+(|C|*\log_2|C|)])$$

mit n als Anzahl der Worte der Anfrage; da dies eine Konstante ist, ergibt sich:

$$O(|C|*[(\log_2|WT|+|C|/2)+(|C|*\log_2|C|)])=O(|C|*\log_2|WT|+|C|^2+(|C|^2*\log_2|C|))$$
$$= O(|C|*\log_2|WT|+|C|^2+|C|^2*\log_2|C|) > O(|C|^2)$$

Um dieses quadratische Wachstum der Laufzeitkomplexität zu vermeiden ist eine Näherung für f_n für alle $T_1, \ldots, T_{|C|}$ in C erforderlich. Für diese Näherung wird eine Symmetrie von f_n angenommen und lediglich für die $T_1, \ldots, T_n$ des Ergebnisses von $f_n(T_{neu})$ wird eine Ersatzfunktion f_{sub} der fehlenden inversen Funktion f_n^{-1} angewendet, die wie folgt definiert ist:

. $T_{i=1\ldots|C|}, T_i \in f_n(T_{neu}): f_{sub}(T_i) = f_n(T_i) \cup \{T_{neu}\}$

Veranschaulicht wird T selbst den dazu verwandten Texten aller T_i als inhaltlich verwandter Text hinzugefügt. Die korrekte Funktion $f_n(T_{i=1\ldots|C|})$ wird in Paketen von t Texten berechnet wenn das System Leerlauf hat.

Ein Konzept für Hochlastszenarien

Wenn zu einem Text T die "verwandten" Texte angezeigt werden sollen, muss eine Suche für die verwandten Texte in der CRIC Hilfstabelle V durchgeführt werden. Die Zeitkomplexität dieser Operation ist (die Kardinalität von V ist $15*|C|$):

$$O(\log_2(15*|C|)+15) = O(\log_2(15*|C|))$$

Angesichts der zu erwartenden niedrigen Änderungsrate in C (im Vergleich zur Leserate) und in Anbetracht der geringen Wahrscheinlichkeit, dass sich $f_n(T)$ für einen Text T dadurch ändert, ist es nicht sinnvoll, diese Suche bei jedem Aufruf der verwandten Texte eines Textes T durchzuführen. Durch ein Caching des Suchergebnisses für einen Text T kann die Zeitkomplexität deutlich reduziert werden. Mit p als Cache-Periode, A_w als der Anzahl der wiederholten Zugriffe auf T_W (deren verwandte Texte) innerhalb von p und A_n als der Anzahl der neuen Zugriffe auf T_e (deren verwandte Texte) in p resultiert:

$$O(\log_2(15*|C|))*(A_n)$$

anstelle von

$$O(\log_2(15*|C|))*(A_n+A_w)$$

Es ist offensichtlich, dass Caching besonders dann sinnvoll ist, wenn A_w sehr viel größer ist als A_n.

Geschwindigkeit in Echtzeitumgebungen

Die Reaktionszeit und Lastverträglichkeit des Systems wurde auf einem großen deutschen Internet-Portal getestet. Mit rund 320.000 Texten und durchschnittlich über 50.000.000 Seitenabrufen pro Monat kann es als repräsentativ für Hochlastumgebungen gelten. Insbesondere sind deutliche Nutzungsspitzen mit bis zu 322 Seitenabrufen pro Sekunde aufgrund aktueller Nachrichten oder Promotion (Newsletter) zu verzeichnen. Als Testumgebung kam ein Server mit folgender Ausstattung zum Einsatz:

- CPU: Intel Doppelprozessor Xeon 700
- Primärspeicher: 2 GB
- Sekundärspeicher: 100 GB RAID5

In 32,68% aller Seitenabrufe (durchschnittlich ca. 16 Mio. Seitenabrufe pro Monat; die übrigen Seitenabrufe waren solche von Übersichtsseiten) war die Ausgabe der Empfehlungen über einen Zeitraum von 12 Monaten aktiv. Bei den Nutzungsspitzen waren es 47,31% der Seitenabrufe, was einem Maximum von 152,34 Aufrufen mit Empfehlung verwandter Texte pro Sekunde entspricht. Bei den Lastspitzen zeigte wie erwartet das Caching Wirkung. Hier wurden in Spitzenzeiten bis zu 41,45% der Aufrufe aus dem Cache geliefert (mit Cache-Periode p = 5 Minuten). Daraus lässt sich eine maximale Berechnungsrate von 89,19 Ausgaben pro Sekunde ableiten. Die CPU-Auslastung bei dieser Berechnungsrate lag bei rund 40 Prozent. Durch einfache Umrechnung erhält man bis zu 40 Mio. Empfehlungen pro Monat als Obergrenze für die gegebene Hardware.

Qualität im Vergleich zu manueller Selektion

Die Qualität des beschriebenen Verfahrens wurde durch Integration in ein Internet-Portal für Elektrotechnik- und Life-Science-Themen evaluiert. Hinter dem Portal steht ein Fachverlag mit eine professionellen Redaktion. Die Nutzer des Portals weisen eine hohe Wiederkehrrate auf. Die Disposition dafür schafft die Tatsache, dass ausschließlich Leser des Printmediums Zugang zu der für die Untersuchung relevanten Funktion des Portals haben.

Zu jedem Text erhält der Benutzer zwischen ein bis fünf redaktionell erstellte Links zu verwandten Texten. Zur Evaluation werden zusätzlich mit CRIC ermittelte Textempfehlungen in gleichem Umfang angeboten. Schließlich wird noch ein dritter Link-Block mit ebenfalls automatisch ermittelten Textempfehlungen auf Basis von Metadaten (hierarchische, manuelle Strukturierung) angeboten.

Die drei Linkblöcke werden in der Reihenfolge "Redaktion", "Semantische Distanz", "Metadaten" angeboten und für den Benutzer in visuell voneinander abgesetzten Listen präsentiert. Dem Benutzer ist die Herkunft der einzelnen Empfehlungen nicht bekannt. Über einen Zeitraum von zwölf Monaten wurden folgende Link-Nutzungsraten (angezeigte Links pro genutzte Links als Durchschnittswerte mit Standardabweichung [S]) erzielt:

- CRIC : 2,18 (S: 0,267472775)
- Redaktion : 2,03 (S: 0,289491708)
- Metadaten : 6,26 (S: 0,744574389)

Berücksichtigt man die Bevorzugung von weiter oben stehenden Links, ist der Wert von 2,18 für die CRIC Empfehlungen ein Wert, der die Qualität der Textempfehlungen in die Nähe redaktionell gepflegter Textempfehlungen rückt. Der deutliche Abfall der auf Basis der Metadaten erzeugten Links bestärkt dies nochmals.

12 English Summary

The issue of ‘information overload’ has been a topic of concern in the sciences since the beginning of the twentieth century (see [OPP1927] for an example). As the American trend researcher John Naisbitt once put it, “We are drowning in information but starved for knowledge" [NAI1982]. The fact that the quantity of information is no longer growing exponentially is due only to the level of information output already achieved [MAR2002]. Hence, the ability to select from an unmanageable quantity of information only text that is relevant has become a particularly important task.

The approaches to solve the problem of finding relevant text can be divided into two large categories: the ‘push’ and the ‘pull’ approaches. The former includes Recommender Systems of all variants; the latter encompasses various search procedures.

In this thesis, a new procedure for a Recommender System will be introduced. It differs from similar procedures by, first, employing heuristics that link a TF-IDF derivative with the properties of the text structure and, second, generating an asymmetrical pre-calculated distance matrix that is independent of any particular language (at least as far as Roman and Indo-German languages are concerned). Any unstructured text may be used as the underlying basis, and no meta-data, thesauri or corpora are required. Further, premises (i) to (iv) below are applied, including the side constraints stated:

(i) User acceptance
  a. High quality of recommendations

(ii) Guaranteed fast response times
  a. even for large texts
  b. even in high-load environments

(iii) Tapping into all available sources
  a. Unstructured texts without metadata
  b. Different data formats

(iv) No manual intervention by the author or user
  a. Automated processing

The best-known representatives of the Recommender Systems are based on Content-based Filtering (CBF), Collaborative Filtering (CF) or hybrid variants. These approaches are treated extensively in the literature (see, for example [RES1997], [SAR2000] et cetera).

Content-based filtering (CBF)

In the case of CBF, the data and/or metadata of a text are used to find related text. The user is then offered those texts that are most similar to the text which is currently displayed or which the user has ranked either implicitly or explicitly. The ‘information lens’ [MAL1986] is one procedure employing such an approach. However, it presupposes that the author has structured the texts explicitly. Relevant texts are then made available to the user via structured filters. Another example is INFOSCOPE [FIS1991], which does not require any preliminary work by the author and which supports the user in generating and adapting the filters. Further prominent representatives are Newsweeder [LAN1995], InfoFinder [1997], News Dude [BIL1999] and LIBRA [MOO2000].

Collaborative filtering (CF)

CF relies on the similarity of user profiles rather than the relationship between texts. Recommendations for a specific user are generated from the behaviour of other users with similar profiles. Hey [HEY1989] outlines many examples of CF, such as Tapestry [GOL1992], GroupLens [RES1994], Ringo [SHA1995], Siteseer [RUC1997] and Jester [GOL2001].

Hybrid Systems

In practice, hybrid systems can be found that link Content-based Filtering and Collaborative Filtering. Known representatives of this group are INFOS [MOC1996], Fab [BAL1997] and Tango [CLA1999]..

The identification of related texts

Numerous approaches exist to identify texts that are related to a given text T. The process can be separated into two phases: the texts are transformed into a comparable form, before a distance measure can calculate the semantic distance between them.

The best-known transformation variants tend to determine fragments (words, sentences et cetera) of a text. From approximately 50 Recommender Systems examined, TF-IDF was the most widely used. The calculation of the distance is predominantly based on symmetrical vector-based procedures.

Many approaches presuppose the manual intervention on the part of the authors and/or users of the text and, thus, do not fulfil the objective of using unstructured data without metadata. A procedure that fulfils these requirements is, for instance, the Remembrance Agent [RHO1996], which determines related texts as an ad hoc query based on the vector distance. Watson [BUD1999] and Margin Notes [RHO2000] identify the most important terms of a text and use them subsequently to search for related texts in an ad hoc fashion. However, no pre-calculation is carried out beforehand, a fact that may pose problems in high-load scenarios.

The identification of key words in a text

Standard procedures are available to identify the key words of an unstructured text. They usually identify the relative meaning of a word for a given text and derive the most important words as key words. Arguably the best-known procedure is the TF-IDF approach [24], which constitutes a simple distance measure that 'rewards' the frequent appearance of a word in a specific text but 'penalises' the frequent occurrence across all texts.

The CRIC approach to key word selection represents a derivative of TF-IDF that deduces the meaning of the word W of a given text T, which is a component of the text base C, from the following parameters:

- $N_T(W)$: Number of instances of word W in text T
- $N_C(W)$: Number of instances of word W in C, where $C=\{T_1, \ldots, T_n\}$
- $P_T(W)$: Factor that depends on position P of word W in text T
- $I_C(W)$: Number of texts with at least one instance of word W
- W_i: instance i of word W in text T
- I: upper threshold for $I_C(W)$
- S: upper threshold for $N_C(W)$

In order to apply these parameters, a word identification (token generation) for text T is carried out.

Text structure and frequency

A word attains a higher ranking when it appears either at the beginning or the end of text T. This weighting corresponds to the distribution of importance in texts, where significant content occurs mostly at the beginning (title, introduction) and the end (summary, results).

In numerous tests, the following breakdown turned out to be particularly efficient

$$\sum_{i=1\ldots N_T(W)} P_T(W_i)$$

as weighting of word W across all instances of W in text T, and

$$P_T(W_i)=\begin{cases} 4{,}5 & falls \quad Pos_T(W) \le |T| * 0{,}20 \\ 3 & falls \quad Pos_T(W) > |T| * 0{,}90 \\ 2 & sonst \end{cases}$$

where $Pos_T(W)$ is the position of W in T and $|T|$ the number of characters in text T.

If the text is very short (smaller than 100 characters), no breakdown in ranking sections is carried out; instead, the complete text is evaluated as if it were the beginning of a text (weighting "4.5"). If the text is very large (greater than 10,000 characters), the beginning of the text is set at 2,000 characters and the end is set at 1,000 characters.

The Stopword List (SWL)

Die Stoppwortliste SWL wird dynamisch erzeugt, wenn ein oder mehrere Texte in die Textbasis C eingefügt werden. Eine individuelle Stoppwortliste berücksichtigt die Ausprägung der Textbasis C. So ist beispielsweise der Name eines Unternehmens ("Miller & Co") in der Textbasis eben dieses Unternehmens sehr wahrscheinlich ein Stoppwort, das in einem Großteil der Texte vorkommt (E-Mail Signatur, Fußzeile in Dokumenten et cetera), in einer generischen Stoppwortliste aber nicht vertreten wäre.

Word W is no stopword of stopword list SWL if the number of instances of W in texts of C ($N_C(W)$) is smaller than a given percentage S, or if the number of texts of C where W occurs ($I_C(W)$) is smaller than a given percentage I. Otherwise W is a stopword and is, if not included already, added to the stopword list SWL. The weight for word W is computed as:

$$w(W) = \begin{cases} 0 & falls \quad \left(\frac{N_C(W)}{\max\limits_{I_C(W_i)>0} (N_c(W_i))} \geq S \right) \wedge \left(\frac{I_C(W)}{|C|} \geq I \right) \\ \sum_{i=1\ldots N_T(W)} P_T(W_i) & sonst \end{cases}$$

The stopword list SWL is extended by value W if $w(W)=0$. The 'stop value' S is defined as the percentage of the maximum number of instances of a word in C in proportion to the number of all words W_C of C. A suitable value for S turned out to be 1.0%. For I, which is the maximum percentage of all texts of C in which a word can occur so as not to be a stopword, the value 3.0% is used.

The stopword list SWL is dynamically generated when one or several items of text are inserted into text base C. An individual stopword list takes into account the characteristic of text base C. For example, the name of a corporation ("Miller & Co") in the text base of this firm is very likely to be a stopword occurring in most texts (e-mail signatures, footers in documents, et cetera), whereas in a generic stopword list it would not be represented.

The identification of related texts

As soon as the key words of text T are identified and sorted in descending order in line with their weighting, the first m keywords $W_1, \ldots, W_m$ are used for a full-text search query on a database system. To that end, text base C has to be stored as full text in the relevant database system. Given that stemming is a double-edged sword (ambiguation through stemming; [KAN2000] [HUL1996]) and search algorithms in database systems take stemming into account anyway, it was deliberately not incorporated into the CRIC procedure to generate the key words.

Tests have shown that the optimum value of m depends significantly on the database system used. Even so, all values were within a range of 5 to 10 key words. In the case of the MySQL database system used during the evaluation, the value was set to 10. Generally speaking, large m tends to increase the precision, but decrease the recall ratio.

The result of the query is a number of texts T ('Resultset'), which is sorted according to the Relevance (Con_T). The Resultset consists of tuples (T,Con). The first n texts $T_{i=1\ldots n}$ with a Relevance $Con(T_i)$ greater than the threshold value ConMin are selected.

Linguistic motivation

The CRIC approach chosen here is based on the assumption that the meaning of short texts can be deduced from only few words. Further, these words experience a mutual disambiguation, an observation that has some wider linguistic and philosophical roots:

- Firth: „You shall know a word by the company it keeps" [FIR1957]
- Wittgenstein: „For a large class of cases – though not for all – in which we employ the word 'meaning' it can be defined thus: the meaning of a word is its use in the language". [WIT1968]

Runtime complexity

The runtimes were determined in O-notation through an analysis of the DBMS operations and other algorithms that are required.

A new text

The following factors determine the runtime complexity for the processing of new text:

(i) The storing of a text and the identification and storing of the relevant words,

(ii) The inclusion of the text in the inverse index {characters} of the DBMS,

(iii) The search for related texts on the DBMS.

With C as the text base; WL and WT as database structures for the words and word-text relationships within CRIC; and |C|, |WL| and |WT| as their cardinalities, the following time complexity applies:

$$(i) \quad O(\log_2(|C|*|WL|)$$
$$(ii) \quad +O(\log_2(|C|*|WL|*|WT|)$$
$$(iii) \quad + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}|C|/2)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^2*|WL|^2*|WT|) + O((n*\log_2|WT|)+\Sigma_{i=1\ldots n}|C|/2)+O(|C|*\log_2|C|)$$

Given that CRIC uses the constant value 10 for n, the following simplified runtime complexity applies:

$$= O(\log_2(|C|^2*|WL|^2*|WT|) + O((\log_2|WT|)+10*|C|/2)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^2*|WL|^2*|WT|^2) + O(|C|)+O(|C|*\log_2|C|)$$

The deletion of a text

The following factors determine the runtime complexity for the deletion of text:

(i) Deletion of the text, decrementing the number of words in the word index,

(ii) Deletion of the text in the inverse index of the DBMS.

With C as text base; WL and WT as data structures for the words and word-text relationships in CRIC; and |C|, |WL| and |WT| as their cardinalities, the following time complexity applies:

$$(i) \quad O(\log_2(|C|*|WL|))$$
$$(ii) \quad + O(\log_2(|C|*|WL|))+ O(\log_2|WT|^2)$$

In summary, then, the following runtime applies:

$$O(\log_2(|C|^2*|WL|^2 *|WT|^2))$$

The modification of a text

A text is modified by concatenating the ‘delete’ and ‘new text’ operations and therefore produces the following accumulated runtime:

$$O(\log_2(|C|^2*|WL|^2*|WT|^2)) +O(\log_2(|C|^2*|WL|^2*|WT|^2) +O(|C|)+O(|C|*\log_2|C|)$$
$$= O(\log_2(|C|^4*|WL|^4 *|WT|^4)) + O(|C|)+O(|C|*\log_2|C|)$$

The reduction of time complexity through approximation

If the CRIC search for related texts is expressed as a function f_n, then it is clearly not a symmetric function. Hence, an inverse function fn-1 does not exist either. As a result, f_n has to be calculated anew for each text T_{new} in C for all T_i of C, because the function value may have changed with the new text T_{new}. The runtime complexity of the operation $f_n(T_{new})$ for the recalculation of all texts $T_{i=1\ldots|C|}$ of C is:

$$O(|C|*[(n*\log_2|WT|+\Sigma_{i=1\ldots n}|C|/2)+(|C|*\log_2|C|)])$$

where n is the number of words in the query. Given that this is a constant, the following applies:

$$O(|C|*[(\log_2|WT|+|C|/2)+(|C|*\log_2|C|)])=O(|C|*\log_2|WT|+|C|^2+(|C|^2*\log_2|C|))$$
$$= O(|C|*\log_2|WT|+|C|^2+|C|^2*\log_2|C|) > O(|C|^2)$$

In order to avoid the exponential growth of the runtime complexity, an approximation for f_n for all $T_1, \ldots, T_{|C|}$ in C is required. In so doing, symmetry of f_n is assumed, and merely for $T_1, \ldots, T_n$ of the result of $f_n(T_{new})$ a substitute function f_{sub} of the missing inverse function f_n^{-1} is applied. This function is defined as follows:

. $T_{i=1\ldots|C|}, T_i \in f_n(T_{new}): f_{sub}(T_i) = f_n(T_i) \cup \{T_{new}\}$

Illustratively, T is added to texts T_i as a text with related content. The correct function $f_n(T_{i=1\ldots|C|})$ is calculated in packets of t texts while the system is idle.

A concept for high-load scenarios

For texts that are related to text T to be displayed, a query for related texts in the CRIC auxiliary table V has to be carried out. The time complexity of this operation is (the cardinality of V is $15*|C|$):

$$O(\log_2(15*|C|)+15) = O(\log_2(15*|C|))$$

In view of the low change ratio to be expected in C (when compared to the reading ratio) as well as the low probability that $f_n(T)$ for text T changes in the process, it is not sensible to carry out this query for every access of the texts that are related to text T. The time complexity can be reduced by caching the search results for text T. With p as Cache period; A_w as the number of repeated accesses of T_W (the related texts) within p; and A_n as the number of new accesses of T_e (the related texts) in p, the following result is produced:

$$O(\log_2(15*|C|))*(A_n)$$

instead of

$$O(\log_2(15*|C|))*(A_n+A_w)$$

Obviously, caching is particularly sensible an option if A_w is much larger than A_n.

Speed in real-time environments

The response time and load compatibility of the system was tested on a large German Internet portal. With approximately 320,000 texts and average page impressions of more than 50,000,000 per month, the portal is regarded as representative for high-load environments. Equally important, breaking news, promotion campaigns and newsletters give rise to significant usage peaks of up to 322 page impressions per second. The test environment consisted of the following server specification:

- CPU: Intel Dual Processor Xeon 700
- Primary memory: 2 GB
- Secondary memory: 100 GB RAID5

In 32.68% of all page impressions (approx. 16 million impressions per month on average, with the remaining impressions stemming from index pages), the recommendations generated were active for a period of 12 months. During usage peaks this ratio reached 47.31%, which corresponds to a maximum of 152.34 impressions per second of recommendations to related texts. During usage peaks the caching effects were expectedly significant. In peak hours, 41.45% of page impressions were delivered from the Cache (with Cache period p = 5 minutes). A maximum calculation ratio of 89.19 recommendations per second can be derived from this. With this calculation ratio, the CPU usage amounted to approximately 40%. A simple conversion yields up to 40 million recommendations per month as the upper limit for the given hardware.

The quality compared to manual selection

The quality of the procedure described above was evaluated by integrating it into an Internet portal for electrical engineering and life sciences. The portal is maintained by a professional publishing company and its editorial staff. The users of the portal have a high return rate, because the section of the portal relevant for the evaluation is accessed exclusively by readers of the print medium.

Every user receives between one and five editorially generated links to related texts. For purposes of evaluation, additional text recommendations of the same scope are provided that are generated through CRIC. A third and final set of links is provided that also consists of automated text recommendations, yet is based on metadata (of a hierarchical, manual structure).

The three sets of links are provided in the order ‘Editorial Staff’, ‘CRIC’, and ‘Metadata’ and are presented to the user visually in separate lists. The origin of the individual recommendations is unknown to the user. During a 12-month period the following link usage rates was achieved (links displayed as a ratio of links used, given as average values with the standard divergence [S]):

- Editorial Staff : 2.03 (S: 0.289491708)
- CRIC : 2.18 (S: 0.267472775)
- Metadata : 6.26 (S: 0.744574389)

When accounting for the preferences usually given to links at the top of a list, the value of 2.18 moves the CRIC recommendations close to the quality of the text recommendations provided by editorial staff. The clear drop in usage for the links generated on the basis of metadata reinforces this point.

13 Ausblick

Das hier vorgestellte CBF Verfahren liefert semantisch verwandte Texte zu dem gerade vom Benutzer gelesenen Text. Dabei wird nur unstrukturierter Text vorausgesetzt.

Die ausführlichen Praxistests haben die theoretischen Ansätze sowohl bezüglich der Qualität, als auch hinsichtlich der Leistungsfähigkeit in Hochlastszenarien bestätigt. Aufgrund der limitierten Laufzeitkomplexität stellen auch sehr große Textbasen und hohe Nutzungsraten kein Problem dar.

Das Verfahren ist bisher allerdings nur für kleine Texte (bis 500 Worte) getestet worden. Bei größeren Texten besteht das Problem, dass ein reiner Verweis auf den Text dem Benutzer das Auffinden der relevanten Stellen im Text überlässt. Zum anderen müssen Texte mit mehreren Themen in Form einzelner Aufsätze oder durch wechselnde Schwerpunkte im Textverlauf zunächst in logische Themen-Einheiten zerlegt werden. Für diese Problemstellungen gibt es bereits zahlreiche Lösungsansätze, die von der Typographie (Überschriften als Marker für neue Themen et cetera) bis zu Kollokationen verschiedene Merkmale der Texte verwenden, um diese zu zerlegen. Um das CRIC Verfahren auch für größere Texte nutzen zu können, muss ein solches Zerlegungsverfahren vorgeschaltet werden.

Eine Nutzung der durch CRIC ermittelten bestimmenden Worte der Texte im Rahmen eines interaktiven Query Expansion Ansatzes hat in ersten Benutzertests bereits positive Ergebnisse erzielt.

Möglichkeiten zur Weiterentwicklung des Ansatzes im Kontext einer Personalisierung auf Basis eines persistenten Profils werden derzeit untersucht.

TEIL V - ANHANG

14 Verzeichnisse

14.1 Abbildungsverzeichnis

14.2 Tabellenverzeichnis

14.3 Algorithmenverzeichnis

14.4 Literaturverzeichnis

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[AHO1974]	Aho, A.V.; Hopcroft, J.E.; Ullman J.D.	1974	The Design and Analysis of Computer Algorithms	AddisonWesley, Reading, Massachusetts, United States, 1974	-
[ALS1998]	Alspector, J.; Kolcz, A.; Karunanithi, N.	1998	Comparing feature-based and clique-based user models for movie selection	Proceedings of the third ACM conference on Digital libraries, S. 11-18, Pittsburgh, Pennsylvania, United States, 1998	The huge amount of information available in the currently evolving world wide information infrastructure at any one time can easily overwhelm end-users. One way to address the information explosion is to use an "information filtering agent" which can select information according to the interest and/or need of an end-user. However, at present few such information filtering agents exist. In this study, we evaluate the use of feature-based approaches to user modeling with the purpose of creating a filtering agent for the video-on-demand application. We evaluate several feature and clique-based models for 10 voluntary subjects who provided ratings for the movies. Our preliminary results suggest that feature-based selection can be a useful tool to recommend movies according to the taste of the user and can be as effective as a movie rating expert. We compare our feature-based approach with a clique-based approach, which has advantages where information from other users is available.
[ARM1998]	Armstrong, R.; Freitag, D.; Joachims, T.; Mitchell, T.	1998	WebWatcher: A Learning Apprentice for the World Wide Web	Machine Learning and Data Mining, R. Michalski and I. Bratko and M. Kubat (ed.), Wiley, 1998	We describe an information seeking assistant for the world wide web. This agent, called WebWatcher, interactively helps users locate desired information by employing learned knowledge about which hyperlinks are likely to lead to the target information. Our primary focus to date has been on two issues: (1) organizing WebWatcher to provide interactive advice to Mosaic users while logging their successful and unsuccessful searches as training data, and (2) incorporating machine learning methods to automatically acquire knowledge for selecting an appropriate hyperlink given the current web page viewed by the user and the user's information goal. We describe the initial design of WebWatcher, and the results of our preliminary learning experiments.
[BAD2000]	Sarwar, B.M.; Karypis, G.; Konstan, J.A.; Riedl, J.	2000	Analysis of recommendation algorithms for e-commerce	ACM Conference on Electronic Commerce, S. 158-167, Minneapolis, Minnesota, UNITED STATES, 2000	Recommender systems apply statistical and knowledge discovery techniques to the problem of making product recommendations during a live customer interaction and they are achieving widespread success in E-Commerce nowadays. In this paper, we investigate several techniques for analysing large-scale purchase and preference data for the purpose of producing useful recommendations to customers. In particular, we apply a collection of algorithms such as traditional data mining, nearest neighbor collaborative filtering, and dimensionality reduction on two different data sets. The fi

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					rst data set was derived from the web-purchasing transaction of a large E-commerce company whereas the second data set was collected from MovieLens movie recommendation site. For the experimental purpose, we divide the recommendation generation process into three sub processes -representation of input data, neighborhood formation, and recommendation generation. We devise different techniques for different sub processes and apply their combinations on our data sets to compare for recommendation quality and performance.
[BAL1997]		1997	Fab - Content-Based, Collaborative Recommendation	Communications of the ACM, vol.40, no.3, S. 66-72, 1997	-
[BAL1998]	Balabanovic, M.	1998	Learning to Surf: Multiagent Systems for Adaptive Web Page Recommendation	PhD Thesis, Stanford University Department of Computer Science, completed March 1998	Imagine a newspaper personalized for your tastes. Instead of a selection of articles chosen for a general audience by a human editor, a software agent picks items just for you, covering your particular topics of interest. Since there are no journalists at its disposal, the agent searches the Web for appropriate articles. Over time, it uses your feedback on recommended articles to build a model of your interests. This thesis investigates the design of \recommender systems" which create such personalized newspapers. Two research issues motivate this work and distinguish it from approaches usually taken by information retrieval or machine learning researchers. First, a recommender system will have many users, with overlapping interests. How can this be exploited? Second, each edition of a personalized newspaper consists of a small set of articles. Techniques for deciding on the relevance of individual articles are well known, but how is the composition of the set determined? One of the primary contributions of this research is an implemented architecture linking populations of adaptive software agents. Common interests among its users are used both to increase efficiency and scalability, and to improve the quality of recommendations. A novel interface infers document preferences by monitoring user drag-and-drop actions, and affords control over the composition of sets of recommendations. Results are presented from a variety of experiments: user tests measuring learning performance, simulation studies isolating particular tradeoffs, and usability tests investigating interaction designs.
[BAS1998]	Basu, C.; Hirsh, H.; Cohen, W.W.	1998	Recommendation as classification: Using social and content-based information in recommendation	Proceedings of the Fifteenth National Conference on Artificial Intelligence, S. 714–720, Madison, WI, United States,	Recommendation systems make suggestions about artifacts to a user. For instance, they may predict whether a user would be interested in seeing a particular movie. Social recommendation methods collect ratings of artifacts from many individuals and use nearest-neighbor techniques to make

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				1998	recommendations to a user concerning new artifacts. However, these methods do not use the significant amount of other information that is often available about the nature of each artifact --- such as cast lists or movie reviews, for example. This paper presents an inductive learning approach to recommendation that is able to use both ratings information and other forms of information about each artifact in predicting user preferences. We show that our method outperforms an existing social-filtering method in the domain of movie recommendations on a dataset of more than 45,000 movie ratings collected from a community of over 250 users.
[BAS2004]	Basilico, J.; Hofmann, T.	2004	Unifying collaborative and content-based filtering	Proceedings of the twenty-first international conference on Machine learning, Banff, Alberta, Canada, S. 9, ISBN 1581138285, 2004	Collaborative and content-based filtering are two paradigms that have been applied in the context of recommender systems and user preference prediction. This paper proposes a novel, unified approach that systematically integrates all available training information such as past user-item ratings as well as attributes of items or users to learn a prediction function. The key ingredient of our method is the design of a suitable kernel or similarity function between user-item pairs that allows simultaneous generalization across the user and item dimensions. We propose an on-line algorithm (JRank) that generalizes perceptron learning. Experimental results on the EachMovie data set show significant improvements over standard approaches.
[BAU2001]	Bauer, T.; Leake, D.B.	2001	A Research Agent Architecture for Real Time Data	Fifth International Conference on Intelligent Agents, Montreal, Canada, S. 61-66, 2001	Collecting and analyzing real-time data from multiple sources requires processes to continuously monitor and respond to a wide variety of events. Such processes are well suited to execution by intelligent agents. Architectures for such agents need to be general enough to support experimentation with various analysis techniques but must also implement enough functionality to provide a solid back end for data collection, storage, and reuse. In this paper, we present the architecture of Calvin, an agent for supporting users' document access. Calvin provides specific utilities for collecting, storing, and retrieving data to be used by information retrieval methods, but its extensible object oriented implementation of resource types makes the architecture sufficiently flexible to be useful as a research agent in multiple task domains. In addition, the architecture supports research by providing the ability to capture and ``replay'' data streams during processing, enabling the automatic creation of data testbeds that can be used in experiments for comparing alternative methodologies.
[BAU2001a]	Bauer, T.; Leake, D.B.	2001	WordSieve: A Method for Real-Time Context Extraction Collection	Lecture Notes in Computer Science, vol.2116, S. 30-43, 2001	In order to be useful, intelligent information retrieval agents must provide their users with context-relevant information. This paper presents WordSieve, an algorithm for

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			and Analysis		automatically extracting information about the context in which documents are consulted during web browsing. Using information extracted from the stream of documents consulted by the user, WordSieve automatically builds context profiles which differentiate sets of documents that users tend to access in groups. These profiles are used in a research-aiding system to index documents consulted in the current context and pro-actively suggest them to users in similar future contexts. In initial experiments on the capability to match documents to the task contexts in which they were consulted, WordSieve indexing outperformed indexing based on Term Frequency/Inverse Document Frequency. a common document indexing approach for intelligent agents in information retrieval.
[BAU2002]	Bauer, T.; Leake, D.B.	2002	Exploiting information access patterns for context-based retrieval	Proceedings of the 7th international conference on Intelligent user interfaces, San Francisco, California, United States, Session: Short Papers, S. 176-177, ISBN 1581134592, 2002	order for intelligent interfaces to provide proactive assistance, they must customize their behavior based on the user's task context. Existing systems often assess context based on a single snapshot of the user's current activities (e. g., examining the content of the document that the user is currently consulting). However, an accurate picture of the user's context may depend not only on this local information, but also on information about the user's behavior over time. This paper discusses work on a recommender system, Calvin, which learns to identify broader contexts by relating documents that tend to be accessed together. Calvin's text analysis algorithm, WordSieve, develops term vector descriptions of these contexts in real time, without needing to accumulate comprehensive statistics about an entire corpus. Calvin uses these descriptions (1) to index documents to suggest them in similar future contexts and (2) to formulate contextbased queries for search engines. Results of initial experiments are encouraging for the approach's improved ability to associate documents with the research tasks in which they were consulted, compared to methods using only local information. This paper sketches the project goals, the current implementation of the system, and plans for its continued development and evaluation.
[BAX1957]	Baxendale, P.B.	1958	Machine-made index for technical literature – an experiment	IBM journal, vol. 10, S. 354-361, 1957	Machine techniques for reducing technical documents to their essential discriminating indices are investigated. Human scanning patterns in selecting ”topic sentences“ and phrases composed of nouns and modifiers were simulated by computer program. The amount of condensation resulting from each method and the relative uniformity in indices are examined. It is shown that the coordinated index provided by the phrase is the more meaningful and discriminating.
[BER1994]	Berner-Lee, T.	1994	RFC 1630 - Universal Resource Identifiers in WWW:	Internet RFCs: http://www.faqs.org /rfcs/rfc1630.html	This document defines the syntax used by the World-Wide Web initiative to encode the names and addresses of objects on the Internet. The web is

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web		considered to include objects accessed using an extendable number of protocols, existing, invented for the web itself, or to be invented in the future. Access instructions for an individual object under a given protocol are encoded into forms of address string. Other protocols allow the use of object names of various forms. In order to abstract the idea of a generic object, the web needs the concepts of the universal set of objects, and of the universal set of names or addresses of objects. A Universal Resource Identifier (URI) is a member of this universal set of names in registered name spaces and addresses referring to registered protocols or name spaces. A Uniform Resource Locator (URL), defined elsewhere, is a form of URI which expresses an address which maps onto an access algorithm using network protocols. Existing URI schemes which correspond to the (still mutating) concept of IETF URLs are listed here. The Uniform Resource Name (URN) debate attempts to define a name space (and presumably resolution protocols) for persistent object names. This area is not addressed by this document, which is written in order to document existing practice and provide a reference point for URL and URN discussions.
[BER1999]	Berney, B.; Ferneley, E.	1999	CASMIR - a community of software agents collaborating in order to retrieve multimedia data	Proceedings of the third annual conference on Autonomous Agents, Seattle, Washington, United States, S. 428-429, 1999	This paper outlines the development and implementation of CASMIR (Collaborative Agent-based System for Multimedia Information retrieval), a Multimedia Information Retrieval (MIR) system that attempts to aid users in searching for multimedia data by using a connectionist IR model within amulti-agent framework. This IR model allows the system to provide query by reformulation, closest match searching and collaborative user profiling whilst its implementation as a community of Java software agents, communicating in KQML, provides the benefits of parallelism and scalability. A preliminary evaluation is also described, the results of which indicate that not only is the system able to learn about its users, but that collaborative user profiling can provide a significant increase in the learning rate.
[BIL1999]	Billsus, D.; Pazzani, M.J.	1999	A Personal News Agent that Talks, Learns and Explains	Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, UNITED STATES, 1999	Most work on intelligent information agents has thus far focused on systems that are accessible through the World Wide Web. As demanding schedules prohibit people from continuous access to their computers, there is a clear demand for information systems that do not require workstation access or graphical user interfaces. We present a personal news agent that is designed to become part of an intelligent, IP-enabled radio, which uses synthesized speech to read news stories to a user. Based on voice feedback from the user, the system automatically adapts to the user's preferences and interests. In addition to time-coded feedback, we explore two components of the system that facilitate the automated induction of accurate

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					interest profiles. First, we motivate the use of a multistrategy machine learning approach that allows for the induction of user models that consist of separate models for long-term and short-term interests. Second, we investigate the use of "concept feedback", a novel form of user feedback that is based on our agent's capability to construct explanations for the reasons that have led to a specific classification. Users can then critique these explanations which, from a machine learning perspective, allows for more direct changes to an induced concept than through the inclusion of additional training examples. We evaluate the proposed algorithms on user data collected with a prototype of our system, and assess the performance contributions of the system's individual components.
[BIL2005]	Billsus, D.; Hilbert, D.M.; Maynes-Aminzade, D.	2005	Improving proactive information systems	Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, United States, Session: long papers: recommendation and instruction, S.159 -166, 2005, ISBN 1581138946	Proactive contextual information systems help people locate information by automatically suggesting potentially relevant resources based on their current tasks or interests. Such systems are becoming increasingly popular, but designing user interfaces that effectively communicate recommended information is a challenge: the interface must be unobtrusive, yet communicate enough information at the right time to provide value to the user. In this paper we describe our experience with the FXPAL Bar, a proactive information system designed to provide contextual access to corporate and personal resources. In particular, we present three features designed to communicate proactive recommendations more effectively: translucent recommendation windows increase the user's awareness of particularly highly-ranked recommendations, query term highlighting communicates the relationship between a recommended document and the user's current context, and a novel recommendation digest function allows users to return to the most relevant previously recommended resources. We present empirical evidence supporting our design decisions and relate lessons learned for other designers of contextual recommendation systems
[BMI2002]	Bundesministerium für Wirtschaft und Technologie	2002	Wissensmanagement	e-F@cts - Informationen zum E-Business, Ausgabe 10/2002, S. 1-7, 2002	-
[BRO1987]	Bronstein, I.; Semendjajew, K.; Musiol, G	1987	Taschenbuch der Mathematik	Verlag Harri Deutsch	Ein Nachschlagewerk und Taschenbuch der Mathematik, wie es der Bronstein darstellt, lebt insbesondere auch dadurch, dass die Autoren von Auflage zu Auflage den sich wandelnden Anforderungen eines breiten Nutzerkreises gerecht werden und immer wieder den praxisnahen zeitgerechten Bezug sicherstellen. Das Taschenbuch enthält einen Querschnitt der Mathematik, wie er sowohl für Studenten als auch für praktisch tätige

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					Ingenieure, Naturwissenschaftler und Mathematiker sowie für die einschlägigen Hochschullehrer erforderlich ist. Dem traditionellen Anliegen des Buches - vorgegeben von den Erstautoren I. N. Bronstein und K. A. Semendjajew (1937) - folgend, stehen Anschaulichkeit und leichte Verständlichkeit im Vordergrund, während der Forderung nach mathematischer Strenge bei der gebotenen Kürze der Darstellung nur in angemessenem, für den Ingenieur und Naturwissenschaftler im allgemeinen ausreichendem Umfange Rechnung getragen wird. So sind für diesen Nutzerkreis mathematischer Sachverhalt, Voraussetzungen, Grenzen der Anwendbarkeit und Hinweise auf Besonderheiten bei Anwendungen wichtiger als strenge mathematische Beweise. Für weitergehende Bedürfnisse wird jeweils auf die Fachliteratur verwiesen.
[BUD1999]	Budzik, J.; Hammond, K.	1999	Watson: Anticipating and Contextualizing Information Needs	Proceedings of the 62nd Annual Meeting of the American Society for Information Science, 1999	In this paper, we introduce a class of systems called Information Management Assistants (IMAs). IMAs automatically discover related material on behalf of the user by serving as an intermediary between the user and information retrieval systems. IMAs observe users interact with everyday applications and then anticipate their information needs using a model of the task at hand. IMAs then automatically fulfill these needs using the text of the document the user is manipulating and a knowledge of how to form queries to traditional information retrieval systems (e.g., Internet search engines, abstract databases, etc.). IMAs automatically query information systems on behalf of users as well as provide an interface by which the user can pose queries explicitly. Because IMAs are aware of the user's task, they can augment their explicit query with terms representative of the context of this task. In this way, IMAs provide a framework for bringing implicit task context to bear on servicing explicit information requests, significantly reducing ambiguity. IMAs embody a just-in-time information infrastructure in which information is brought to users as they need it, without requiring explicit requests. In this paper, we present our work on an architecture for this class of system, and our progress implementing Watson, a prototype of such a system. Watson observes users in word processing and Web browsing applications and uses a simple model of the user's tasks, knowledge of term importance, and an understanding of query generation to find relevant documents and service explicit queries. We close by discussing our experimental evaluations of the system.
[BUD2000]	Budzik, J.; Hammond, K.J.	2000	User Interactions with Everyday Applications as Context for Just-in-time Information	Proceedings of the International Conference on Intelligent User Interfaces, New	Our central claim is that user interactions with everyday productivity applications (e.g., word processors, Web browsers, etc.) provide rich contextual information that can be leveraged to support just-in-time access to task-

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			Access	Orleans, Louisiana, United States, 2000	relevant information. We discuss the requirements for such systems, and develop a general architecture for systems of this type. As evidence for our claim, we present Watson, a system which gathers contextual information in the form of the text of the document the user is manipulating in order to proactively retrieve documents from distributed information repositories. We close by describing the results of several experiments with Watson, which show it consistently provides useful information to its users.
[BUS1945]	Bush, V.	1945	As We May Think	The Atlantic monthly, July 1945	As Director of the Office of Scientific Research and Development, Dr. Vannevar Bush has coordinated the activities of some six thousand leading American scientists in the application of science to warfare. In this significant article he holds up an incentive for scientists when the fighting has ceased. He urges that men of science should then turn to the massive task of making more accessible our bewildering store of knowledge. For years inventions have extended man's physical powers rather than the powers of his mind. Trip hammers that multiply the fists, microscopes that sharpen the eye, and engines of destruction and detection are new results, but not the end results, of modern science. Now, says Dr. Bush, instruments are at hand which, if properly developed, will give man access to and command over the inherited knowledge of the ages. The perfection of these pacific instruments should be the first objective of our scientists as they emerge from their war work. Like Emerson's famous address of 1837 on "The American Scholar," this paper by Dr. Bush calls for a new relationship between thinking man and the sum of our knowledge.
[CAR1998]	Carr, L.A.; Hall, W.; Hitchcock, S.	1998	Link services or link agents?	Proceedings of the ninth ACM conference on Hypertext and hypermedia, Pittsburgh, Pennsylvania, United States, S. 113-122, ISBN:0897919726, 1998	
[CHA1995]	Chaudhuri, S.; Shim, K.	1995	An overview of query optimization in relational systems	Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, Seattle, Washington, United States, S. 34-43, 1998	There has been extensive work in query optimization since the early ‘70s. It is hard to capture the breadth and depth of this large body of work in a short article. Therefore, I have decided to focus primarily on the optimization of SQL queries in relational database systems and present my biased and incomplete view of this field, The goal of this article is not to be comprehensive, but rather to explain the foundations and present samplings of significant work in this area. I would like to apologize to the many contributors in this area whose

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					work I have failed to explicitly knowledge due to oversight or lack of space. I take the liberty of trading technical precision for ease of presentation.
[CHE1994]	Chen, C.M.; Roussopoulos, N.	1994	Adaptive selectivity estimation using query feedback	Proceedings of the 1994 ACM SIGMOD international conference on Management of data table of contents, Minneapolis, Minnesota, United States, S. 161-172, 1994	In this paper, we propose a novel approach for estimating the record selectivities of database queries. The real attribute value distribution is adaptively approximated by a curve-fitting function using a query feedback mechanism. This approach has the advantages of requiring no extra database access overhead for gathering statistics and of being able to continuously adapt the value distribution through queries and updates. Experimental results show that the estimation accuracy of this approach is comparable to traditional methods based on statistics gathering.
[CHE1997]	Cheng, J.		Learning Belief Networks from Data: An Information Theory Based Approach	Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, CIKM'97, 1997	This paper presents an efficient algorithm for learning Bayesian belief networks from databases. The algorithm takes a database as input and constructs the belief network structure as output. The construction process is based on the computation of mutual information of attribute pairs. Given a data set that is large enough, this algorithm can generate a belief network very close to the underlying model, and at the same time, enjoys the time complexity of $O(N^4)$ on conditional independence (CI) tests. When the data set has a normal DAG-Faithful probability distribution, the algorithm guarantees that the structure of a perfect map of the underlying dependency model is generated. To evaluate this algorithm, we present the experimental results on three versions of the well-known ALARM network database, which has 37 attributes and 10,000 records. The results show that this algorithm is accurate and efficient. The proof of correctness and the analysis of computational complexity are also presented.
[CHE1998]	Chen, L.; Sycara, K.	1998	WebMate: a personal agent for browsing and searching	Proceedings of the second international conference on Autonomous agents table of contents, Minneapolis, Minnesota, United States, S. 132-139, 1998	The World-Wide Web is developing very fast. Currently, finding useful information on the Web is a time consuming process, In this paper, we present WebMate, an agent that helps users to effectively browse and search the Web. WebMatee extends the state of the art in Web-based information retrieval in many ways. First, it uses multiple TF-IDF vectors to keep track of user interests in different domains. These domains are automatically learned by WebMate. Second, WebMate uses the Trigger Pair Model to automatically extract keywords for refining document search. Third, during search, the user can provide multiple pages as similarity/ relevance guidance for the search. The system extracts and combines relevant keywords from these relevant pages and uses them for keyword refinement. Using these techniques, WebMate provides effective browsing and searching help and also compiles and sends to users personal newspaper by

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					automatically spiding news sources. We have experimentally evaluated the performance of the system.
[CHE1999]	Cheng, J. and Greiner, R.	1999	Comparing Bayesian Network Classifiers	Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99), S. 101-108, Sweden, 1999	In this paper, we empirically evaluate algorithms for learning four Bayesian network (BN) classifiers: Naïve-Bayes, tree augmented Naïve-Bayes (TANs), BN augmented NaïveBayes (BANs) and general BNs (GBNs), where the GBNs and BANs are learned using two variants of a conditional independence based BN-learning algorithm. Experimental results show the GBNs and BANs learned using the proposing learning algorithms are competitive with (or superior to) the best classifiers based on both Bayesian networks and other formalisms, and that the computational time for learning and using these classifiers is relatively small. These results argue that BN classifiers deserve more attention in machine learning and data mining communities.
[CHE2000]	Chen, Z.; Meng, X.	2000	WebSail: From On-line Learning to Web Search	Web Information Systems Engineering, S. 206-213, 2000	In this paper we investigate the applicability of on-line learning algorithms to the real-world problem of web search. Consider that web documents are indexed using Boolean features. We first present a practically efficient on-line learning algorithm TW2 to search for web documents represented by a disjunction of at most k relevant features. We then design and implement WebSail, a real-time adaptive web search learner, with TW2 as its learning component. WebSail learns from the user's relevance feedback in real-time and helps the user to search for the desired web documents. The architecture and performance of WebSail are also discussed.
[CHEn1998]	Chen, C.M.; Roussopoulos, N.	1998	Adaptive selectivity estimation using query feedback	Proceedings of the 1994 ACM SIGMOD international conference on Management of data, Minneapolis, Minnesota, United States, S. 161-172, 1994	In this paper, we propose a novel approach for estimating the record selectivities of database queries. The real attribute value distribution is adaptively approximated by a curve-fitting function using a query feedback mechanism. This approach has the advantages of requiring no extra database access overhead for gathering statistics and of being able to continuously adapt the value distribution through queries and updates. Experimental results show that the estimation accuracy of this approach is comparable to traditional methods based on statistics gathering.
[CHU1989]	Church, K.W.; Hanks, P.	1989	Word Association Norms, Mutual Information, and Lexicography	Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics, Vancouver, B.C., Canada, S. 76-83, 1989	The term word association is used in a very particular sense in the psycholinguistic literature. (Generally speaking, subjects respond quicker than normal to the word "nurse" if it follows a highly associated word such as "doctor.") We wilt extend the term to provide the basis for a statistical description of a variety of interesting linguistic phenomena, ranging from semantic relations of the doctor/nurse type (content word/content word) to lexico-syntactic co-occurrence constraints between verbs and prepositions (content word/function word). This paper will propose a new

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					objective measure based on the information theoretic notion of mutual information, for estimating word association norms from computer readable corpora. (The standard method of obtaining word association norms, testing a few thousand subjects on a few hundred words, is both costly and unreliable.) The proposed measure, the association ratio, estimates word association norms directly from computer readable corpora, making it possible to estimate norms for tens of thousands of words.
[CLA1999]	Claypool, M.; Gokhale, A.; Miranda, T.; Murnikov P.; Netes D.; Sartin, M.	1999	Combining Content-Based and Collaborative Filters in an Online Newspaper	Proceedings of ACM SIGIR Workshop on Recommender Systems, Berkeley, UNITED STATES, 1999	The explosive growth of mailing listes, Web sites and Usenet news demands effective filtering solutions. Collaborative filtering combines the informed opinions of humans to make personalized, accurate predictions. Content-based filtering uses the speed of computers to make complete, fast predictions. In this work, we present a new filtering approach that combines the coverage and speed of content-filters with the depth of collaborative filtering. We apply our research approach to an online newspaper, an as yet untapped opportunity for filters useful to the wide-spread news reading populace. We present the design of our filtering system and describe the results from preliminary experiments that suggest merits to our approach.
[CLE1967]	Cleverdon, C.W.	1967	The Cranfield tests on index language devices	Aslib Proceedings, 19, 6, S. 173-194	-
[COD1970]	Codd, E.F.	1970	A Relational Model of Data for Large Shared Data Banks	Communications of the ACM, vol.13, no 6, S. 377-387, 1970	This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question - answering systems. Levein and Maron [2] provide numerous references to work in this area. In contrast, the problems treated here are those of data independence - the independence of application programs and terminal activities from growth in data types and changes in data representation N and certain kinds of data inconsistency which are expected to become troublesome even in nondeductive systems. The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only -- that is, without superimposing any additional structure for machine representation poses. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other. A further advantage of the relational view is that it forms a sound basis for treating derivability,

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					redundancy, and consistency of relations - these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap"). Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.
[COH1995]	Cohen, W.W.	1995	Fast Effective Rule Induction	Proceedings of the 12th International Conference on Machine Learning, S. 115-123, Tahoe City, CA, United States, ISBN 1558603778, 1995",	Many existing rule learning systems are computationally expensive on large noisy datasets. In this paper we evaluate the recently-proposed rule learning algorithm IREP on a large and diverse collection of benchmark problems. We show that while IREP is extremely efficient, it frequently gives error rates higher than those of C4.5 and C4.5rules. We then propose a number of modifications resulting in an algorithm RIPPERk that is very competitive with C4.5rules with respect to error rates, but much more efficient on large samples. RIPPERk obtains error rates lower than or equivalent to C4.5rules on 22 of 37 benchmark problems, scales nearly linearly with the number of training examples, and can efficiently process noisy datasets containing hundreds of thousands of examples.
[COV1967]	Cover, T.M.; Hart, P. E.	1967	Nearest Neighbor Pattern Classification	IEEE Transactions on Information Theory, vol. IT-IS, nO.1, 1967	The nearest neighbor decision rule assigns to an unclassified sample point the classification of the nearest of a set of previously classified points. This rule is independent of the underlying joint distribution on the sample points and their classifications, and hence the probability of error R of such a rule must be at least as great as the Bayes probability of error R*--the minimum probability of error over all decision rules taking underlying probability structure into account. However, in a large sample analysis, we will show in the M-category case that R* < R < R*(Z - MR*/(M-I)), where these bounds are the tightest possible, for all suitably smooth underlying distributions. Thus for any number of categories, the probability of error of the nearest neighbor rule is bounded above by twice the Bayes probability of error. In this sense, it may be said that half the classification information in an insuite sample set is contained in the nearest neighbor.
[CRO1997]	Croft, W.B., Harper, D.J.	1997	Using probabilistic models of document retrieval without relevance information	Readings in information retrieval archive, S. 339-344, ISBN:1-55860-454-5, 1997	-

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[DAD1996]	Dadam, P.	1996	Datenbanken in Rechnernetzen	Unterlagen zu Kurs 1666 des Fachbereichs Informatik der Fernuniversität GHS Hagen, 1996	Verteilte Datenbanken sind Datenbanken,die physisch zwar auf mehrere, unabhängige Rechner verteilt sind, für den Benutzer jedoch wie eine (zentrale) Datenbank in Erscheinung treten. Der Kurs vermittelt einen Überblick über die hierbei auftretenden Probleme und Lösungsansätze. Im Einzelnen wird auf Rechnernetze, auf System- und Schema-Architektur, auf Aspekte der Daten- und Verteilungsunabhängigkeit, auf die Fragen der optimalen Datenverteilung sowie auf Synchronisations- und Recovery-Aspekte in diesem Kontext eingegangen werden.
[DAV2002]	Davison, B.D.	2002	Predicting Web Actions from HTML Content	Proceedings of The Thirteenth ACM Conference on Hypertext and Hypermedia , College Park, Maryland, United States, 2002	Most proposed Web prefetching techniques make predictions based on the historical references to requested objects. In contrast, this paper examines the accuracy of predicting a user's next action based on analysis of the content of the pages requested recently by the user. Predictions are made using the similarity of a model of the user's interest to the text in and around the hypertext anchors of recently requested Web pages. This approach can make predictions of actions that have never been taken by the user and potentially make predictions that reflect current user interests. We evaluate this technique using data from a full-content log of Web activity and that textual similarity-based predictions outperform simpler approaches.
[DEU2005]	Die Deutsche Bibliothek	2005	Normdaten-CD-ROM (Sprachwortnormdatei "SWD")	Die Deutschen Bibliothek, CD-ROM, 2005	Der Deutschen Bibliothek in Kooperation mit dem Bibliotheksverbund Bayern (BVB) dem Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (HBZ) dem Südwestdeutschen Bibliotheks-verbund (SWB) dem Gemeinsamen Bibliotheksverbund der Länder Bremen, Hamburg, Mecklenburg-Vorpommern, Niedersachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen (GBV) dem Kooperativen Bibliotheksverbund Berlin-Brandenburg (KOBV) dem Verbund der Wissenschaftlichen Bibliotheken Österreichs der Schweizerischen Landesbibliothek (SLB) dem Kunstbibliotheken-Fachverbund Florenz-München-Rom und dem Hessischen Bibliotheks-Informationssystem (HeBIS)
[EDM1969]	Edmundson, H.P.	1969	New methods in automatic extracting	Journal of the ACM, vol. 16, no. 2, S. 264-285, 1969	This paper describes new methods of automatically extracting documents for screening purposes, i.e. the computer selection of sentences having the greatest potential for conveying to the reader the substance of the document. While previous work has focused on one component of sentence significance, namely, the presence of high-frequency content words (key words), the methods described here also treat three additional components: pragmatic words (cue words); title and heading words; and structural indicators (sentence location). The research has resulted in an operating

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					system and a research methodology. The extracting system is parameterized to control and vary the influence of the above four components. The research methodology includes procedures for the compilation of the required dictionaries, the setting of the control parameters, and the comparative evaluation of the automatic extracts with manually produced extracts. The results indicate that the three newly proposed components dominate the frequency component in the production of better extracts.
[ENC2005]	-	2005	Ockham's razor	Encyclopaedia Britannica	Ockham's razor : also spelled Occam's razor, also called law of economy, or law of parsimony, principle stated by William of Ockham (1285–1347/49), a scholastic, that Pluralitas non est ponenda sine necessitate; "Plurality should not be posited without necessity." The principle gives precedence to simplicity; of two competing theories, the simplest explanation of an entity is to be preferred.
[EUK2003]	EU-Kommission	2003	Definition für KMU	L 124 (2003), S. 36-41 der EU-Kommission vom 6. Mai 2003; Aktenzeichen K(2003) 1422, 2003	-
[FEL1998]	Fellbaum, C.	1998	WordNet. An electronic lexical database.	Cambridge, MA: MIT Press;. 422 Seiten, 1998	This is a landmark book. For anyone interested in language, in dictionaries and thesauri, or natural language processing, the introduction, Chapters 1-4, and Chapter 16 are must reading. (Select other chapters according to your special interests; see the chapter-by-chapter review). These chapters provide a thorough introduction to the preeminent electronic lexical database of today in terms of accessibility and usage in a wide range of applications. But what does that have to do with digital libraries? Natural language processing is essential for dealing efficiently with the large quantities of text now available online: fact extraction and summarization, automated indexing and text categorization, and machine translation. Another essential function is helping the user with query formulation through synonym relationships between words and hierarchical and other relationships between concepts. WordNet supports both of these functions and thus deserves careful study by the digital library community.
[FEN2003]	Fensel, D.	2003	Spinning the semantic web	MIT-Press, ISBN 0262062321, 2003	As the World Wide Web continues to expand, it becomes increasingly difficult for users to obtain information efficiently. Because most search engines read format languages such as HTML or SGML, search results reflect formatting tags more than actual page content, which is expressed in natural language. Spinning the Semantic Web describes an exciting new type of hierarchy and standardization that will replace the current "Web of links" with a "Web of

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					meaning." Using a flexible set of languages and tools, the Semantic Web will make all available information - display elements, metadata, services, images, and especially content - accessible. The result will be an immense repository of information accessible for a wide range of new applications. This first handbook for the Semantic Web covers, among other topics, software agents that can negotiate and collect information, markup languages that can tag many more types of information in a document, and knowledge systems that enable machines to read Web pages and determine their reliability. The truly interdisciplinary Semantic Web combines aspects of artificial intelligence, markup languages, natural language processing, information retrieval, knowledge representation, intelligent agents, and databases.
[FER2002]	Ferman, A.M.; Errico, J.H.; Beek, P.v.; Sezan, M.I.	2002	Content-Based Filtering and Personalization Using Structured Metadata	Proceedings of the second ACM/IEEE-CS joint conference on Digital libraries, S. 393, Portland, Oregon, UNITED STATES, 2002.	Structured descriptions of multimedia content and automatically generated user profiles are used to filter content.
[FIR1957]	Firth, J.R.	1957	A synopsis of linguistic theory, 1930-1955	In Palmer, F.R. (Herausgeber), Selected papers of J.R. Firth 1952-1959, Harlow: Longman, 1957	-
[FIS1991]	Fischer, G.; Stevens, C.	1991	INFOSCOPE - Information access in complex poorly structured information spaces	Conference on Human Factors in Computing Systems , Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology New Orleans, Louisiana, United States , S. 63–70, 1991	Large information spaces present several problems including information overload. This research effort focuses on the domain of Usenet News, an open access computer-based bulletin board system that distributes messages and software. A conceptual framework is developed that shows the need for (a) flexible organization of information access interfaces and (b) personalized structure to deal with vocabulary mismatches. An operational innovative system building effort (INFO SCOPE) instantiates the framework. In INFOSCOPE, users can evolve the predefine system structure to suit their own semantic interpretations. The approach taken by IN FOSCOPE differs from other approaches by requiring less up-front structuring by message senders.
[FRE1998]	Freeze, W.S.	1998	The SQL Programmer's Reference	Ventana Communications Group, ISBN 1566047609, 1998	-
[FU2000]	Fu, W.; Budzik, J.; Hammond, K.J.	2000	Mining Navigation History for Recommendation	In Proceedings of the international conference for intelligent user interfaces, S.106-112, New Orleans,	Although a user's navigation history contains a lot of hidden information about the relationship between web pages and between users, this information is usually not exploited. The information hidden in the history can be an invaluable source of knowledge in

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				Louisiana, United States, 2000	assisting a user to better surf the Web. We presented a system which actively monitors and tracks a user's navigation. Once a user's navigation history is captured, we apply data mining techniques to discover the hidden knowledge contained in the history. The knowledge is then used to suggest potentially interesting web pages to users.
[FUN2001]	Funakoshi, K. and Takeshi Ohguro	2001	Evaluation of integrated content-based collaborative filtering	ACM SIGIR Workshop on Recommender Systems, New Orleans, UNITED STATES	In this paper, we present a system evaluation of a content-based collaborative information filtering method called Nakif, which uses user evaluation data and item content data together. Although Nakif is originally an incremental collaborative filtering approach that modifies user pro les while obtaining evaluations, we additionally introduce a non-incremental pro le construction strategy. Both the incremental and non-incremental methods are measured by using the MovieLens dataset, which includes evaluation data by real users to real movies. The results show that the content-based collaborative filtering approach can provide information items efficiently.
[GAL1993]	Gale, W.A.; Church, K.W.; Yarowsky, D.	1993	A method for disambiguating word senses in a large corpus.	Computers and the Humanities, Vol.26, S. 415-439, 1993	-
[GAN1996]	Ganguly, S.; Gibbons, P.B.; Matias, Y.; Silberschatz ,A.	1996	Bifocal sampling for skew-resistant join size estimation	Proceedings of the 1996 ACM SIGMOD international conference on Management of data table of contents, S. 271 – 281, Montreal, Quebec, Canada, 1996	This paper introduces bifocal sampling, a new technique for estimating the size of an equi-join of two relations. Bifocal sampling classifies tuples in each relation into two groups, sparse and dense, based on the number of tuples with the same join value. Distinct estimation procedures are employed that focus on various combinations for joining tuples (e.g., for estimating the number of joining tuples that are dense in both relations). This combination of estimation procedures overcomes some well-known problems in previous schemes, enabling good estimates with no a priori knowledge about the data distribution. The estimate obtained by the bifocal sampling algorithm is proven to lie with high probability within a small constant factor of the actual join size, regardless of the skew, as long as the join size is O(n lg n), for relations consisting of n tuples. The algorithm requires a sample of size at most O(sqr(n) lg n). By contrast, previous algorithms using a sample of similar size may require the join size to be O(sqr(n) lg n) to guarantee an accurate estimate. Experimental results support the theoretical claims and show that bifocal sampling is practical and effective.
[GAO2000]	Gao, Q.; Li, M.; Vitányi, P.	2000	Applying MDL to learn best model	Artificial Intelligence,	The Minimum Description Length (MDL) principle is solidly based on a provably ideal method of inference

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			granularity	vol.121, no.2, S. 1-29, 2000	using Kolmogorov complexity. We test how the theory behaves in practice on a general problem in model selection: that of learning the best model granularity. The performance of a model depends critically on the granularity, for example the choice of precision of the parameters. Too high precision generally involves modeling of accidental noise and too low precision may lead to confusion of models that should be distinguished. This precision is often determined ad hoc. In MDL the best model is the one that most compresses a two-part code of the data set: this embodies "Occam's Razor". In two quite different experimental settings the theoretical value determined using MDL coincides with the best value found experimentally. In the first experiment the task is to recognize isolated handwritten characters in one subject's handwriting, irrespective of size and orientation. Based on a new modification of elastic matching, using multiple prototypes per character, the optimal prediction rate is predicted for the learned parameter (length of sampling interval) considered most likely by MDL, which is shown to coincide with the best value found experimentally. In the second experiment the task is to model a robot arm with two degrees of freedom using a three layer feed-forward neural network where we need to determine the number of nodes in the hidden layer giving best modeling performance. The optimal model (the one that extrapolizes best on unseen examples) is predicted for the number of nodes in the hidden layer considered most likely by MDL, which again is found to coincide with the best value found experimentally.
[GIG1999]	Gigerenzer, G.; Todd, P.M.	1999	Simple Heuristics that make us smart	Oxford University Press, New York, United States	-
[GLU2000]	Glück, H.	2000	Metzler-Lexikon Sprache	Metzler, Stuttgart, ISBN 347601519X, 2000	-
[GOL1992]	Goldberg, D.; Nichols, D.; Oki, B. M.; Terry, D.	1992	Using collaborative filtering to weave an information tapestry.	Communications of the ACM, vol. 35, no. 12, S. 61-70, 1992	-
[GOL2000]	Goldberg, K.; Roeder, T.; Gupta, D.; Perkins, C.	2000	Eigentaste: A Constant Time Collaborative Filtering Algorithm	Information Retrieval, vol.4, no.2, Kluwer Academic Publishers, S. 133-151, 2001	Eigentaste is a collaborative filtering algorithm that uses universal queries to elicit real-valued user ratings on a common set of items and applies principal component analysis (PCA) to the resulting dense subset of the ratings matrix. PCA facilitates dimensionality reduction for offline clustering of users and rapid computation of recommendations. For a database of n users, standard nearest-neighbor techniques require O(n) processing time to compute recommendations, whereas Eigentaste requires O(1) (constant) time. We compare Eigentaste to alternative algorithms using data from Jester, an

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					online joke recommending system. Jester has collected approximately 2,500,000 ratings from 57,000 users. We use the Normalized Mean Absolute Error (NMAE) measure to compare performance of different algorithms. In the Appendix we use Uniform and Normal distribution models to derive analytic estimates of NMAE when predictions are random. On the Jester dataset, Eigentaste computes recommendations two orders of magnitude faster with no loss of accuracy. Jester is online at: http://eigentaste.berkeley.edu
[GOL2003]	Golubchik, S.	2003	MySQL Fulltext Search	International PHP Conference, Frankfurt/Main, Germany, 2003	-
[GOO1999]	Good, N.; Schafer, J.B.; Konstan, J.; Borchers, A.; Sarwar, B.; Herlocker, J.; Riedl, J.	1999	Combining Collaborative Filtering with Personal Agents for Better Recommendations	Proceedings of the 1999 Conference of the American Association of Artificial Intelligence (AAAI-99), S. 439-446	Information filtering agents and collaborative filtering both attempt to alleviate information overload by identifying which items a user will find worthwhile. Information filtering (IF) focuses on the analysis of item content and the development of a personal user interest profile. Collaborative filtering (CF) focuses on identification of other users with similar tastes and the use of their opinions to recommend items. Each technique has advantages and limitations that suggest that the two could be beneficially combined. This paper shows that a CF framework can be used to combine personal IF agents and the opinions of a community of users to produce better recommendations than either agents or users can produce alone. It also shows that using CF to create a personal combination of a set of agents produces better results than either individual agents or other combination mechanisms. One key implication of these results is that users can avoid having to select among agents; they can use them all and let the CF framework select the best ones for them.
[GRE1998]	Green, S.J.	1998	Automated link generation: can we do better than term repetition?	Proceedings of the seventh international conference on World Wide Web 7, Brisbane, Australia, S.75-84, ISSN 01697552, 1998	Most current automatic hypertext generation systems rely on term repetition to calculate the relatedness of two documents. There are well-recognized problems with such approaches, most notably, they are vulnerable to the linguistic effects of synonymy (many words for the same concept) and polysemy (many concepts for the same word). I propose a novel method for automatic hypertext generation that is based on a technique called lexical chaining, a method for discovering sets of related words in a text. I will also present the results of an empirical study designed to test this method in the context of a question answering task from a database of newspaper articles.
[GRE2000]	Greisdorf, H.	2000	Relevance: An Interdisciplinary and Information Science Perspective	Informing Science, Vol. 3, S. 67-71, 2000	Although relevance has represented a key concept in the field of information science for evaluating information retrieval effectiveness, the broader context established by interdisciplinary

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					frameworks could provide greater depth and breadth to on-going research in the field. This work provides an overview of the nature of relevance in the field of information science with a cursory view of how cross-disciplinary approaches to relevance could represent avenues for further investigation into the evaluative characteristics of relevance as a means for enhanced understanding of human information behavior.
[GRU2005]	Grünwald, P. D.	2005	A tutorial introduction to the minimum description length principle	Chapter 1 and 2 in the collection Advances in Minimum Description Length: Theory and Applications (edited by P. Grünwald, I.J. Myung, M. Pitt), MIT Press, 2005.	This tutorial provides an overview of and introduction to Rissanen's Minimum Description Length (MDL) Principle. The first chapter provides a conceptual, entirely non-technical introduction to the subject. It serves as a basis for the technical introduction given in the second chapter, in which all the ideas of the first chapter are made mathematically precise. This tutorial will appear as the first two chapters in the collection Advances in Minimum Description Length: Theory and Applications [Grünwald, Myung, and Pitt 2004], to be published by the MIT Press.
[GUE1990]	Güting, R.H.	1990	Datenstrukturen	Unterlagen zu Kurs 1663 des Fachbereichs Informatik der Fernuniversität GHS Hagen, 1990	Effiziente Algorithmen und Datenstrukturen bilden ein zentrales Thema der Informatik. Algorithmen sind Methoden zum Lösen von Problemen. Ein Datentyp ist eine Menge von Objekten zusammen mit Operationen auf diesen Objekten; eine Datenstruktur realisiert einen Datentyp, indem sie eine Repräsentation für die Objekte und Algorithmen für die Operationen anbietet. In diesem Kurs werden grundlegende Algorithmen und Datenstrukturen der Informatik behandelt; im Vordergrund steht dabei jeweils die Analyse der entstehenden Kosten (Laufzeit und Speicherplatzbedarf). Gliederung: Programmiersprachliche Konzepte für Datenstrukturen, grundlegende Datentypen (Listen, Stacks, Queues, Bäume), Datentypen zur Darstellung von Mengen (u.a. Hashing, binäre Suchbäume, AVL-Bäume), Sortieralgorithmen, Graphen, Graph-Algorithmen, geometrische Algorithmen, externes Suchen und Sortieren.
[HAA1992]	Haas, P.J.; Swami, A.N.	1992	Sequential sampling procedures for query size estimation	Proceedings of the 1992 ACM SIGMOD international conference on Management of data, San Diego, California, United States, S. 341-350, 1992	We provide a procedure, based on random sampling, for estimation of the size of a query result. The procedure is sequential in that sampling terminates after a random number of steps according to a stopping rule that depends upon the observations obtained so far. Enough observations are obtained so that, with a pre-specified probability, the estimate differs from the true size of the query result by no more than a prespecified amount. Unlike previous sequential estimation procedures for queries, our procedure is asymptotically efficient and requires no ad hoc pilot sample or a a priori assumptions about data characteristics. In addition to establishing the asymptotic properties of the estimation procedure, we provide

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					techniques for reducing undercoverage at small sample sizes and show that the sampling cost of the procedure can be reduced through stratified sampling techniques.
[HAN1997]	Hand, T.F.	1997	A proposal for task-based evaluation of text summarization systems	ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spanien, S. 31-36, 1997	Evaluation is a key part of any research and development effort, but the goals and focus of evaluations are often narrow in scope, addressing a specific algorithm or technique, or analyzing a single result All of the evaluation work clone to date on text summarization systems has been by the developers of individual systems, usually to study and improve sentence selection criteria Under TIPSTER III, DARPA is sponsoring a task-based evaluation of multiple text summarization systems This focus of this evaluation wall be on user needs, and the feasibility of applying summarization technology to a variety of task.
[HEL1993]	Hellerstein, J.M.; Stonebraker, M.	1993	Predicate Migration: Optimizing Queries with Expensive Predicates	Proceedings of the 1993 ACM SIGMOD international conference on Management of data, Washington, D.C., United States, S. 267-276, 1993	The traditional focus of relational query optimization schemes has been on the choice of join methods and join orders. Restrictions have typically been handled in query optimizers by “predicate pushdown” rules, which apply restrictions in some random order before as many joins as possible. These rules work under the assumption that restriction is essentially a zero-time operation. However, today's extensible and object-oriented database systems allow users to define time-consuming functions, which may be used in a query's restriction and join predicates. Furthermore, SQL has long supported subquery predicates, which may be arbitrarily time-consuming to check. Thus restrictions should not be considered zero-time operations, and the model of query optimization must be enhanced. In this paper we develop a theory for moving expensive predicates in a query plan so that the total cost of the plan — including the costs of both joins and restrictions — is minimal. We present an algorithm to implement the theory, as well as results of our implementation in POSTGRES. Our experience with the newly enhanced POSTGRES query optimizer demonstrates that correctly optimizing queries with expensive predicates often produces plans that are orders of magnitude faster than plans generated by a traditional query optimizer. The additional complexity of considering expensive predicates during optimization is found to be manageably small.
[HER2000]	Herlocker, J.L.; Konstan, J.; A., Riedl , J.	2000	Explaining Collaborative Filtering Recommendations	Proceedings of the 2000 ACM conference on Computer supported cooperative work, S. 241-250, Philadelphia, PA., UNITED STATES,	Automated collaborative filtering (ACF) systems predict a person’s affinity for items or information by connecting that person’s recorded interests with the recorded interests of a community of people and sharing ratings between likeminded persons. However, current recommender systems are black boxes, providing no transparency into the working of the recommendation. Explanations provide that transparency,

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				2000	exposing the reasoning and data behind a recommendation. In this paper, we address explanation interfaces for ACF systems – how they should be implemented and why they should be implemented. To explore how, we present a model for explanations based on the user's conceptual model of the recommendation process. We then present experimental results demonstrating what components of an explanation are the most compelling. To address why, we present experimental evidence that shows that providing explanations can improve the acceptance of ACF systems. We also describe some initial explorations into measuring how explanations can improve the filtering performance of users.
[HEY1989]	Hey, J.B.	1989	System and method of predicting subjective reactions	United States Patent 4.870.579, 1989	-
[HEY2005]	Heyer, Gerhard	2005	Folien zur Vorlesung "Computerlinguistik"	"http://www.asv.informatik.uni-leipzig.de/lehre/ws-0506/H-CL-Praktikum-Aufgabe5-ws0506.pdf", Stand vom 2006-07-09	-
[HUL1996]	Hull, D.A.	1996	Stemming Algorithms A Case Study for Detailed Evaluation	Journal of the American Society of Information Science, vol.47, no.1, S.70-84, 1996	The majority of information retrieval experiments are evaluated by measures such as average precision and average recall. Fundamental decisions about the superiority of one retrieval technique over another are made solely on the basis of these measures. We claim that average performance figures need to be validated with a careful statistical analysis and that there is a great deal of additional information that can be uncovered by looking closely at the results of individual queries. This paper is a case study of stemming algorithms which describes a number of novel approaches to evaluation and demonstrates their value.
[IDE1998]	Ide, N.; Véronis, J.	1998	Word Sense Disambiguatio: The State of the Art	Computational Linguistics, Vol. 24, S.1-40, 1998	The automatic disambiguation of word senses has been an interest and concern since the earliest days of computer treatment of language in the 1950's. Sense disambiguation is an "intermediate task" (Wilks and Stevenson, 1996) which is not an end in itself, but rather is necessary at one level or another to accomplish most natural language processing tasks.
[IOA1995]	Ionnidis, Y., Viswanath, P.V.	1995	Histogram-Based Solutions to diverse Database Estimitaion Problems.	IEEE Data Engineering Bulletin, vol.18, no.3, September 1995, S. 10-18	Many current database systems use some form of histograms to approximate the frequency distribution of values in the attributes of relations and based on them estimate some query result sizes and access plan costs. In this paper, we overview the line of research on histograms that we have followed at the Univ. of Wisconsin. Our goal has been to identify classes of histograms that combine three features in most realistic cases: (i) they produce estimates with

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					small errors, (ii) they are inexpensive to construct, use, and maintain, and (iii) they can be used for many diverse estimation problems. Based on that goal, we present several results, which eventually point towards a class of histograms that are practical, close to optimal, and effective in estimating sizes of query results, frequency distributions of attribute values in query results, and even costs of accesses using secondary indices.
[JAC1992]	Jacobs, K.	1992	Diskrete Stochastik	Unterlagen zu Kurs 1174 des Fachbereichs Mathematik der Fernuniversität GHS Hagen, 1992	-
[JOA1995]	Joachims, T.; Mitchell, T.; Freitag, D.; Armstrong,R.	1995	WebWatcher: Machine Learning and Hypertext	AAAI Spring Symposium on Information Gathering, 1995	This paper describes the first implementation of WebWatcher, a Learning Apprentice for the World Wide Web. We also explore the possibility of extracting information from the structure of hypertext. We introduce an algorithm which identifies pages that are related to a given page using only hypertext structure. We motivate the algorithm by using the Minimum Description Length principle.
[JOA1997]	Joachims, T.; Freitag, D.; Mitchell, T.	1997	WebWatcher: A Tour Guide for the World Wide Web	Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufmann, 1997	We explore the notion of a tour guide software agent for assisting users browsing the World Wide Web A Web tour guide agent provides assistance similar to that provided by a human tour guide in a museum it guides the user along an appropriate path through the collection based on its knowledge of the users interests of the location and relevance of various items in the collection and of the way in which others have interacted with the collection in the past. This paper describes a simple but operational tour guide called WebWatcher which has given over tours to people browsing CMUs School of Computer Science Web pages WebWatcher accompanies users from page to page suggests appropriate hyperlinks and learns from experience to improve its advice giving skills. We describe the learning algorithms used by WebWatcher experimental results showing their effectiveness and lessons learned from this case study in Web tour guide agents.
[JOH1993]	Johnson, T.; Sasha, D.	1993	The performance of current B-tree algorithms	ACM Transactions on Database Systems (TODS), vol.18, no.1 (Mar 1993), S. 51-101, 1993	Many concurrent B-tree algorithms have been proposed, but their performances have not yet been analyzed satisfactorily. When transaction processing systems require high levels of concurrency, a restrictive serialization technique on the B-tree index can cause a bottleneck. In this paper we present a framework for constructing analytical performance models of concurrent B-tree algorithms, The models can predict the response time and maximum throughput. We analyze a variety of locking algorithms including naive lock-coupling, optimistic descent, two-phase locking, and the Lehman–Yao

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					algorithm. The analyses are validated by simulations of the algorithms on actual B-trees, as well as by simulations done by other researchers. We find that the Lehman–Yao algorithm has the best performance by far, that the performance of two-phase locking is limited, and that the response time of two-phase locking has a high variance. Simple and instructive rules of thumb for predicting performance are also derived. We apply the analyses to determine the effect of database recovery on B-tree concurrency. We find that holding nonleaf locks for recovery purposes significantly reduces performance.
[JOR1990]	Jorgensen, J.	1990	The Psychological Reality of Word Senses	Journal of Psychalinguistic Research, Vol.19, S.167-190, 1990.	-
[KAN2000]	Kantrowitz, M.; Mohit, B.; Mittal, V.	2000	Stemming and its effects on TFIDF Ranking.	Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 357-359, Athens, Greece, 2000.	High precision IR is often hard for a variety of reasons; one of these is the large number of morphological variants for any given term. To address some of the issues arising from a mismatch between different word forms used in the queries and the relevant documents, researchers have long proposed the use of various stemming algorithms to reduce terms to a common base form. Stemming, in general, has not been an unmitigated success in improving IR. This poster argues that stemming can help in certain contexts and that an empirical investigation of the relationship between stemming performance and retrieval performance can be valuable. We extend previously reported work on stemming and IR (e.g., [2,4,5]) by using a novel, dictionary based "perfect" stemmer, which can be parameterized for different accuracy and coverage levels. This allows us to measure changes in IR performance for any given change in stemming performance on a given data set. To place this work in context, we discuss an empirical evaluation of stemming accuracy for three stemming algorithms - including the widely used Porter algorithm [9]. Section 2 briefly discusses the three variants of stemming, and presents experimental evidence for their relative coverage and accuracy. Section 3 discusses the use of these stemmers for IR and presents some of our findings. Finally, Section 4 concludes with a discussion of our results and possible future directions.
[KLA1993]	Klahold, A.	1993	Client/Server-Datenbanken	Vogel Verlag, Würzburg, 100 Seiten, ISBN 380231189, 1993	-
[KLA1995]	Klahold, A., von Harlessem, M., Janoschek, J.	1995	Datenbanken 1995	Vogel Verlag, Würzburg, 82 Seiten, ISBN 3825913481, 1995	-
[KON1997]	Konstan, J.A.;	1997	Grouplens: Applying	Communications of	This article discusses the challenges involved in creating a collaborative

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
	Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J.		collaborative filtering to Usenet news.	the ACM, vol. 40, no. 3, pages 77-87, 1997	filtering system for Usenet news. The public trial of GroupLens invited users from over a dozen newsgroups selected to represent a cross-section of Usenet to apply our news reader software to enter ratings and receive predictions (we provided GroupLens-adapted versions of Gnus, xrn, and tin). Over a seven-week trial starting February 8, 1996, we registered 250 users who submitted a total of 47,569 ratings and received over 600,000 predictions for 22,862 different articles. These users were volunteers who saw our announcement postings or our Web page. They downloaded specially modified news browsers that accepted ratings and displayed predictions on a 1–5 scale where 1 was described as “this item is really bad! a waste of net.bandwidth” and 5 as “this article is great, I would like to see more like it.” For privacy reasons, users were known to us only by pseudonyms. Qualitative results are therefore the compilation of feedback from the GroupLens mailing list and private email rather than a comprehensive survey.
[KRU1997]	Krulwosh, B.; Burkey, C.	1997	The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction	IEEE Expert/Intelligent Systems & Their Applications vol.12, no.5, S. 22-27, September/October 1997	InfoFinder is an intelligent agent that learns user information interests from sets of messages or other online documents that users have classified. While this problem has been addressed by a number of recent research initiatives, InfoFinder's approach is innovative in a number of ways. First, the agent uses heuristics to extract significant phrases from documents for learning rather than using statistical techniques. This enables it to learn highly general search criteria based on a small number of sample documents. Second, the agent's induction algorithm is based on the observation that sample documents in such an application will not be uniformly distributed, because of the fact that users will tend to classify positive examples while browsing while classifying negative examples only when the agent makes a bad recommendation. Third, the agent learns standard decision trees for each user category. These decision trees are easily transformed into search query strings for standard search systems rather than requiring specialized search engines, and are significantly more expressive than other representations such as positive and negative word lists.
[LAD1997]	Ladányi, H.	1997	SQL Unleashed	SAMS, ISBN 067231133X, 1997	SQL Unleashed, Second Edition, covers ANSII SQL and how to implement them with several major relational database platforms used on a day-to-day basis. SQL Unleashed focuses on more intermediate to advanced topics and the latest trends forthcoming in SQL. Designed to be used as both a reference and implementation tool, this book serves as a practical, pragmatic guide to day-to-day SQL programming and provide the most efficient solutions for getting the job done. Topics include database design and data definition, data manipulation, data selection, more

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					advanced, QL implementations, the future of SQL, and vendor SQL extensions.
[LAN1995]	Lang, K.	1995	News{W}eeder: learning to filter netnews	Proceedings of the 12th International Conference on Machine Learning. Lake Tahoe, CA, UNITED STATES ,1995	A significant problem in many information filtering systems is the dependence on the user for the creation and maintenance of a user profile, which describes the user's interests. NewsWeeder is a netnews-filtering system that addresses this problem by letting the user rate his or her interest level for each article being read (1-5), and then learning a user profill based on these ratings. This paper describes how NewsWeeder accomplishes this task, an examines the alternative learning methods used. The results show that a learning algorithm based on the Minimum Description Length (MDL) principle was able to raise the percentage of interesting articles to be shown to users from 14% to 52% on average. Further, this performance significantly outperformed (by 21%) one of the most successful techniques in Information Retrieval (IR), termfrequency/inverse-document-frequency (tf-idf) weighting.
[LEA2000]	Leake, D.B.; Bauer, T.; Maguitman, A.; Wilson, D.C.	2000	Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search	Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. AAAI Press, Menlo Park, 2000	Learning how to find relevant information sources is an important part of solving novel problems and mastering new domains. This paper introduces work on developing a lessons learned system that supports task-driven research by (1) automatically storing cases recording which information resources researchers consult during their decision-making; (2) using these cases to proactively suggest information resources to consult in similar future task contexts; and (3) augmenting existing information resources by providing tools to support users in elucidating and capturing records of useful information that they have found, for future reuse. Our approach integrates aspects of case-based reasoning, ``just-in-time'' task-based information retrieval, and concept mapping. We describe the motivations for this work and how lessons learned systems for suggesting research resources complement those that store task solutions. We present an initial system implementation that illustrates the desired properties, and close with a discussion of the primary questions and open issues to address.
[LEA2005]	Leake, D.; Maguitman, A.; Reichherzer, T.	2005	Exploiting Rich Context: An Incremental Approach to Context-Based Web Search	CONTEXT'05, Fifth International and Interdisciplinary Conference on Modeling and Using Context, Paris, France, 2005	Abstract. Proactive retrieval systems monitor a user's task context and automatically provide the user with related resources. The effectiveness of such systems depends on their ability to perform context-based retrieval, generating queries which return context-relevant results. Two factors make this task especially challenging for Web-based retrieval. First, the quality of Web retrieval can be strongly affected by the vocabulary used to generate the queries. If the system's vocabulary for describing the context differs from the vocabulary used in the resources themselves,

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					relevant resources may be missed. Second, search engine restrictions on query length may make it difficult to include sufficient contextual information in a single query. This paper presents an algorithm, IACS (Incremental Algorithm for Context-Based Search), which addresses these problems by building up, applying, and refining partial context descriptions incrementally. In IACS, an initial term-based context description is the starting point for a cycle of mining search engines, performing context-based filtering of results, and refining context descriptions to generate new rounds of queries in an expanded vocabulary. IACS has been applied in a system for proactively supporting concept-map-based knowledge modeling, by retrieving resources relevant to target concepts in the context of the rich information provided by in progress concept maps. An evaluation of the system shows that it provides significant improvements over a baseline for retrieving context-relevant resources. We expect the algorithm to have broad applicability to context-based Web retrieval for rich contexts.
[LEI2006]	-	2006	Wortschatz Lexikon der Universität Leipzig	http://wortschatz.uni-leipzig.de/index_js.html	-
[LIE1995]	Lieberman, H.	1995	Letizia: An Agent That Assists Web Browsing	Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Morgan Kaufmann publishers Inc, San Mateo, CA, United States, ISBN 1558603638, S. 924-929, 1995	Letizia is a user interface agent that assists a user browsing the World Wide Web. As the user operates a conventional Web browser such as Netscape, the agent tracks user behavior and attempts to anticipate items of interest by doing concurrent, autonomous exploration of links from the user's current position. The agent automates a browsing strategy consisting of a best-first search augmented by heuristics inferring user interest from browsing behavior.
[LIE1998]	Lieberman, H.; van Dyke, N.W.; Vivacqua, A.S.	1998	Let's browse: a collaborative Web browsing agent	Proceedings of the 4th international conference on Intelligent user interfaces, Los Angeles, California, United States, S. 65-68, 1998	Web browsing, like most of today's desktop applications, is usually a solitary activity. Other forms of media, such as watching television, are often done by groups of people, such as families or friends. What would it be like to do collaborative Web browsing? Could the computer provide assistance to group browsing by trying to help find mutual interests among the participants? Let's Browse is an experiment in building an agent to assist a group of people in browsing, by suggesting new material likely to be of common interest. It is built as an extension to the single user Web browsing agent Letizia. Let's Browse features automatic detection of the presence of users, automated "channel surfing" browsing, and dynamic display of the user profiles and explanation of recommendations.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[LIE2001]	Lieberman, H.; Fry, C.; Weitzman, L.	2001	Exploring the Web with reconnaissance agents	Communications of the ACM archive vol.44 , no.8 (August 2001) S. 69-75, 2001, ISSN:0001-0782	Every click on a link is a leap of faith. When you click on the blue underlined text or on a picture on a Web page, there is always a (sometimes much too long) moment of suspense when you are waiting for the page to load. Until you actually see what is behind the link, you don't know whether it will lead to the reward of another interesting page, to the disappointment of a junk page, or worse, to a "404 Not Found" error message. But what if you had an assistant always looking ahead for you—clicking on the Web links and checking out the page behind the link before you even get to it? An assistant that, like a good secretary, has a good idea of what you might like. The assistant could warn you if the page was irrelevant or alert you if that link or some other link merited your attention. The assistant could save you time and frustration. The function of such an assistant represents a new category of computer agents that will soon be as common as search engines in browsing assistance on the Web and in large databases and hypermedia networks. These agents are called reconnaissance agents—programs that look ahead in the user's browsing activities and act as an advance scout to save the user needless searching and recommend the best paths to follow. Reconnaissance agents are also among the first representatives of a new class of computer applications—learning agents that infer user preferences and interests by tracking interactions between the user and the machine over the long term. We'll provide two examples of reconnaissance agents: Letizia and Powerscout. The main difference is that Letizia uses local reconnaissance—searching the neighborhood of the current page, while Powerscout uses global reconnaissance—making use of a traditional search engine to search the Web in general. Both learn user preferences from watching the user's browsing, and both provide continuous, real-time display of recommendations. Reconnaissance agents treat Web browsing as a cooperative search activity between the human user and the computer agent, providing a middle ground between narrowly targeted retrieval that search engines typically provide and completely unconstrained manual browsing. One description of this landscape of systems organizes them around two axes, one characterizing reconnaissance connectivity (local vs. global) and one characterizing user effort (active vs. passive). Local reconnaissance traces links locally, while global reconnaissance uses global repositories such as search engines. Figure 1 plots a number of tools and agents against these attributes. Typical file browsing is in the lower-left quadrant while standard search engines are located in the upper-left quadrant.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[LIN1997]	Lin, C.; Hovy, E.	1997	Identifying Topics by Position	Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP--97), S. 283-290, 1997	This paper addresses the problem of identifying likely topics of texts by their posi- tion in the text. It describes the automated training and evaluation of an Optimal Posi- tion Policy, a method of locating the likely positions of topic-bearing sentences based on genre-specific regularities of discourse structure. This method can be used in applications such as information retrieval, routing, and text summarization
[LIN2001]	Linden, G.D.; Jacobi, J.A.; Benson, E.A.	2001	Collaborative recommendations using item-to-item similarity mappings	United States Patent 6.266.649, 2001	A recommendations service recommends items to individual users based on a set of items that are known to be of interest to the user, such as a set of items previously purchased by the user. In the disclosed embodiments, the service is used to recommend products to users of a merchant's Web site. The service generates the recommendations using a previously-generated table which maps items to lists of "similar" items. The similarities reflected by the table are based on the collective interests of the community of users. For example, in one embodiment, the similarities are based on correlations between the purchases of items by users (e.g., items A and B are similar because a relatively large portion of the users that purchased item A also bought item B). The table also includes scores which indicate degrees of similarity between individual items. To generate personal recommendations, the service retrieves from the table the similar items lists corresponding to the items known to be of interest to the user. These similar items lists are appropriately combined into a single list, which is then sorted (based on combined similarity scores) and filtered to generate a list of recommended items. Also disclosed are various methods for using the current and/or past contents of a user's electronic shopping cart to generate recommendations. In one embodiment, the user can create multiple shopping carts, and can use the recommendation service to obtain recommendations that are specific to a designated shopping cart. In another embodiment, the recommendations are generated based on the current contents of a user's shopping cart, so that the recommendations tend to correspond to the current shopping task being performed by the user.
[LIN2003]	Linden, G.D.; Smith, B.; York, J.	2003	Amazon.com Recommendations - Item-to-Item Collaborative Filtering	IEEE Internet Computing, vol.7, no.1, S. 76-80, 2003	Recommendation algorithms are best known for their use on e-commerce Web sites, where they use input about a customer’s interests to generate a list of recommended items. Many applications use only the items that customers purchase and explicitly rate to represent their interests, but they can also use other attributes, including items viewed, demographic data, subject interests, and favorite artists. At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					programming titles to a software engineer and baby toys to a new mother. The click-through and conversion rates - two important measures of Web-based and email advertising effectiveness - vastly exceed those of untargeted content such as banner advertisements and top-seller lists
[LIP1990]	Lipton, R.J.; Naughton, J.F., Schneider, D.A.	1990	Practical selectivity estimation through adaptive sampling	Proceedings of the 1990 ACM SIGMOD international conference on Management of data table of contents, Atlantic City, New Jersey, United States S. 1-11, 1990	Recently we have proposed an adaptive, random sampling algorithm for general query size estimation. In earlier work we analyzed the asymptotic efficiency and accuracy of the algorithm, in this paper we investigate its practicality as applied to selects and joins. First, we extend our previous analysis to provide significantly improved bounds on the amount of sampling necessary for a given level of accuracy. Next, we provide “sanity bounds” to deal with queries for which the underlying data 1s extremely skewed or the query result is very small. Finally, we report on the performance of the estimation algorithm as implemented in a host language on a commercial relational system. The results are encouraging, even with this loose coupling between the estimation algorithm and the DBMS.
[LIP1990b]	Lipton, R.J.; Naughton, J.F.	1990	Query size estimation by adaptive sampling	Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems table of contents, Nashville, Tennessee, United States, S. 40-46, 1990	We present an adaptive, random sampling algorithm for estimating the size of general queries. The algorithmcan be used for any query Q over a database D such that 1) for some n, the answer to Q can be partitioned into n disjoint subsets Q1, Q2, . . ., Qn and 2) for 1 <= i <= n, the size of Qi is bounded by some function b(D, Q), and 3) there is some algorithm by which we can compute the size of Qi, where i is chosen randomly. We consider the performance of the algorithm on three special cases of the algorithm: join queries, transitive closure queries, and general recursive Datalog queries.
[LIT1987]	Littlestone, N.	1987	Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm	Machine Learning 2, S. 285-318, Kluwer Academic Publisher, Boston, United Staes, 1988	Abstract. Valiant (1984) and others have studied the problem of learning various classes of Boolean functions from examples. Here we discuss incremental learning of these functions. We consider a setting in which the learner responds to each example according to a current hypothesis. Then the learner updates the hypothesis, if necessary, based on the correct classification of the example. One natural measure of the quality of learning in this setting is the number of mistakes the learner makes. For suitable classes of functions, learning algorithms are available that make a bounded number of mistakes, with the bound independent of the number of examples seen by the learner. We present one such algorithm that learns disjunctive Boolean functions, along with variants for learning other classes of Boolean functions. The basic method can be expressed as a linear-threshold algorithm. A primary advantage of this algorithm is that the number of mistakes grows only logarithmically with the number of irrelevant attributes in the examples. At the same time, the

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					algorithm is computationally efficient in both time and space.
[LOC1987]	Lockemann, P.C.; Schmidt, J.W.	1987	Datenbank-Handbuch	Springer, ISBN 354010741X	-
[LOE2003]	Löbner, S.	2003	Semantik	De Gruyter, ISBN 3110156741, Berlin, 2003	-
[LOV1968]	Lovins, J.B.	1968	Development of a stemming algorithm	Mechanical Translation and Computational Linguistics, vol. 11, S. 22-31, 1968	-
[LUH1958]	Luhn, H.P.	1958	The Automatic Creation of Literature Abstracts.	IBM Journal, pages S. 159-165, 1958	-
[MAC2005]	MacKay, D.J.C.	2005	Information Theory, Inference, and Learning Algorithms	640 S., Cambridge University Press 2005	Conventional courses on information theory cover not only the beautiful theoretical ideas of Shannon, but also practical solutions to communication problems. This book goes further, bringing in Bayesian data modelling, Monte Carlo methods, variational methods, clustering algorithms, and neural networks. Why unify information theory and machine learning? Because they are two sides of the same coin. In the 1960s, a single eld, cybernetics, was populated by information theorists, computer scientists, and neuroscientists, all studying common problems. Information theory and machine learning still belong together. Brains are the ultimate compression and communication systems. And the state-of-the-art algorithms for both data compression and error-correcting codes use the same tools as machine learning.
[MAG2000]	Maglio, P.P.; Barrett, R.; Campbell, C.S.; Selker, T.	2000	SUITOR: an attentive information system	Proceedings of the 5th international conference on Intelligent user interfaces, New Orleans, Louisiana, United States , S. 169-176, ISBN:1581131348, 2000	Attentive systems pay attention to what users do so that they can attend to what users need. Such systems track user behavior, model user interests, and anticipate user desires and actions. Because the general class of attentive systems is broad — ranging from human butlers to web sites that profile users — we have focused specifically on attentive information systems, which observe user actions with information resources, model user information states, and suggest information that might be helpful to users. In particular, we describe an implemented system, Simple User Interest Tracker (Suitor), that tracks computer users through multiple channels — gaze, web browsing, application focus — to determine their interests and to satisfy their information needs. By observing behavior and modeling users, Suitor finds and displays potentially relevant information that is both timely and non-disruptive to the users' ongoing activities.
[MAL1986]	Malone, T. W.; Grant, K. R.; Turbak, F. A.	1986	The information lens: an intelligent system for information	Conference on Human Factors in Computing	This paper describes an intelligent system to help people share and filter information communicated by computer-based messaging systems. The

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			sharing in organizations	Systems; Proceedings of the SIGCHI conference on Human factors in computing systems table of contents, Boston, Massachusetts, United States, 1986	system exploits concepts from artificial intelligence such as frames, production rules, and inheritance networks, but it avoids the unsolved problems of natural language understanding by providing users with a rich set of semi-structured message templates. A consistent set of "direct manipulation" editors simplifies the use of the system by individuals, and an incremental enhancement path simplifies the adoption of the system by groups.
[MAN1988]	Mannino, M.V.; Chu, P.; Sager, T.	1988	Statistical profile estimation in database systems	ACM Computing Surveys (CSUR) archive, vol. 20, no.3, Sep. 1988, S.191-221, 1988	A statistical profile summarizes the instances of a database. It describes aspects such as the number of tuples, the number of values, the distribution of values, the correlation between value sets, and the distribution of tuples among secondary storage units. Estimation of database profiles is critical in the problems of query optimization, physical database design, and database performance prediction. This paper describes a model of a database of profile, relates this model to estimating the cost of database operations, and surveys methods of estimating profiles. The operators and objects in the model include build profile, estimate profile, and update profile. The estimate operator is classified by the relational algebra operator (select, project, join), the property to be estimated (cardinality, distribution of values, and other parameters), and the underlying method (parametric, nonparametric, and ad-hoc). The accuracy, overhead, and assumptions of methods are discussed in detail. Relevant research in both the database and the statistics disciplines is incorporated in the detailed discussion.
[MAN2000]	Manber, U.; Patel, A.; Robison, J.	2000	Experience with personalization of Yahoo	Communications of the ACM, vol.43, no.8, S. 35-39, 2000.	-
[MAR2002]	Marx, W.; Gramm, G.	2002	Literaturflut - Informationslawine – Wissensexplosion. Wächst der Wissenschaft das Wissen über den Kopf?	Arbeitspapier des Max-Planck-Institut für Festkörperforschung Stuttgart (http://www.mpi-stuttgart.mpg.de/IVS/literaturflut.html)	Scientific information has stopped growing exponentially as in the last 300 years. Nevertheless, the number of scientific papers published yearly remains dramatic. Well ordered databases and sophisticated search systems allow scientists to find the needle in the haystack. A growing number of factual databases as well as more reviews compress and refine information. Not searching but controlling and working up information appear to become the most important problems in the future.
[MCC1998]	McCallum, A; Nigam, K.	1998	Naive Bayes algorithm for learning to classify text	AAAI-98 Workshop on "Learning for Text Categorization,Madison, Wisconsin, United States, 1998	Recent work in text classification has used two different first-order probabilistic models for classification, both of which make the naïve Bayes assumption. Some use a multi-variate Bernoulli model, that is, a Bayesian Network with no dependencies between words and binary word features (e.g. Larkey and Croft 1996; Koller and Sahami 1997). Others use a multinomial model, that is, a uni-gram language model with integer word counts (e.g.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					Lewis and Gale 1994; Mitchell 1997). This paper aims to clarify the confusion by describing the differences and details of these two models, and by empirically comparing their classification performance on five text corpora. We find that the multi-variate Bernoulli performs well with small vocabulary sizes, but that the multinomial performs usually performs even better at larger vocabulary sizes providing on average a 27% reduction in error over the multi-variate Bernoulli model at any vocabulary size.
[MCL2004]	McLaughlin, M.R.; Herlocker, J.L	2004	A collaborative filtering algorithm and evaluation metric that accurately model the user experience	Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom, Session: Content-based filtering & collaborative filtering, S. 329-336, 2004, ISBN:1581138814	Collaborative Filtering (CF) systems have been researched for over a decade as a tool to deal with information overload. At the heart of these systems are the algorithms which generate the predictions and recommendations.In this article we empirically demonstrate that two of the most acclaimed CF recommendation algorithms have flaws that result in a dramatically unacceptable user experience.In response, we introduce a new Belief Distribution Algorithm that overcomes these flaws and provides substantially richer user modeling. The Belief Distribution Algorithm retains the qualities of nearest-neighbor algorithms which have performed well in the past, yet produces predictions of belief distributions across rating values rather than a point rating value.In addition, we illustrate how the exclusive use of the mean absolute error metric has concealed these flaws for so long, and we propose the use of a modified Precision metric for more accurately evaluating the user experience.
[MEI1978]	Meier, H.	1978	Deutsche Sprachstatistik	Georg Olms Verlag, Hildesheim, Deutschland, ISBN 3487017695, 1978	-
[MEL2002]	Melville, P.; Mooney, R.J; Nagarajan, R.	2002	Content-Boosted Collaborative Filtering for Improved Recommendations	Eighteenth national conference on Artificial intelligence, Edmonton, Alberta, Canada, S.187-192, ISBN 0262511290, 2002	Most recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings. In this paper, we present an elegant and effective framework for combining content and collaboration. Our approach uses a content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering. We present experimental results that show how this approach, "Content-Boosted Collaborative Filtering", performs better than a pure content-based predictor, pure collaborative filter, and a naive hybrid approach.
[MEN1999]	Menascé, D.A.; Almeida,	1999	A Methodology for Workload	Proceedings of the 1st ACM	Performance analysis and capacity planning for e-commerce sites poses an

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
	V.A.F.; Fonseca, R.; Mendes, M.A:		Characterization of E-commerce Sites	conference on Electronic commerce, Denver, Colorado, United States S. 119-128, 1999	interesting problem: how to best characterize the workload of these sites. Tradition al workload characterization methods, based on hits/set, page views/set, or visits/set, are not appropriate for e-commerce sites. In these environments, customers interact with the site through a series of consecutive and related requests, called sessions. Different navigational patterns can be observed for different groups of customers. In this paper, we propose a methodology for characterizing and generating e-commerce workload models. First, we introduce a state transition graph called Customer Behaviour Model Graph (CBMG), that is used to describe the behaviour of groups of customers who exhibit similar navigational patterns. A set of useful metrics, analytically derived from the analysis of the CBMG, is presented. Next, we define a workload model and show the steps required to obtain its parameters. We then propose a clustering algorithm to characterize workloads of e-commerce sites in terms of CBMGs. Finally, we present and discuss experimental results of the use of proposed methodology.
[MET2000]	van Meteren, R.; van Someren, M.	2000	Using Content-Based Filtering for Recommendation	University of Amsterdam, Netherlands, 2000	Finding information on a large web site can be a difficult and time-consuming process. Recommender systems can help users find information by providing them with personalized suggestions. In this paper the recommender system PRES is described that uses content-based filtering techniques to suggest small articles about home improvements. A domain such as this implicates that the user model has to be very dynamic and learned from positive feedback only. The relevance feedback method seems to be a good candidate for learning such a user model, as it is both efficient and dynamic.
[MID2004]	Middleton, S.E.; Shadbolt, N.R.; De Roure, D.C.	2004	Ontological user profiling in recommender systems	ACM Transactions on Information Systems (TOIS) archive, vol.22 , no.1 (January 2004) , S. 54-88, 2004, ISSN:1046-8188	We explore a novel ontological approach to user profiling within recommender systems, working on the problem of recommending on-line academic research papers. Our two experimental systems, Quickstep and Foxtrot, create user profiles from unobtrusively monitored behaviour and relevance feedback, representing the profiles in terms of a research paper topic ontology. A novel profile visualization approach is taken to acquire profile feedback. Research papers are classified using ontological classes and collaborative recommendation algorithms used to recommend papers seen by similar people on their current topics of interest. Two small-scale experiments, with 24 subjects over 3 months, and a large-scale experiment, with 260 subjects over an academic year, are conducted to evaluate different aspects of our approach. Ontological inference is shown to improve user profiling, external ontological knowledge used to successfully bootstrap a recommender system and profile visualization

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					employed to improve profiling accuracy. The overall performance of our ontological recommender systems are also presented and favourably compared to other systems in the literature.
[MIL2004]	Miller, B.N.; Konstan, J.A.; Riedl, J.	2004	PocketLens: Toward a personal recommender system	ACM Transactions on Information Systems (TOIS) archive, vol. 22 , no.3 (July 2004), S. 437-476, 2004, ISSN:1046-8188	Recommender systems using collaborative filtering are a popular technique for reducing information overload and finding products to purchase. One limitation of current recommenders is that they are not portable. They can only run on large computers connected to the Internet. A second limitation is that they require the user to trust the owner of the recommender with personal preference data. Personal recommenders hold the promise of delivering high quality recommendations on palmtop computers, even when disconnected from the Internet. Further, they can protect the user's privacy by storing personal information locally, or by sharing it in encrypted form. In this article we present the new PocketLens collaborative filtering algorithm along with five peer-to-peer architectures for finding neighbors. We evaluate the architectures and algorithms in a series of offline experiments. These experiments show that Pocketlens can run on connected servers, on usually connected workstations, or on occasionally connected portable devices, and produce recommendations that are as good as the best published algorithms to date.
[MIT1995]	Mitschang, B.	1995	Anfrageverarbeitung in Datenbanksystemen	Vieweg Verlag, ISBN 3528054883, 420 Seiten, 1995	In diesem Buch werden die wichtigsten konzeptionellen und auch implementierungstechnischen Grundlagen eines Framework zur Anfrageverarbeitung entwickelt. Dieses Framework stellt eine wiederverwendbare und erweiterbare Basis zur Entwicklung von angepassten Anfrageprozessoren bereit. Damit ist es möglich, die Anfrageverarbeitung (und damit auch das DBS) auf eine konkrete Einsatzumgebung zuzuschneiden. Der offensichtliche Nutzen dieser Methodik liegt in der deutlich verringerten Systementwicklungszeit, der flexiblen Anpassungsfähigkeit sowie der hohen Wiederverwendung von Technologien, Implementierungskonzepten und auch von bereits existierender Software.
[MOB2000]	Mobasher, A; Dai, H.; Luo, T.; Nakagawa, M.; Witshire, J..	2000	Discovery of Aggregate Usage Profiles for Web Personalization	Proceedings of the WebKDD Workshop, 2000	Web usage mining, possibly used in conjunction with standard approaches to personalization such as collaborative filtering, can help address some of the shortcomings of these techniques, including reliance on subjective user ratings, lack of scalability, and poor performance in the face high-dimensional and sparse data. However, the discovery of patterns from usage data by itself is not sufficient for performing the personalization tasks. The critical step is the effective derivation of good quality and useful (i.e., actionable) "aggregate usage profiles" from these patterns. In this paper we present and experimentally

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					evaluate two techniques, based on clustering of user transactions and clustering of pageviews, in order to discover overlapping aggregate profiles that can be effectively used by recommender systems for real-time personalization. We evaluate these techniques both in terms of the quality of the individual profiles generated, as well as in the context of providing recommendations as an integrated part of a personalization engine.
[MOC1996]	Mock, K.J	1996	Intelligent Information Filtering via Hybrid Techniques: Hill Climbing, Case-Based Reasoning, Index Patterns, and Genetic Algorithms	Ph.D. thesis, University of California Davis, United States, 1996	As the size of the Internet increases, the amount of data available to users has dramatically risen, resulting in an information overload for users. This work shows that information overload is a problem, and that data is organized poorly by existing browsers. To address these problems, an intelligent information news filtering system named INFOS (Intelligent News Filtering Organizational System) was created to reduce the user's search burden by automatically eliminating Usenet news articles predicted to be irrelevant. These predictions are learned automatically by adapting an internal user model that is based upon features taken from articles and collaborative features derived from other users. The features are manipulated through keyword-based techniques, knowledge-based techniques, and genetic algorithms to build a user model to perform the actual filtering. The integration of knowledge-based techniques for in-depth analysis, statistical and keyword approaches for scalability, and genetic algorithms for exploration allows INFOS to achieve better filtering performance than by using either technique alone. Experimental results collected from the prototype of INFOS validate the gain in performance within the domain of news articles posted to electronic bulletin boards.
[MON2003]	2003	2003	A Taxonomy of Recommender Agents on the Internet	Artificial Intelligence Review 19, S. 285-330, Netherlands, 2003.	Recently, Artificial Intelligence techniques have proved useful in helping users to handle the large amount of information on the Internet. The idea of personalized search engines, intelligent software agents, and recommender systems has been widely accepted among users who require assistance in searching, sorting, classifying, filtering and sharing this vast quantity of information. In this paper, we present a state-of-the-art taxonomy of intelligent recommender agents on the Internet. We have analyzed 37 different systems and their references and have sorted them into a list of 8 basic dimensions. These dimensions are then used to establish a taxonomy under which the systems analyzed are classified. Finally, we conclude this paper with a cross-dimensional analysis with the
[MOO2000]	Mooney, R.J; Roy, L.	2000	Content-Based Book Recommending Using Learning for Text Categorization	Proceedings of the fifth ACM conference on Digital libraries, S.	Recommender systems improve access to relevant products and information by making personalized suggestions based on previous examples of a user's likes and dislikes. Most existing

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				195-204, San Antonio, Texas, United States, 2000	recommender systems use collaborative filtering methods that base recommendations on other users' preferences. By contrast, content-based methods use information about an item itself to make suggestions. This approach has the advantage of being able to recommend previously unrated items to users with unique interests and to provide explanations for its recommendations. We describe a content-based book recommending system that utilizes information extraction and a machine-learning algorithm for text categorization. Initial experimental results demonstrate that this approach can produce accurate recommendations.
[MOU1997]	Moukas, A.	1997	Amalthaea: Information Discovery and Filtering using a Multiagent Evolving Ecosystem	Proceedings of the Conference on Practical Applications of Agents and Multiagent Technology, 1996	Agents are semi-intelligent programs that assist the user in performing repetitive and time-consuming tasks. Information discovery and information filtering are a suitable domain for applying agent technology. Ideas drawn from the field of autonomous agents and artificial life are combined in the creation of an evolving ecosystem composed of competing and cooperating agents. A co-evolution model of information filtering agents that adapt to the various user's interests and information discovery agents that monitor and adapt to the various on-line information sources is analyzed. Results from a number of experiments are presented and discussed.
[MYS2005]	MySQL AB	2005	MySQL Full-text Search	MySQL Internals Manual, 4.7, 2005	-
[NAI1982]	Naisbitt, J.	1982	Megatrends: Ten New Directions Transforming Our Lives.	Warner Books, 1982, 290 Seiten, New York, United States, ISBN 0446512516, 1982	-
[NIS1997]	National Information Standards Organization	1997	Guidelines for Abstracts	ANSIINISO 239.14-1997, Revision of ANSI 239.14-1979 (R1987), ISSN: 1041-5653, Approved November 27,1996 by the American National Standards Institute	Guidance is presented for authors and editors preparing abstracts that represent the content of texts reporting on the results of experimental work or descriptive or discursive studies. Suggestions for the placement of abstracts within publications or other media are given, along with recommendations for abstracting specific documents. Types of abstracts and their content are described. Also included are suggestions on the style of abstracts and a list of selected readings on the subject of abstracting. Examples of abstracts are appended.
[OPP1927]	Oppenheimer, C.	1927	Die papierne Sintflut	Chemiker-Zeitung 51(24), S. 229-230 (1927)	-
[PAI1994]	Paice, C.D.	1994	An evaluation method for stemming algorithms	Proceedings of the 17th annual international ACM SIGIR conference on Research and development in	The effectiveness of stemming algorithms has usually been measured in terms of their effect on retrieval performance with test collections. This however does not provide any insights which might help in stemmer optimisation. This paper describes a method in which stemming performance

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				information retrieval table of contents, Dublin, Ireland, S.42-50, ISBN 038719889X, 1994	is assessed against predefine concept groups in samples of words. This enables various indices of stemming performance and weight to be computed. Results are reported for three stemming algorithms. The validity and usefulness of the approach, and the problems of conceptual grouping, are discussed, and directions for further research are identified.
[PAI2001]	Paik, W.; Yilmazel, S.; Brown, E.; Poulin, M.; Dubon, S.; Amice, C.	2001	Applying natural language processing (NLP) based metadata extraction to automatically acquire user preferences	Proceedings of the 1st international conference on Knowledge capture, Victoria, British Columbia, Canada, S.116-122, ISBN 1581133804, 2001	This paper describes a metadata extraction technique based on natural language processing (NLP) which extracts personalized information from email communications between financial analysts and their clients. Personalized means connecting users with content in a personally meaningful way to create, grow, and retain online relationships. Personalization often results in the creation of user profiles that store individuals' preferences regarding goods or services offered by various e-commerce merchants. With the introduction of e-commerce, it has become more difficult to develop and maintain personalized information due to larger transaction volumes. <!metaMarker> is an NLP and Machine Learning (ML)-based automatic metadata extraction system designed to process textual data such as emails, discussion group postings, or chat group transcriptions. <!metaMarker> extracts both explicit and implicit metadata elements including proper names, numeric concepts, and topic/subject information. In addition, Speech Act Theory inspired metadata elements, which represent the message creators' intention, mood, and urgency are also extracted. In a typical dialogue between financial analysts and their clients, clients often discuss the items that they liked or have an interest. By extracting this information, <!metaMarker> constructs user profiles automatically. This system has been designed, implemented, and tested with real-world data. The overall accuracy and coverage of extracting explicit and implicit metadata is about 90%. In summary, the paper shows that an NLP-based metadata extraction system enables automatic user profiling with high effectiveness.
[PAZ1996]	Pazzani, M.; Muramatsu, J; Billsus, D.	1996	Syskill & Webert: Identifying interesting web sites	Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, Irvine, CA, United States, 1996.	We describe Syskill & Webert, a software agent that learns to rate pages on the World Wide Web (WWW), deciding what pages might interest a user. The user rates explored pages on a three point scale, and Syskill & Webert learns a user profile by analyzing the information on each page. The user profile can be used in two ways. First, it can be used to suggest which links a user would be interested in exploring. Second, it can be used to construct a LYCOS query to find pages that would interest a user. We compare six different algorithms from machine learning and information retrieval on this task. We find that the naïve Bayesian classifier offers several advantages over other

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					learning algorithms on this task. Furthermore, we find that an initial portion of a web page is sufficient for making predictions on its interestingness substantially reducing the amount of network transmission required to make predictions.
[PEA1901]	Pearson, K.	1901	On lines and planes of closest fit to systems of points in space	The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, 6 (2), S. 559-572, 1901.	-
[POH1996]	Pohl, W.	1996	Learning About the User – User Modeling and Machine Learning	In Proceedings of the ICML'96 Workshop Machine Learning meets Human-Computer Interaction, S. 29-40,1996	User modeling is employed by applications that need to maintain explicit models of their users in order to exhibit individualized behaviour. The user modeling task involves representation and acquisition of assumptions about the user. Particularly user model acquisition is closely related to the machine learning task of automatically acquiring new information as well as new representations of existing information. This paper shows how and for which purposes machine learning techniques have been and could be employed in user modeling. Also usage modeling, a more action-centered approach to user modeling, is considered. Finally, the LaboUr approach to user modeling is sketched, which regards user modeling as learning problem.
[POH1997]	Pohl, W.	1997	LaboUr - Machine Learning for User Modeling	Human-Computer Interaction, 1997	Traditional user modeling systems are often limited, as far as processing of observations about user behavior and handling of user model dynamics are concerned. In this paper, the LaboUr architecture for user modeling systems is discussed. It realizes user modeling as open learning process, thus overcoming the mentioned limitations.
[POH1999]	Pohl, W.; Nick, A.	1999	Machine learning and knowledge representation in the LaboUr approach to user modeling	Proceedings of the seventh international conference on User modelling, Banff, Canada, S.179-188, 1999	Abstract. In early user-adaptive systems, the use of knowledge representation methods for user modeling has often been the focus of research. In recent years, however, the application of machine learning techniques to control user-adapted interaction has become popular. In this paper, we present and compare adaptive systems that use either knowledge representation or machine learning for user modeling. Based on this comparison, several dimensions are identified that can be used to distinguish both approaches, but also to characterize user modeling systems in general. The LaboUr (Learning about the User) approach to user modeling is presented which attempts to take an ideal position in the resulting multi-dimensional space by combining machine learning and knowledge representation techniques. Finally, an implementation of LaboUr ideas into the information server ELFI is sketched.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[POR1980]	Porter, M.F.	1980	An algorithm for suffix stripping	In "New models in probabilistic information retrieval", British Library Research and Development Report, no. 5587, chapter 6, 1980	Removing suffixes by automatic means is an operation which is especially useful in the field of information retrieval. In a typical IR environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or \terms\. Terms with a common stem will usually have similar meanings, for example: CONNECT CONNECTED CONNECTING CONNECTION CONNECTIONS Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.
[POU1996]	Poulovassilis, A.; Small, C.	1996	Investigation of Algebraic Query Optimisation for Database Programming Languages	Proceedings of the 20th VLDB Conference, Santiago, Chile, S. 415-426,1994	A major challenge still facing the designers and implementors of database programming languages (DBPLs) is that of query optimisation. We investigate algebraic query optimisation techniques for DBPLs in the context of a purely declarative functional language that supports sets as first-class objects. Since the language is computationally complete issues such as non-termination of expressions and construction of infinite data structures can be investigated, whilst its declarative nature allows the issue of side effects to be avoided and a richer set of equivalences to be developed. The support of a set bulk data type enables much prior work on the optimisation of relational languages to be utilised. Finally, the language has a well-defined semantics which permits us to reason formally about the properties of expressions, such as their equivalence with other expressions and their termination.
[PUT2004]	Putnam, H.	2004	Die Bedeutung von Bedeutung	Klostermann Texte Philosophie, ISBN 3465032314, 2004	Hilary Putnam zählt zu den bedeutendsten zeitgenössischen Philosophen der Vereinigten Staaten. Dem herrschenden Empirismus stellt er seinen Realismus entgegen, der ihn auch in der Sprachphilosophie zu einer Position führte, die schon im Kern Neuheit und Originalität beanspruchen darf, was selten vorkommt und um so aufregender ist. In dem hier übersetzten Aufsatz "The Meaning of Meaning" hat er seine Position sowie seine Kritik an anderen Auffassungen am ausführlichsten und in leicht fasslicher, nicht technischer Form dargelegt.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[QUI1986]	Quinlan, J. R.	1986	Induction of Decision Trees	Machine Learning, vol.1, no.1, Kluwer Academic Publishers, 1986	The technology for building knowledge-based systems by inductive inference from examples has been demonstrated successfully in several practical applications. This paper summarizes an approach to synthesizing decision trees that has been used in a variety of systems, and it describes one such system, ID3, in detail. Results from recent studies show ways in which the methodology can be modified to deal with information that is noisy and/or incomplete. A reported shortcoming of the basic algorithm is discussed and two means of overcoming it are compared. The paper concludes with illustrations of current research directions.
[RAM1988]	Ramakrishna, M. V.	1988	Hashing practice: analysis of hashing and universal hashing	Proceedings of the 1988 ACM SIGMOD international conference on Management of data, Chicago, Illinois, United States, S. 191-199, 1988	Much of the literature on hashing deals with overflow handling (collision resolution) techniques and its analysis. What does all the analytical results mean in practice and how can they be achieved with practical files? This paper considers the problem of achieving analytical performance of hashing techniques in practice with reference to successful search lengths, unsuccessful search lengths and the expected worst case performance (expected length of the longest probe sequence). There has been no previous attempt to explicitly link the analytical results to performance of real life files. Also, the previously reported experimental results deal mostly with successful search lengths. We show why the well known division method performs “well” under a specific model of selecting the test file. We formulate and justify an hypothesis that by choosing functions from a particular class of hashing functions, the analytical performance can be obtained in practice on real life files. Experimental results presented strongly support our hypothesis. Several interesting problems arising are mentioned in conclusion.
[RES1994]	Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J.	1994	GroupLens: An Open Architecture for Collaborative Filtering of Netnews	Proceedings of ACM, Conference on Computer Supported Cooperative Work, S. 175-186, Chapel Hill, NC, UNITED STATES, 1994	Collaborative filters help people make choices based on the opinions of other people. GroupLens is a system for collaborative filtering of netnews, to help people find articles they will like in the huge stream of available articles. News reader clients display predicted scores and make it easy for users to rate articles after they read them. Rating servers, called Better Bit Bureaus, gather and disseminate the ratings. The rating servers predict scores based on the heuristic that people who agreed in the past will probably agree again. Users can protect their privacy by entering ratings under pseudonyms, without reducing the effectiveness of the score prediction. The entire architecture is open: alternative software for news clients and Better Bit Bureaus can be developed independently and can interoperate with the components we have developed.
[RES1997]	Resnick, P.; Varian, H.R.	1997	Recommender systems	Communications of the ACM, vol. 40, no. 3, S. 56-58,	-

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				1997	
[RHO1996]	Rhodes, B.J; Starner, T.	1996	The Remembrance Agent: A Continuously Running Information Retrieval System	Proceedings of the First International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology, S. 486-495, 1996	The Remembrance Agent (RA) is a program which augments human memory by displaying a list of documents which might be relevant to the user's current context. Unlike most information retrieval systems, the RA runs continuously without user intervention. Its unobtrusive interface allows a user to pursue or ignore the RA's suggestions as desired.
[RHO1999]	Rhodes, B.J; Starner, T.	1999	Everyday-use Wearable Computers	MIT Technical Report, 1999	Since 1993, members of the MIT Wearable Computing Project have been engaged in a \living experiment," incorporating wearable computing into their everyday lives. Such immediate access to computation power enables a unique lifestyle and has many social implications. Through the use of anecdotes, this paper will attempt to relate our observations on the perception and adoption of new technology, interface issues, collaboration, and privacy as related to the intimate use of wearable computing.
[RHO2000]	Rhodes, B.J.	2000	Margin Notes: Building a Contextually Aware Associative Memory	Proceedings of the International Conference on Intelligent User Interfaces, , S. 219-224., 2000	Both the Human Computer Interaction and Information Retrieval fields have developed techniques to allow a searcher to find the information they seek quickly. However, these techniques are designed to augment one’s direct-recall memory, where the searcher is actively trying to find information. Associative memory, in contrast, happens automatically and continuously, triggering memories that relate to the observed world. This paper presents design techniques and heuristics for building “remembrance agents,” applications that watch a user’s context and proactively suggest information that may be of use. General design issues are discussed and illuminated by a description of Margin Notes, an automatic just-in-time information system for the Web.
[RHO2000a]	Rhodes, B.J.; Maes, P.	2000	Just-in-time information retrieval agents	IBM Systems Journal archive, vol. 39, no.3-4 , S. 685-704, ISSN:0018-8670, 2000	A just-in-time information retrieval agent (JITIR agent) is software that proactively retrieves and presents information based on a person's local context in an easily accessible yet nonintrusive manner. This paper describes three implemented JITIR agents: the Remembrance Agent, Margin Notes, and Jimminy. Theory and design lessons learned from these implementations are presented, drawing from behavioral psychology, information retrieval, and interface design. They are followed by evaluations and experimental results. The key lesson is that users of JITIR agents are not merely more efficient at retrieving information, but actually retrieve and use more information than they would with traditional search engines.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[RHO2001]	Rhodes, B.J.	2001	Method and apparatus for automated, context-dependent retrieval of information	United States Patent 6.236.768, 2001	Documents stored in a database are searched for relevance to contextual information, instead of (or in addition to) similar text. Each stored document is indexed in term of meta-information specifying contextual information about the document. Current contextual information is acquired, either from the user or the current computational or physical environment, and this "meta-information" is used as the basis for identifying stored documents of possible relevance.
[RIL1995]	Riloff, E.	1995	Little words can make a big difference for text classification	Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, United States, S.130-136, ISBN 0897917146, 1995	Most information retrieval systems use stopword lists and stemming algorithms. However, we have found that recognizing singular and plural nouns, verb forms, negation, and prepositions can produce dramatically different text classification results. We present results from text classification experiments that compare relevancy signatures, which use local linguistic context, with corresponding indexing terms that do not. In two different domains, relevancy signatures produced better results than the simple indexing terms. These experiments suggest that stopword lists and stemming algorithms may remove or conflate many words that could be used to create more effective indexing terms.
[RIS1978]	Rissanen, J.	1978	Modeling by shortest data description	Automatica, vol. 14, S. 465-471, 1978	-
[ROB1977]	Robertson, S.E.	1977	The probability ranking principle in IR	Journal of Documentation, Vol.33, S. 294-304, 1977	-
[ROB1992]	Robertson, S.E.; Walker, S.; Hancock-Beaulieu, M.; Gull, A.; Lau, M.	1992	Okapi at TREC	Proceedings of the first Text REtrieval Conference (TREC-1), S. 21-30, Gaithersburg, Maryland, United States, 1992.	The Okapi retrieval system is described, technically and in terms of its design principles. These include simplicity, robustness and ease of use. The version of Okapi used for TREC is further discussed. Designing experiments within the TREC constraints but using Okapi's supposed strengths proved problematic, and some compromise was necessary. The official TREC runs were (a) very simple automatic processing of the ad-hoc topics; (b) manually constructed ad-hoc queries; (c) feedback on the manual queries from searchers' relevance judgements; and (d) routing queries automatically obtained using the training set in a form of relevance feedback. The best run (manual with feedback), although not up to the best reported TREC results, was respectable, and an encouragement to further development within the same principles.
[ROB1997]	Robertson, S.E.; Walzer, S.	1997	On relevance weights with little relevance information	N.J. Belkin, A.D. Narasimhalu and P. Willett (Herausgeber), SIGIR '97, ACM, S.16-24, 1997	-

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[ROB2004]	Robertson, S.E.	2004	Understanding inverse document frequency: on theoretical arguments for IDF	Journal of Documentation 60, 503-520, 2004	The term weighting function known as IDF was proposed in 1972, and has since been extremely widely used, usually as part of a TF*IDF function. It is often described as a heuristic, and many papers have been written (some based on Shannon's Information Theory) seeking to establish some theoretical basis for it. Some of these attempts are reviewed, and it is shown that the Information Theory approaches are problematic, but that there are good theoretical justifications of both IDF and TF*IDF in traditional probabilistic model of information retrieval.
[RUC1997]	Rucker, J.; Polanco, M.J.	1997	Siteseer: Personalized Navigation for the Web	Communications of the ACM, vol. 40, no. 3, S. 73-75, 1997	Siteseer is a web-page recommendation system that uses an individual's bookmarks and the organization of bookmarks within folders for predicting and recommending relevant pages. Siteseer utilizes each user's bookmarks as an implicit declaration of interest in the underlying content, and the user's grouping behavior (such as the placement of subjects in folders) as an indication of semantic coherency or relevant groupings between subjects. In addition, Siteseer treats folders as a personal classification system which enables it to contextualize recommendations in classes defined by the user. Over time, Siteseer learns each user's preferences and the categories through which they view the world, and at the same time it learns for each Web page how different communities or affinity-based clusters of users regard it. Siteseer then delivers personalized recommendations of online content, Web pages, organized according to each user's folders.
[SAL1965]	Salton, G.; Lesk, M.E.	1965	The SMART automatic document retrieval systems—an illustration	Communications of the ACM, vol.8 , no.6 (June 1965) , S. 391-398, ISSN:0001-0782, 1965	A fully automatic document retrieval system operating on the IBM 7094 is described. The system is characterized by the fact that several hundred different methods are available to analyze documents and search requests. This feature is used in the retrieval process by leaving the exact sequence of operations initially unspecified, and adapting the search strategy to the needs of individual users. The system is used not only to simulate an actual operating environment, but also to test the effectiveness of the various available processing methods. Results obtained so far seem to indicate that some combination of analysis procedures can in general be relied upon to retrieve the wanted information. A typical search request is used as an example in the present report to illustrate systems operations and evaluation procedures.
[SAL1971]	Salton, G.	1971	The SMART Retrieval System - Experiments in Automatic Document Processing	Prentice-Hall, Inc. Upper Saddle River, NJ, United States, 1971	-
[SAL1975]	Salton, G.; Wong, A.;	1975	A vector space model for automatic	Communications of the ACM, vol.18	In a document retrieval, or other pattern matching environment where stored entities (documents) are compared with

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
	Yang, C.S.		indexing	no. 11, S.613-620, 1975	each other or with incoming patterns (search requests), it appears that the best indexing (property) space is one where each entity lies as far away from the others as possible; in these circumstances the value of an indexing system may be expressible as a function of the density of the object space; in particular, retrieval performance may correlate inversely with space density. An approach based on space density computations is used to choose an optimum indexing vocabulary for a collection of documents. Typical evaluation results are shown, demonstating the usefulness of the model.
[SAL1973]	Salton, G.; Yang, C.S.	1973	On the specification of term values in automatic indexing	Journal of Documentation, vol. 29, S. 351-372, 1973	-
[SAR2001]	Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J.	2001	Item-based collaborative filtering recommendation algorithms	Proceedings of the 10th international conference on World Wide Web, Hong Kong, S. 285-295, ISBN 1581133480, 2001	Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. These systems, especially the k-nearest neighbor collaborative filtering based ones, are achieving widespread success on the Web. The tremendous growth in the amount of available information and the number of visitors to Web sites in recent years poses some key challenges for recommender systems. These are: producing high quality recommendations, performing many recommendations per second for millions of users and items and achieving high coverage in the face of data sparsity. In traditional collaborative filtering systems the amount of work increases with the number of participants in the system. New recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale problems. To address these issues we have explored item-based collaborative filtering techniques. Itembased techniques first analyze the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users. In this paper we analyze different item-based recommendation generation algorithms. We look into different techniques for computing item-item similarities (e.g., item-item correlation vs. cosine similarities between item vectors) and different techniques for obtaining recommendations from them (e.g., weighted sum vs. regression model). Finally, we experimentally evaluate our results and compare them to the basic k-nearest neighbor approach. Our experiments suggest that item-based algorithms provide dramatically better performance than user-based algorithms, while at the same time providing better quality than the best available user-based algorithms.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[SCH1995]	Schlageter, G.	1995	Datenbanksysteme	Unterlagen zu Kurs 1665 des Fachbereichs Informatik der Fernuniversität GHS Hagen, 1995	-
[SCH1999]	Schultz, J.M.; Liberman, M.	1999	Topic Detection and Tracking using idf-Weighted Cosine Coefficient	Proceedings of the DARPA Broadcast News Workshop, S. 189-192, 1999	The goal of TDT Topic Detection and Tracking is to develop automatic methods of identifying topically related stories within a stream of news media. We describe approaches for both detection and tracking based on the well-known idf-weighted cosine coefficient similarity metric. The surprising outcome of this research is that we achieved very competitive results for tracking using a very simple method of feature selection, without word stemming and without a score normalization scheme. The detection task results were not as encouraging though we attribute this more to the clustering algorithm than the underlying similarity metric.
[SCH2002]	Schein, A.I.; Popescul, A.; Ungar, L.H.; Pennock, D.M.	2002	Collaborative Filtering: Methods and Metrics for Cold-Start Recommendations	Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, S. 253–260, Tampere, Finland, 2002	We have developed a method for recommending items that combines content and collaborative data under a single probabilistic framework. We benchmark our algorithm against a naive Bayes classifier on the cold-start problem, where we wish to recommend items that no one in the community has yet rated. We systematically explore three testing methodologies using a publicly available data set, and explain how these methods apply to specific real-world applications. We advocate heuristic recommenders when benchmarking to give competent baseline performance. We introduce a new performance metric, the CROC curve, and demonstrate empirically that the various components of our testing strategy combine to obtain deeper understanding of the performance characteristics of recommender systems. Though the emphasis of our testing is on cold-start recommending, our methods for recommending and evaluation are general.
[SED1995]	Sedgewick, R.	1995	Algorithmen	Addison-Wesley, Bonn	-
[SEO2000]	Seo, Y-W; Zhang, B-T	2000	Personalized web-document filtering using reinforcement learning	Applied Artificial Intelligence, vol.15, no.7, August 2001, S. 665-685	Document filtering is increasingly deployed in Web environments to reduce information overload of users. We formulate online information filtering as a reinforcement learning problem, i.e. TD(0). The goal is to learn user profiles that best represent his information needs and thus maximize the expected value of user relevance feedback. A method is then presented that acquires reinforcement signals automatically by estimating user's implicit feedback from direct observations of browsing behaviors. This "learning by observation" approach is contrasted with conventional relevance feedback methods which require explicit user feedbacks. Field

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					tests have been performed which involved 10 users reading a total of 18,750 HTML documents during 45 days. Compared to the existing document filtering techniques, the proposed learning method showed superior performance in information quality and adaptation speed to user preferences in online filtering.
[SES1994]	Seshadri, P.; Livny, M.; Ramakrishnan, R.	1993	Sequence query processing	Proceedings of the 1994 ACM SIGMOD international conference on Management of data, Minneapolis, Minnesota, United States, S. 430-441, 1994	Many applications require the ability to manipulate sequences of data. We motivate the importance of sequence query processing, and present a framework for the optimization of sequence queries based on several novel techniques. These include query transformations, optimizations that utilize meta-data, and caching of intermediate results. We present a bottom-up algorithm that generates an efficient query evaluation plan based on cost estimates. This work also identifies a number of directions in which future research can be directed.
[SHA1948]	Shannon, C.E.	1948	A Mathematical Theory of Communication	Reprinted with corrections from The Bell System Technical Journal, Vol. 27, S. 379–423, 623–656, 1948, United States	The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist1 and Hartley2 on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.
[SHA1986]	Shapiro, L.D.	1986	Join processing in database systems with large main memories	ACM Transactions on Database Systems (TODS) archive, vol.11, no.3 (Sep 1986), S. 239-264, 1986	We study algorithms for computing the equijoin of two relations in a system with a standard architecture hut with large amounts of main memory. Our algorithms are especially efficient when the main memory available is a significant fraction of the size of one of the relations to he joined; but they can be applied whenever there is memory equal to approximately the square root of the size of one relation. We present a new algorithm which is a hybrid of two hash-based algorithms and which dominates the other algorithms we present, including sort-merge. Even in a virtual memory environment, the hybrid algorithm dominates all the others we study. Finally, we describe how three popular tools to increase the efficiency of joins, namely filters, Babb arrays, and semijoins, can he grafted onto any of our algorithms.
[SHA1995]	Shardanand, U.; Maes, P.	1995	Social Information Filtering: Algorithms for Automating "Word of Mouth"	Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, pages 210—217 , Denver, Colorado, UNITED	This paper describes a technique for making personalized recommendations from any type of database to a user based on similarities between the interest profile of that user and those of other users. In particular, we discuss the implementation of a networked system called Ringo, which makes personalized recommendations for music albums and artists. Ringo's database of users and

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				STATES (1995)	artists grows dynamically as more people use the system and enter more information. Four different algorithms for making recommendations by using social information filtering were tested and compared. We present quantitative and qualitative results obtained from the use of Ringo by more than 2000 people.
[SHE1990]	Shekita, E.J.; Carey, M.J.	1990	A performance evaluation of pointer-based joins	Proceedings of the 1990 ACM SIGMOD international conference on Management of data, Atlantic City, New Jersey, United States S. 300-311, 1990	In this paper we describe three pointer-based join algorithms that are simple variants of the nested-loops, sort-merge, and hybrid-hash join algorithms used in relational database systems. Each join algorithm is described and an analysis is carried out to compare the performance of the pointer-based algorithms to their standard, non-pointer-based counterparts. The results of the analysis show that the pointer-based algorithms can provide significant performance gains in many situations. The results also show that the pointer-based nested-loops join algorithm, which is perhaps the most natural pointer-based join algorithm to consider using in an object-oriented database system, performs quite poorly on most medium to large joins.
[SIN2001]	Singhal, A.	2001	Modern information retrieval: A brief overview	IEEE Data Eng. Bull., Vol. 24 (4), S.35-43, 2001.	For thousands of years people have realized the importance of archiving and finding information. With the advent of computers, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. The field of Information Retrieval (IR) was born in the 1950s out of this necessity. Over the last forty years, the field has matured considerably. Several IR systems are used on an everyday basis by a wide variety of users. This article is a brief overview of the key advances in the field of Information Retrieval, and a description of where the state-of-the-art is at in the field.
[SIM1996]	Simmen, D.; Shekita, E.; Malkemus, T.	1996	Fundamental techniques for order optimization	Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, Quebec, Canada, S. 57-67, 1996	Decision support applications are growing in popularity as more business data is kept on-line. Such applications typically include complex SQL queries that can test a query optimizer's ability to produce an efficient access plan. Many access plan strategies exploit the physical ordering of data provided by indexes or sorting. Sorting is an expensive operation, however. Therefore, it is imperative that sorting is optimized in some way or avoided all together. Toward that goal, this paper describes novel optimization techniques for pushing down sorts in joins, minimizing the number of sorting columns, and detecting when sorting can be avoided because of predicates, keys, or indexes. A set of fundamental operations is described that provide the foundation for implementing such techniques. The operations exploit data properties that arise from predicate application, uniqueness, and functional dependencies. These operations and techniques have been implemented in IBM's DB2/CS.

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
[SPA1972]	Spärck-Jones, Karen	1972	A statistical interpretation of term specificity and its application in retrieval	Journal of Documentaion, Vol. 28, S. 11-21	The exhaustivity of document descriptions and the specificity of index terms are usually regarded as independent. It is suggested that specificity should be interpreted statistically, as a function of term use rather than of term meaning. The effects on retrieval of variations in term specificity are examined, experiments with three test collections showing, in particular, that frequently-occurring terms are required for good overall performance. It is argued that terms should be weighted according to collection frequency, so that matches on less frequent, more specific, terms are of greater value than matches on frequent terms. Results for the test collections show that considerable improvements in performance are obtained with this very simple procedure.
[SPA2000]	Spärck-Jones, Karen; Walker, S.; Robertson, S.E.	2000	A probabilistic model of information retrieval: development and comparative experiments	Information Processing and Management – Part 1, vol. 36, S. 779-808, 2000	The paper combines a comprehensive account of the probabilistic model of retrieval with new systematic experiments on TREC Programme material. It presents the model from its foundations through its logical development to cover more aspects of retrieval data and a wider range of system functions. Each step in the argument is matched by comparative retrieval tests, to provide a single coherent account of a major line of research. The experiments demonstrate, for a large test collection, that the probabilistic model is effective and robust, and that it responds appropriately, with major improvements in performance, to key features of retrieval situations.
[SPA2004]	Spärck-Jones, Karen	2004	IDF term weighting and IR research lessons	Journal of Documentation 60, S. 521-523, 2004	Robertson comments on the theoretical status of IDF term weighting. Its history illustrates how ideas develop in a specific research context, in theory/experiment interaction, and in operational practice.
[SUN1993]	Sun, W..; Ling, Y.; Rishe, N.; Deng, Y.	1993	An instant and accurate size estimation method for joins and selections in a retrieval-intensive environment	Proceedings of the 1993 ACM SIGMOD international conference on Management of data table of contents, Washington, D.C., United States, S. 79-88, 1993	This paper proposes a novel strategy for estimating the size of the resulting relation after an equi-join and selection using a regression model. An approximating series representing the underlying data distribution and dependency is derived from the actual data. The proposed method provides an instant and accurate size estimation by performing an evaluation of the series, with no run-time overheads in page faults and space, and with negligible CPU overhead. In contrast, the popular sampling methods incur run-time overheads in page faults (for sampling), CPU time and space. These overheads of sampling methods increase the response time of processing a query. The results of a comprehensive experimental study are also reported, which demonstrate that the estimation accuracy by the proposed method is comparable with that of the sampling methods which are believed to provide the most accurate estimation. The proposed method seems ideal for retrieval-intensive database and

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					information systems. Since the overheads involved in deriving the approximating series are fairly moderate, we believe that this method is also an extremely competent method when moderate or periodical updates are present.
[TER1997]	Terveen, L.; Hill, W.; Amento, B.; McDonald, D.; Creter, J.	1997	PHOAKS – a system for sharing recommendations	Communications of the ACM, vol. 40, no. 3, pages 66-72, 1997	-
[TOP2000]	div.	2000	ISO/IEC 13250: 2000 Topic Maps, Jan, 2000.	http://www.topicmaps.org/	ISO/IEC 13250:2003 (2nd edition) specifies two syntaxes for the interchange of Topic Maps. One of these syntaxes is based on the ISO/IEC 10744:1997 (HyTime) meta-DTD (meta Document Type Definition), and it is itself specified as a meta-DTD. The other, called XTM (XML Topic Maps), is specified as an eXtensible Markup Language (XML) DTD. Both syntaxes allow the expression of: - associations between subjects of discourse and zero or more of their names, to be used in various contexts, - associations between subjects and occurrences (pieces of relevant addressable information), - user-defined associations among any arbitrary subjects of discourse, - user-defined subjects which are classes of subjects, including classes of associations and the distinct roles that are played in instances of associations, and - scopes, expressed as sets of subjects, within the context of which specific associations are meant to be understood as valid and/or relevant.
[TVE1977]	Tversky, A.	1977	Features of Similarity	Psychological Review, vol.84, no.4, S. 327-352, 1977,	The metric and dimensional assumptions that underlie the geometric representation of similarity are questioned on both theoretical and empirical grounds. A new set-theoretical approach to similarity is developed in which objects are represented as collections of features, and similarity is described as a feature matching process. Specifically, a set of qualitative assumptions is shown to imply the contrast model, which expresses the similarity between objects as a linear combination of the measures of their common and distinctive features. Several predictions of the contrast model are tested in studies of similarity with both semantic and perceptual stimuli. The model is used to uncover, analyze, and explain a variety of empirical phenomena such as the role of common and distinctive features, the relations between judgments of similarity and difference, the presence of asymmetric similarities, and the effects of context on judgments of similarity. The contrast model generalizes standard representations of similarity data in terms of clusters and trees. It is also used

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					to analyze the relations of prototypicality and family resemblance.
[TUR2001]	Turpin, A.H.; Hersh, W.	2001	Why batch and user evaluations do not give the same results	Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, S. 225-231, New Orleans, Louisiana, United States, 2001	Much system-oriented evaluation of information retrieval systems has used the Cranfield approach based upon queries run against test collections in a batch mode. Some researchers have questioned whether this approach can be applied to the real world, but little data exists for or against that assertion. We have studied this question in the context of the TREC Interactive Track. Previous results demonstrated that improved performance as measured by relevance-based metrics in batch studies did not correspond with the results of outcomes based on real user searching tasks. The experiments in this paper analyzed those results to determine why this occurred. Our assessment showed that while the queries entered by real users into systems yielding better results in batch studies gave comparable gains in ranking of relevant documents for those users, they did not translate into better performance on specific tasks. This was most likely due to users being able to adequately find and utilize relevant documents ranked further down the output list.
[UML1998]	Umlauf, Konrad	1998	Regeln für den Schlagwortkatalog (RSWK) und Einführung in die Regeln für den Schlagwortkatalog RSWK	Berliner Handreichungen zur Bibliotheks- und Informationswissen schaft, ISSN 1438-7662, Heft 66	Die Einführung macht mit den Regeln für den Schlagwortkatalog RSWK (nach der 3. Auflage 1998) bekannt. Die Grundregeln und die wichtigsten Regeln für die einzelnen Schlagwortkategorien einschließlich Sonderregeln für Kunst- und Bauwerke, Werke der Literatur und Rechtsmaterien werden dargestellt. Die Probleme bei verbaler Sacherschließung allgemein werden behandelt und hieraus Anforderungen an Regelwerke und ihre Anwendungen abgeleitet. Die Anwendung der RSWK einschließlich der Anwendung in Verbünden und der Benutzung der Schlagwortnormdatei SWD werden umrissen. Kommentierte Literaturhinweise. Es folgen Übungen zur Beschlagwortung mit Hinweisen, wie die erfolgte Beschlagwortung überprüft werden kann.
[VIT2005]	Vitányi, P.; Li, M.	2005	Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity	Gekürzte ältere Fassung: IEEE Transactions of Information Theory, 46:2, S. 446-464	The relationship between the Bayesian approach and the minimum description length approach is established. We sharpen and clarify the general modeling principles MDL and MML, abstracted as the ideal MDL principle and defined from Bayes's rule by means of Kolmogorov complexity. The basic condition under which the ideal principle should be applied is encapsulated as the Fundamental Inequality, which in broad terms states that the principle is valid when the data are random, relative to every contemplated hypothesis and also these hypotheses are random relative to the (universal) prior. Basically, the ideal principle states that the prior probability associated with the hypothesis should be given by the algorithmic universal probability, and the sum of the log universal probability of the model plus the log of the probability of the data

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					given the model should be minimized. If we restrict the model class to the finite sets then application of the ideal principle turns into Kolmogorov's minimal sufficient statistic. In general we show that data compression is almost always the best strategy, both in hypothesis identification and prediction.
[VOO2002]	Voorhees, E.	2002	The Philosophy of Information Retrieval Evaluation	Proceedings of the 2nd Workshop of the Cross-Language Evaluation Forum, CLEF 2001, Darmstadt, Germany, 2002	Evaluation conferences such as TREC, CLEF, and NTCIR are modern examples of the Cranfield evaluation paradigm. In the Cranfield paradigm, researchers perform experiments on test collections to compare the relative effectiveness of different retrieval approaches. The test collections allow the researchers to control the effects of different system parameters, increasing the power and decreasing the cost of retrieval experiments as compared to user-based evaluations. This paper reviews the fundamental assumptions and appropriate uses of the Cranfield paradigm, especially as they apply in the context of the evaluation conferences.
[WAL1998]	Walker, S.; Robertson, S.E.; Boughanem, M.; Jones, G.J.F.; Sparck Jones, K.	1998	Okapi at TREC-6	Proceedings of the sixth Text REtrieval Conference (TREC-6), S.Gaithersburg, Maryland, United States, 1997.	The Okapi Basic Search System (BSS) is a set-oriented ranked output system designed primarily for probabilistic-type retrieval of textual material using inverted indexes. There is a family of built-in weighting functions. In addition to weighting and ranking facilities, it has the usual Boolean and quasi-Boolean (positional) operations and a number of non-standard set operations. Indexes are of a fairly conventional inverted type. For VLC, we were interested to find out whether the Okapi BSS could handle more than 20 gigabytes of text and 8 million documents without major modification. There was no problem with data structures, but one or two system parameters had to be altered. In the interests of speed and because of limited disk space, indexes without full positional information were used. This meant that it was not possible to use passage searching. Apart from this, the runs were done in the same way as the ad hoc, but with parameters intended to maximize precision at 20 documents. Several pairs of runs were done, but only one-based on the full topic statements-was submitted. For QSDR, some small-scale experiments were run at Cambridge, using Okapi-type methods, with the QSDR data. These tests gave some indication (albeit qualified by the size of the experiment) that the methods are sufficiently robust to give satisfactory performance with appropriate tuning.
[WIT1968]	Wittgenstein, L., Schult, J.	1968	Philosophische Untersuchungen	Suhrkamp, ISBN: 3518223720, 1968	Ludwig Wittgenstein (1889–1951) schrieb zwei Hauptwerke, früh die strenge Logisch-philosophische Abhandlung (BS 1322), spät die offeneren, lebendig in immer neuen Anläufen vorgetragenen Philosophischen Untersuchungen, mit denen der Begriff des »Sprachspiels« in die Welt gekommen ist: »Man kann für eine große Klasse von Fällen der Benützung des Wortes ›Bedeutung‹ –

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
					wenn auch nicht für alle Fälle seiner Benützung – dieses Wort so erklären: Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.« In der Bibliothek Suhrkamp erscheint nun die erste Leseausgabe der Philosophischen Untersuchungen, die sich nach der definitiven, 2001 von Joachim Schulte herausgegebenen »kritisch-genetischen« Edition richtet.d
[WOL2002]	Wolber, D.; Kepe, M.; Ranitovic, I.	2002	Exposing document context in the personal web	Proceedings of the 7th international conference on Intelligent user interfaces, San Francisco, California, United States, Session: Full Papers, S.151-158, 2002, ISBN 1581134592	Reconnaissance agents show context by displaying documents with similar content to the one(s) the user currently has open. Research paper search engines show context by displaying documents that cite or are cited by the currently open document(s). We present a tool that applies such ideas to the personal web, that is, the space rooted in user documents but tightly connected to web documents as well. The tool organizes the personal web with a single topic hierarchy based on direct links, instead of the traditional file, bookmark, and (hidden) direct link hierarchies. The tool allows a user to easily navigate through related user and web documents, no matter whether the documents are related by directory-document, bookmark-document, direct-link, or even similar content relationships.
[WOL2004]	Wolber, D.; Brooks, C.H.	2004	Associative sources and agents for zero-input publishing	Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, New York, NY, United States, Session: Posters, S. 494-495, 2004, ISBN 1581139128	This paper presents an associative agent that allows seamless navigation from one's own personal space to third-party associative sources, as well as the personal spaces of other users. The agent provides users with access to a dynamically growing list of information sources, all of which follow a common associative sources API that we have defined. The agent also allows users act as sources themselves and take part in peer-to peer knowledge sharing.
[WUH1981]	Wu, H.; Salton, G	1981	A comparison of search term weighting: term relevance vs. inverse document frequency	Proceedings of the 4th annual international ACM SIGIR conference on Information storage and retrieval, S. 30-39, 1981	The term relevance weighting method has been shown to produce optimal information retrieval queries under well-defined conditions. The parameters needed to generate the term relevance factors cannot unfortunately be estimated accurately in practice; futhermore, in realistic test situations, it appears difficult to obtain improved retrieval results using the term relevance weights over much simpler term weighting systems such as, for example, the inverse document frequency weights.It is shown in this study that the inverse document frequency weights and the term relevance weights are closely related over a wide range of the frequency spectrum. Methods are introduced for estimating the term relevance weights, and experimental results are given comparing the inverse document frequency with the estimated term relevance weights.
[WUH1981]	Wu, H.; Salton, G.	1981	A comparison of search term	Proceedings of the 4th annual	The term relevance weighting method has been shown to produce optimal information retrieval queries under well-

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
			weighting: term relevance vs. inverse document frequency	international ACM SIGIR conference on Information storage and retrieval: theoretical issues in information retrieval table of contents, S. 30-39, Oakland, California, UNITED STATES, 1981	defined conditions. The parameters needed to generate the term relevance factors cannot unfortunately be estimated accurately in practice; furthermore, in realistic test situations, it appears difficult to obtain improved retrieval results using the term relevance weights over much simpler term weighting systems such as, for example, the inverse document frequency weights. It is shown in this study that the inverse document frequency weights and the term relevance weights are closely related over a wide range of the frequency spectrum. Methods are introduced for estimating the term relevance weights, and experimental results are given comparing the inverse document frequency with the estimated term relevance weights.
[XUE2005]	Xue, G; Lin, C; Yang, Q.; Xi, W.; Zeng, H.; Yu, Y.; Chen, Z.	2005	Scalable collaborative filtering using cluster-based smoothing	Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, Brazil, Session: Filtering, S. 114-121, 2005, ISBN:1595930345	Memory-based approaches for collaborative filtering identify the similarity between two users by comparing their ratings on a set of items. In the past, the memory-based approach has been shown to suffer from two fundamental problems: data sparsity and difficulty in scalability. Alternatively, the model-based approach has been proposed to alleviate these problems, but this approach tends to limit the range of users. In this paper, we present a novel approach that combines the advantages of these two approaches by introducing a smoothing-based method. In our approach, clusters generated from the training data provide the basis for data smoothing and neighborhood selection. As a result, we provide higher accuracy as well as increased efficiency in recommendations. Empirical studies on two datasets (EachMovie and MovieLens) show that our new proposed approach consistently outperforms other state-of-art collaborative filtering algorithms.
[YAN1999]	Yan, T.W.; Garcia-Molina, H.	1999	The SIFT Information Dissemination System	ACM Transactions on Database Systems, vol. 24, no.4, S. 529-565", 1999	Information dissemination is a powerful mechanism for finding information in wide-area environments. An information dissemination server accepts long-term user queries, collects new documents from information sources, matches the documents against the queries, and continuously updates the users with relevant information. This paper is a retrospective of the Stanford Information Filtering Service (SIFT), a system that as of April 1996 was processing over 40,000 worldwide subscriptions and over 80,000 daily documents. The paper describes some of the indexing mechanisms that were developed for SIFT, as well as th evaluations that were conducted to select a scheme to implement. It also describes the implementation of SIFT, and experimental results for the actual system. Finally, it also discusses and experimentally evaluates techniques for distributing a service such as SIFT for added performance and availability.
[YAR1993]	Yarowsky, D.	1993	One sense per collocation	In the Proceedings of ARPA Human	Previous work [Gale, Church and Yarowsky, 1992] showed that with high

Kürzel	Autor(en)	Jahr	Titel	Veröffentlichung	Abstract
				Language Technology Workshop, 1993	probability a polysemous word has one sense per discourse. In this paper we show that for certain definitions of collocation, a polysemous word exhibits essentially only one sense per collocation. We test this empirical hypothesis for several definitions of sense and collocation, and discover that it holds with 90-99% accuracy for binary ambiguities. We utilize this property in a disambiguation algorithm that achieves precision of 92% using combined models of very local context.
[ZAW2004]	Zawodny, J. Balling, D.	2004	High Performance MySQL Optimierung, Datensicherung, Replikation & Lastverteilung	O'Reilly, S.79 ff., ISBN 3897213885	Einführungen in MySQL gibt es viele. Aber wer größere MySQL-Server betreut, die verlässlich laufen müssen, egal was Programmierer oder Benutzer auf sie loslassen, der braucht weiter reichende Informationen. Jeremy Zawodny ist MySQL-Guru bei Yahoo! und hat mit seinem Co-Autor Derek Balling umfangreiches Insider-Wissen für Profis gesammelt. Um die Verlässlichkeit, Skalierbarkeit und Performance von MySQL-Servern sicherzustellen oder zu steigern, braucht man die richtigen Werkzeuge und die geeigneten Techniken. In High Performance MySQL erfahren Sie, wie sich die Serverlast geschickt verteilen lässt, wie man die Performance steigern kann, wie sich das höchste Maß an Sicherheit erreichen lässt und wie man sinnvolle Backups macht. Mit diesem Rüstzeug steht dem reibungslosen Funktionieren Ihrer Site nichts mehr im Wege.
[ZHA2001]	Zhang, B-T; Seo, Y-W	2001	Personalized Web-Document Filtering Using Reinforcement Learning	Applied Artificial Intelligence, vol.15, no.7, S. 665-685", 2001"	Document filtering is increasingly deployed in Web environments to reduce information overload of users. We formulate online information filtering as a reinforcement learning problem, i.e., TD(0). The goal is to learn user profiles that best represent information needs and thus maximize the expected value of user relevance feedback. A method is then presented that acquires reinforcement signals automatically by estimating user's implicit feedback from direct observations of browsing behaviors. This ``learning by observation'' approach is contrasted with conventional relevance feedback methods which require explicit user feedbacks. Field tests have been performed that involved 10 users reading a total of 18,750 HTML documents during 45 days. Compared to the existing document filtering techniques, the proposed learning method showed superior performance in information quality and adaptation speed to user preferences in online filtering.
[ZIP1949]	Zipf, G.K.	1949	Human Bevavior and the Principle of Least Effort	Cambridge, MA, Adisson-Wesley.	-