

Vorlesung Netzbasierte Informationssysteme (WS 2004/05) Information Retrieval für das Web

Robert Tolksdorf
Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme
mailto:tolk@inf.fu-berlin.de
<http://www.robert-tolksdorf.de>
<http://nbi.inf.fu-berlin.de>

[1] © Robert Tolksdorf, Berlin

Überblick

[2] © Robert Tolksdorf, Berlin

Überblick

- Information Retrieval
- Indexerstellung
- Multimedia Indexing
- Collaborative Filtering

[3] © Robert Tolksdorf, Berlin

Information Retrieval

[4] © Robert Tolksdorf, Berlin

Information Retrieval

- Aufgabe des Information Retrievals: Technologien bereitstellen, die für eine Anfrage relevante Dokumente aus einer Sammlung von Dokumenten heraussuchen
- Dokumente werden üblicherweise so vorverarbeitet und repräsentiert dass Anfrage einfach zu beantworten sind
- Bei Suchmaschinen üblich:
 - Volltextindex gesammelter Seiten erstellen
 - Anfragen an den Volltextindex weiterleiten
 - Ergebnisse ordnen
 - Verweise auf Ursprungsdokumente an Nutzer ausliefern

[5] © Robert Tolksdorf, Berlin

Problemstellung

- Information Retrieval:
 - Dokumentensammlung vorhanden
 - Nutzer führen Suchen durch
 - Wollen Untermenge der Dokumente als Ergebnisse
- Im Gegensatz zu Datenbanken:
 - Daten nicht präzise strukturiert
 - Anfragen nicht präzise strukturiert
- *Indexing* ist Erstellung einer Dokumentenrepräsentation durch Zuordnung von Beschreibungstermen
- Auf Basis dieser Terme wird Relevanz eines Dokuments für eine Anfrage bestimmt

[6] © Robert Tolksdorf, Berlin

Terme in IR

- Zwei Arten von Termen in IR
- *Objektive* Terme:
 - Außerhalb des eigentlichen Inhalts
 - Beispiele: Autorenname, URL etc.
 - Einfach und klar zuzuordnen
- *Nichtobjektive* Terme / *Inhaltsterme*:
 - Beschreiben Informationen des Dokumenteninhalts
 - Schwierig zuzuordnen
 - Hauptaufgabe des Indexing

[7] © Robert Tolksdorf, Berlin

Masse für Termzuordnung

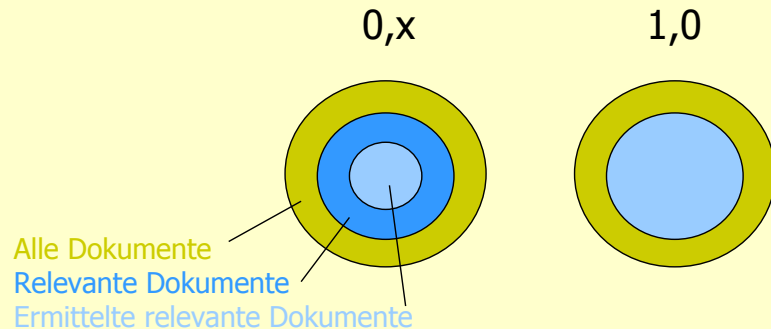
- *Indexing exhaustivity*:
 - Grad zu dem Inhalt durch Indexing erfasst wird
 - Hohe Ausschöpfung: Viele Terme zugeordnet
 - Geringe Ausschöpfung: Weniger Terme zugeordnet
- *Term specificity*:
 - „Breite“ der Terme beim Indexen
 - Breite Terme erfassen viele relevante und viele irrelevante Dokumente bei einer Anfrage
 - „Enge“ Terme erfassen weniger Dokumente und viele relevante nicht

[8] © Robert Tolksdorf, Berlin

Recall und Precision

- *Recall (Nachweisquote):*
Wie gut findet das System relevante Dokument wieder?

$$\text{recall} = \frac{\text{Anzahl ermittelte relevante Dokumente}}{\text{Anzahl relevante Dokumente}}$$

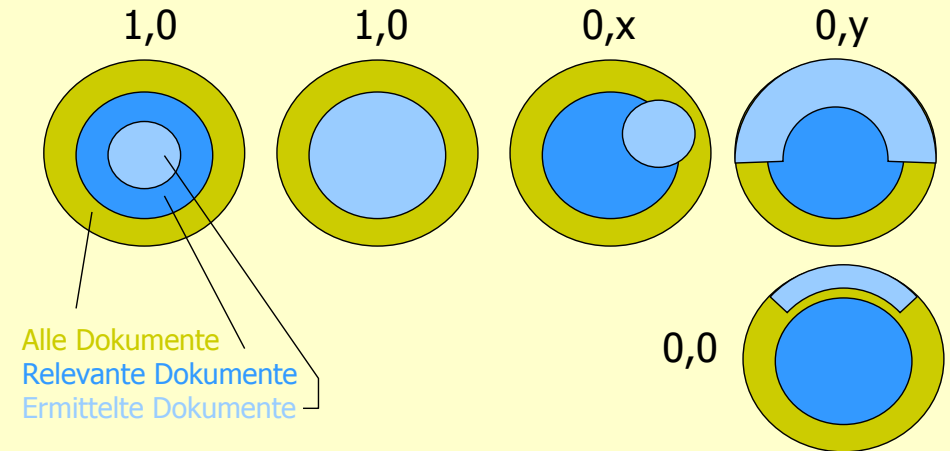


[9] © Robert Tolksdorf, Berlin

Recall und Precision


- *Precision (Präzision):* Wie gut ist die Antwortmenge

$$\text{precision} = \frac{\text{Anzahl ermittelte relevante Dokumente}}{\text{Anzahl ermittelte Dokumente}}$$



[10] © Robert Tolksdorf, Berlin

Idealmasse

- Ziel Recall und Precision bei 1 ()
- Spezifische Terme → höhere Precision, niedrigeres Recall
- Unspezifische Terme → höheres Recall, niedrigere Precision
- Effizienz/Performance eines IR Modells durch Precision auf unterschiedlichen Recall Levels gemessen:
„Precision ist x bei 30% Recall, y bei 50% Recall“

[11] © Robert Tolksdorf, Berlin

IR Modelle

- Vier Eigenschaften
 - Repräsentation von Dokumenten und Anfragen
 - Feststellung der Relevanz eines Dokuments zu einer Anfrage
 - Ordnung der Ergebnismenge
 - Beachtung von Relevanz-Feedback durch Nutzer
- Vier Klassen von Modellen
 - Mengentheoretisch
 - Algebraisch
 - Stochastisch
 - Mischformen

[12] © Robert Tolksdorf, Berlin

Dokumentenrepräsentation

- Menge der Terme $T = \{t_1, \dots, t_n\}$
- Jedes Dokument d_j wird durch einen Vektor repräsentiert:
 $d_j = (w_{1,j}, \dots, w_{n,j})$
- $w_{i,j}$ ist ein Gewicht für den Term t_i im Dokument d_j
- Gesamtmenge der Dokumente ist D
- Ähnlichkeitsmaß sim beschreibt Ähnlichkeit eines Dokuments mit einer Anfrage

[13] © Robert Tolksdorf, Berlin

Mengentheoretisch / Boolesches Retrieval

- Dokumente als Vektor von Indextermen repräsentiert
 - wahr wenn im Dokument vorhanden, falsch sonst
 - Gewichte $w_{i,j}$ also 0 oder 1
 - verstanden als Boolesche Variablen
- Anfragen als Boolesche Ausdrücke
 - Terme sind Anfragen
 - $(q_1 \text{ AND } q_2)$, $(q_1 \text{ OR } q_2)$, $(\neg q_1)$ sind Anfragen
- Dokument ist relevant, wenn Anfrageausdruck belegt mit Dokumentenrepräsentation wahr ergibt
- Ähnlichkeitsmaß ist also auch boolesch

[14] © Robert Tolksdorf, Berlin

Mengentheoretisch / Boolesches Retrieval

- $T = (\text{„heute“}, \text{„ist“}, \text{„dienstag“}, \text{„vorlesung“}, \text{„nicht“})$
- Dokumente d_1 : „heute ist dienstag“, d_2 : „heute ist vorlesung“, d_3 : „dienstag ist vorlesung“

	heute	ist	dienstag	vorlesung	nicht
d_1	1	1	1	0	0
d_2	1	1	0	1	0
d_3	0	1	1	1	0

	ist	dienstag AND vorlesung	heute OR dienstag	NOT vorlesung
d_1	1	1	1	1
d_2	1	0	1	0
d_3	1	1	1	0

[15] © Robert Tolksdorf, Berlin

Mengentheoretisch / Boolesches Retrieval

- Performance eher schlecht
 - Recall ist niedrig:
Durch Fehlen eines einzigen Terms gelten Dokumente als irrelevant
- Keine Ordnung in Ergebnissen möglich
- Wenig intuitive Antworten
(ein einziger fehlender Term führt zu Irrelevanz)

[16] © Robert Tolksdorf, Berlin

DNF

- Anfrageterm lässt sich auch als Vektor darstellen
- q: nicht UND (heute ODER dienstag)

	heute	ist	dienstag	vorlesung	nicht
	1	0	1	0	1
ODER	0	0	1	0	1
ODER	1	0	0	0	1

- d₄: „heute am dienstag ist die vorlesung nicht“
ist genauso bewertet wie
- d₅: „heute ist die vorlesung nicht“
- Intuitiv aber ähnlicher der Anfrage

[17] © Robert Tolksdorf, Berlin

Coordinate Match

- Ähnlichkeitsmaß aus Anzahl der Übereinstimmungen

$$sim(d_j, q) = \sum_{i=1}^n w_{i,j} * p_i$$

- $sim(d_4, q) = 3 > sim(d_5, q) = 2$

[18] © Robert Tolksdorf, Berlin

Fuzzy Mengen Modell

- Gleiches Vorgehen wie beim Booleschen Retrieval
- Operatoren entsprechend umdefiniert
- Vergleichbar schlechte Unterscheidungsmöglichkeiten

- Mengentheoretische Modelle gut zu implementieren
 - geringer Platzbedarf für Dokumentenrepräsentation
 - geringer Rechenbedarf beim Indexing
 - geringer Rechenbedarf beim Ordnen

[19] © Robert Tolksdorf, Berlin

Algebraisch / Vector Space Modell

- Dokumente und Anfragen repräsentiert durch Vector in einem n-dimensionalen Raum
- Dimensionen durch Terme gegeben
- Gewichtet und normalisiert
- Relevanz durch Ähnlichkeitsmaß von Anfrage und Dokument gegeben und geordnet
- Sehr einfaches Modell
- Ausdrucksmächtigkeit boolescher Ausdrücke nicht vorhanden

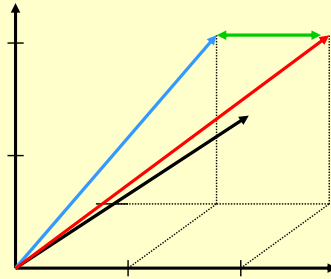
[20] © Robert Tolksdorf, Berlin

Ähnlichkeit im Vektorraum

- $d_j = (w_{1,j}, \dots, w_{n,j})$ als Dokument
- $q = (q_1, \dots, q_n)$ ist Anfrage
- Terme sind gewichtet
- d_j und q sind Punkte in einem n -dimensionalen Raum
- Ähnlichkeitsmaß als Abstand zwischen den Punkten (Euklidische Distanz)

$$sim(d_j, q) = \sqrt{\sum |w_{i,j} - q_i|}$$

- Problem: Je mehr Terme, je größer der Abstand
- Dokumente haben mehr Terme als Anfragen
- Abstand immer groß



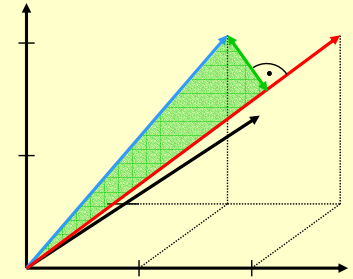
[21] © Robert Tolksdorf, Berlin

Ähnlichkeit im Vektorraum

- Skalarprodukt als Ähnlichkeitsmaß:

$$sim(d_j, q) = w_{1,j} * q_1 + \dots + w_{n,j} * q_n$$

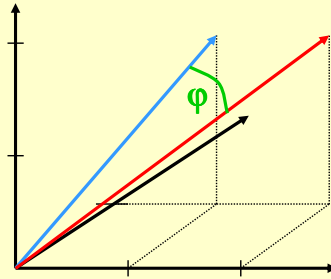
- Entspricht Coordinate Match bei Gewichten 0 und 1
- Problem: Je mehr Terme, je größer der Vektor, je größer das Skalarprodukt



[22] © Robert Tolksdorf, Berlin

Ähnlichkeit im Vektorraum

- Cosinusmaß: Nimmt Unterschied der Richtung der Vektoren, also den Winkel zwischen Dokument und Anfrage



$$\begin{aligned} \cos \varphi * |d_j| * |q| &= w_{1,j} * q_1 + \dots + w_{n,j} * q_n \\ sim(d_j, q) &= \cos \varphi \\ &= \frac{d_j \bullet q}{|d_j| * |q|} \\ &= \frac{\sum w_{i,j} * q_i}{\sqrt{\sum w_{i,j}^2} * \sqrt{\sum q_i^2}} \end{aligned}$$

[23] © Robert Tolksdorf, Berlin

Termgewichtung

- Woher kommt die Gewichtung der Terme bei der Dokumentenrepräsentation?
 - Manuell: Nicht skalierbar, nicht objektiv
 - Automatisch: Heuristik, kein Verstehen des Inhalts

[24] © Robert Tolksdorf, Berlin

Automatische Gewichtung

- *Termfrequenz*: Ein in einem Dokument häufiger Term ist charakteristischer als ein seltener Term
 tf_{ij} : Häufigkeit von Term T_j im Dokument i
- Führt zu hohem Recall
- Leicht zu beeinflussende Dokumentenbewertung: Wiederholung eines Wortes
- *Document frequency*: Seltener Term ist für eine Dokument charakteristisch in dem er häufig auftritt
- df_j : Anzahl des Auftretens von T_j in N Dokumenten
- *Inverse document frequency*: Wie stark ist T_j charakteristisch
- $idf_j = \log(N/df_j)$
- auch: $idf_j = \log(df_{\max}/df_j)$ und andere

[25] © Robert Tolksdorf, Berlin

Automatische Gewichtung

- *tfidf* Regel:
Anzahl des Vorkommens eines Terms gewichtet mit dessen Charakterisierungsfähigkeit
 $w_{ij} = tf_{ij} * \log(N/df_j)$
- Es existieren sehr viele weitere Gewichtungsmöglichkeiten
- Gewichtung wählen
 - für Dokumentenvektoren
 - für Anfragevektoren

[26] © Robert Tolksdorf, Berlin

Probabilistische Modelle

- Vectorspace-Modell:
 - Dimensionen sind orthogonal
 - Dimensionen sind unabhängig
 - Ähnlichkeitsabstand frei gewählt
 - Für jeden Term wird Wahrscheinlichkeit bestimmt
 - N Dokumente, R sind relevant, R_t enthalten t , t kommt in f_t Dokumenten vor
 - $\text{pr}[t \in D_j \mid D_j \text{ relevant}] = R_t/R$
 - $\text{pr}[t \in D_j \mid D_j \text{ irrelevant}] = (f_t - R_t)/(N - R)$
- $$w_t = \log \frac{R_t / (R - R_t)}{(f_t - R_t) / (N - f_t - (R - R_t))}$$
- $w_i < 0$: Auftreten von Term i zeigt Relevanz des Dokuments an
 - $w_i > 0$: Auftreten von Term i zeigt Irrelevanz des Dokuments an

[27] © Robert Tolksdorf, Berlin

Relevanz-Feedback

- Nutzer verfeinern Anfrage nach und nach
- Relevanz-Feedback: Nutzer bewerten Ergebnislänge
- Genutzt um Performance zu verbessern
- Two-Level feedback:
Ergebnisdokument ist relevant oder nicht relevant
- Multi-Level feedback:
 - Ergebnisdokument ist relevant, etwas relevant, irrelevant
 - Ergebnisdokument ist mehr/weniger relevant als anderes Dokument

[28] © Robert Tolksdorf, Berlin

Relevanz-Feedback

- Verwendung des Feedbacks:
 - Modifikation der Anfragerepräsentation
 - Modifikation der Dokumentenrepräsentation
- Annahme: Relevante Dokumente sind ähnlich

Modifikation der Anfrage Repräsentation

- Änderung der Term-Gewichte
 - Addieren der Vektoren relevanter Dokumente
 - Subtrahieren der Vektoren irrelevanter Dokumente
 - Liefert mehr Dokumente, die den relevanten Dokumenten ähnlich sind
- Query Expansion
 - Hinzufügen weiterer Terme aus der Menge der relevanten Dokumente zur Anfrage
 - Sortiert nach diversen Massen
- Query Splitting
 - Falls relevante Dokumente inhomogen sind oder irrelevante Dokumente verstreut auftreten
 - Gruppen ähnlicher Dokumente aus relevanten bilden
 - Je Gruppe mit Änderung der Term-Gewichte oder Query Expansion arbeiten

Modifikation der Dokumentenrepräsentation

- Anpassen der Vektoren der als relevant bewerteten Dokumente in Richtung Anfragevektor
- Anpassen der Vektoren der als irrelevant bewerteten Dokumente vom Anfragevektor weg
- „user oriented clustering“
- Einzelne Bewertung darf nicht zu stark beeinflussen

Literatur

- V. Gudivada, V. Raghavan, W. Grosky and R. Kananagottu. Information retrieval on the World Wide Web. IEEE Internet Computing, 1(5), pp. 58-69, September / October 1997.
- Michael W. Berry and Murray Browne: Understanding Search Engines: Mathematical Modeling and Text Retrieval. 1999. Siam.

Indexerstellung

Manuelles Indexieren

- Indexierer ordnen Dokumente per Hand in Kategorien ein oder bestimmen Indexterme

Service	Editoren	Kategorien	Links...	Datum
Open Directory	36000	361000	2.6 million	04/01
LookSmart	200	200000	2.5 million	08/01
Yahoo	>100	n/a	1.5 to 1.8 million	8/00

[<http://www.searchenginewatch.com/reports/directories.html>]

- Yahoo: In „beste“ Kategorie einordnen
- National Library of Medicine: Einordnung in so viele Kategorien des Medical Subject Headings (MeSH) Katalogs wie möglich
- Skalierungsproblem
- Spezialisierung als Ausweg

Automatisches Indexieren

- Notwendig wegen Web-Grösse
 - Crawling
 - Indexing
 - Anfragemanagement

Normalisierung

- Dokumentenvorbereitung
 - Parsieren/Entfernen von HTML
 - Ermittlung indexierungsrelevanter Informationen
 - alt Attribut bei
 - <meta>-Tags
 - lang Attribut
 - ...
 - Umgang mit Zeichenkodierungen
 - Entitäten expandieren
 - Interpunktion entfernen
- Aufteilen in Token
- Stop words entfernen
- Stemming

Limerik Beispiel

```
<html>
<body>
<p>There once was a searcher
  named Hanna<br>
Who needed some info on
  manna.<br>
She put "rye" and "wheat" in her
  query<br>
Along with "potato" or
  "cranbeery," <br>
But no mention of "sourdough" or
  "banana".</p>
<p>
Instead of rye, &nbsp;cranberry,
  or wheat,<br>
The results had more spiritual
  meat.<br>
So Hanna was not pleased,<br>
Nor was her hunger eased,<br>
`Cause she was looking for
  something to eat.</p>
</body></html>
```

There once was a searcher named
Hanna
Who needed some info on manna
She put rye and wheat in her query
Along with potato or cranbeery
But no mention of sourdough or
banana
Instead of rye cranberry or wheat
The results had more spiritual meat
So Hanna was not pleased
Nor was her hunger eased
Cause she was looking for something
to eat

[37] © Robert Tolksdorf, Berlin

Stop words

- Nicht alle Worte sind für den Inhalt eines Dokuments entscheidend (idf so gering, dass sich ihre Weiterverarbeitung nicht lohnt)
- Diese *stop words* werden aus dem Dokument entfernt
- Übrig bleiben *content words*

[38] © Robert Tolksdorf, Berlin

Englischsprachige Stop-Worte

a a's able about above according accordingly across actually after afterwards again against ain't all allow allows almost alone along already also although always am among amongst an and another any anybody anyhow anyone anything anyway anyways anywhere apart appear appreciate appropriate are aren't around as aside ask asking associated at available away awfully b be became because become becomes becoming been before beforehand behind being believe below beside besides best better between beyond both brief but by c c'mon c's came can can't cannot cant cause causes certain certainly changes clearly co com come comes concerning consequently consider considering contain containing contains corresponding could couldn't course currently d definitely described despite did didn't different do does doesn't doing don't done down downwards during e each edu eg eight either else elsewhere enough entirely especially et etc even ever every everybody everyone everything everywhere ex exactly example except f far few fifth first five followed following follows for former formerly forth four from further furthermore g get gets getting given gives go goes going gone got gotten greetings h had hadn't happens

[39] © Robert Tolksdorf, Berlin

Ca. 570 Worte

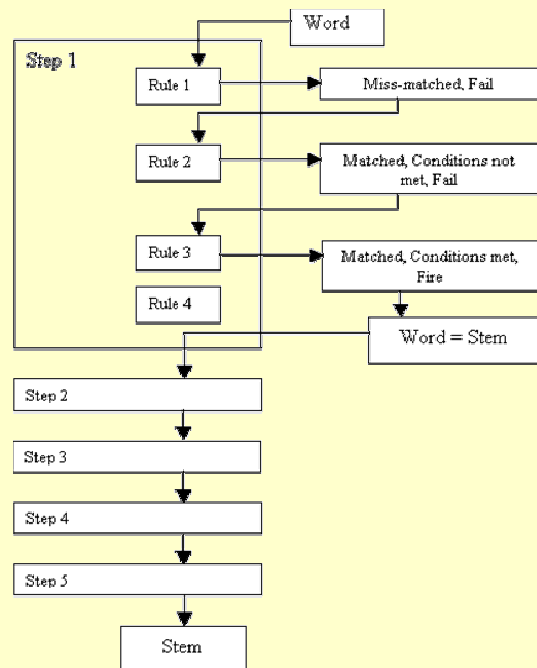
hardly has hasn't have haven't having he he's hello help hence her here here's hereafter hereby herein hereupon hers herself hi him himself his hither hopefully how howbeit however i i'd i'll i'm i've ie if ignored immediate in inasmuch inc indeed indicate indicated indicates inner insofar instead into inward is isn't it it'd it'll it's its itself j just k keep keeps kept know knows known l last lately later latter latterly least less lest let let's like liked likely little look looking looks ltd m mainly many may maybe me mean meanwhile merely might more moreover most mostly much must my myself n name namely nd near nearly necessary need needs neither never nevertheless new next nine no nobody non none noone nor normally not nothing novel now nowhere o obviously of off often oh ok okay old on once one ones only onto or other others otherwise ought our ours ourselves out outside over overall own p particular particularly per perhaps placed please plus possible presumably probably provides q que quite qv r rather rd re really reasonably regarding regardless regards relatively respectively right s said same saw say saying says second secondly see seeing seem seemed seeming seems seen self selves sensible sent serious

[40] © Robert Tolksdorf, Berlin

seriously seven several shall she should shouldn't since six so some somebody somehow someone something sometime sometimes somewhat somewhere soon sorry specified specify specifying still sub such sup sure t t's take taken tell tends th than thank thanks thanx that that's thats the their theirs them themselves then thence there there's thereafter thereby therefore therein theres thereupon these they they'd they'll they're they've think third this thorough thoroughly those though three through throughout thru thus to together too took toward towards tried tries truly try trying twice two u un under unfortunately unless unlikely until unto up upon us use used useful uses using usually uucp v value various very via viz vs w want wants was wasn't way we we'd we'll we're we've welcome well went were weren't what what's whatever when whence whenever where where's whereafter whereas whereby wherein whereupon wherever whether which while whither who who's whoever whole whom whose why will willing wish with within without won't wonder would would wouldn't x y yes yet you you'd you'll you're you've your yours yourself yourselves z zero

- Varianten des gleichen Worts können auf einen Term, den Wortstamm abgebildet werden:
- Beauty, beautiful, beautify -> beaut
- Notwendig *Stemming*
- Sprachabhängig
- Meistbenutzt: Porter Stemmer
Porter, M.F., 1980, An algorithm for suffix stripping, *Program*, **14**(3) :130-137
- Definition plus Implementierungen bei <http://www.tartarus.org/~martin/PorterStemmer/>

Porter Stemmer



Definitionen

- Liste C = ccc... aus Konsonanten
- Liste V = vvv... aus Vokalen
- Jedes Wort: CVCV...C, CVCV...V, VCVC...C, VCVC...V
- [C]VCVC ... [V], [x]: Optional
- [C](VC){m}[V], (x){m}: x in m Wiederholungen
- m: „measure“ des Worts

m=0	TR, EE, TREE, Y, BY
m=1	TROUBLE, OATS, TREES, IVY
m=2	TROUBLES, PRIVATE, OATEN, ORRERY

- Regeln: (condition) S1 -> S2
(m > 1) EMENT ->

Regeln für ersten Schritt

1a:

SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
	ties -> ti
SS -> SS	caress -> caress
S ->	cats -> cat

1b:

(m>0) EED -> EE	feed -> feed
✘ (*v*) ED ->	agreed -> agree
✘ (*v*) ING ->	plastered -> plaster
	bled -> bled
	motoring -> motor
	sing -> sing

Regeln für ersten Schritt

Zusätzlich nach Erfolg mit Regeln ✘:

AT -> ATE	conflat(ed) -> conflate
BL -> BLE	troubl(ed) -> trouble
IZ -> IZE	siz(ed) -> size
(*d and not (*L or *S or *Z)) -> single letter	
	hopp(ing) -> hop
	tann(ed) -> tan
	fall(ing) -> fall
	hiss(ing) -> hiss
	fizz(ed) -> fizz
(m=1 and *o) -> E	fail(ing) -> fail
	fil(ing) -> file

1c:

(*v*) Y -> I	happy -> happi
	sky -> sky

Regeln für zweiten Schritt

2:

(m>0) ATIONAL -> ATE	relational -> relate
(m>0) TIONAL -> TION	conditional -> condition
	rational -> rational
(m>0) ENCI -> ENCE	valenci -> valence
(m>0) ANCI -> ANCE	hesitanci -> hesitance
(m>0) IZER -> IZE	digitizer -> digitize
(m>0) ABLI -> ABLE	conformabli -> conformable
(m>0) ALLI -> AL	radicalli -> radical
(m>0) ENTLI -> ENT	differentli -> different
(m>0) ELI -> E	vileli -> vile
(m>0) OUSLI -> OUS	analogousli -> analogous
(m>0) IZATION -> IZE	vietnamization -> vietnamize
(m>0) ATION -> ATE	predication -> predicate
(m>0) ATOR -> ATE	operator -> operate
(m>0) ALISM -> AL	feudalism -> feudal
(m>0) IVENESS -> IVE	decisiveness -> decisive
(m>0) FULNESS -> FUL	hopefulness -> hopeful
(m>0) OUSNESS -> OUS	callousness -> callous
(m>0) ALITI -> AL	formaliti -> formal
(m>0) IVITI -> IVE	sensitiviti -> sensitive
(m>0) BILITI -> BLE	sensibiliti -> sensible

Regeln für dritten Schritt

3:

(m>0) ICATE -> IC	triplicate -> triplic
(m>0) ATIVE ->	formative -> form
(m>0) ALIZE -> AL	formalize -> formal
(m>0) ICITI -> IC	electriciti -> electric
(m>0) ICAL -> IC	electrical -> electric
(m>0) FUL ->	hopeful -> hope
(m>0) NESS ->	goodness -> good

Regeln für vierten Schritt

4:

(m>1) AL	->	revival	->	reviv
(m>1) ANCE	->	allowance	->	allow
(m>1) ENCE	->	inference	->	infer
(m>1) ER	->	airliner	->	airlin
(m>1) IC	->	gyroscopic	->	gyroscop
(m>1) ABLE	->	adjustable	->	adjust
(m>1) IBLE	->	defensible	->	defens
(m>1) ANT	->	irritant	->	irrit
(m>1) EMENT	->	replacement	->	replac
(m>1) MENT	->	adjustment	->	adjust
(m>1) ENT	->	dependent	->	depend
(m>1 and (*S or *T)) ION	->	adoption	->	adopt
(m>1) OU	->	homologou	->	homolog
(m>1) ISM	->	communism	->	commun
(m>1) ATE	->	activate	->	activ
(m>1) ITI	->	angulariti	->	angular
(m>1) OUS	->	homologous	->	homolog
(m>1) IVE	->	effective	->	effect
(m>1) IZE	->	bowdlerize	->	bowdler

Regeln für fünften Schritt

5a:

(m>1) E	->	probate	->	probat
		rate	->	rate
(m=1 and not *o) E	->	cease	->	ceas

5b:

(m > 1 and *d and *L)	->	single letter		
		controll	->	control
		roll	->	roll

Effekt

Test mit 10000 Worten

Worte reduziert in Schritt 1	3597
Worte reduziert in Schritt 2	766
Worte reduziert in Schritt 3	327
Worte reduziert in Schritt 4	2424
Worte reduziert in Schritt 5	1373
Worte nicht reduziert	3650

Rest von 6370 Worten -> Vokabular um 1/3 reduziert

Beispiel in 10 Dokumenten aufgeteilt

1. There once was a searcher named Hanna
2. Who needed some info on manna
3. She put rye and wheat in her query
4. Along with potato or cranbeery
5. But no mention of sourdough or banana
6. Instead of rye cranberry or wheat
7. The results had more spiritual meat
8. So Hanna was not pleased
9. Nor was her hunger eased
10. Cause she was looking for something to eat

Extrahierte Terme / Document file

- Nach Stemming und Stopword-Entfernung:

Dokument	Terme
1	searcher, Hanna
2	manna
3	rye, wheat, query
4	potato, cranbeery
5	sourdough, banana
6	rye, cranberry, wheat
7	spiritual, meat
8	Hanna
9	hunger
10	-

[53] © Robert Tolksdorf, Berlin

Termliste für Dokumentenmenge / Dictionary

Term	Global/Document frequency
banana	1
cranb	2
Hanna	2
hunger	1
manna	1
meat	1
potato	1
query	1
rye	2
sourdough	1
spiritual	1
wheat	2

[54] © Robert Tolksdorf, Berlin

Inversion list

- Invertierte Liste der Dokumente
- Gibt an, wo in welchen Dokumenten ein Term vorkommt
- Format hier: (Dokument, Position)
- Verschiedene Optionen für die Repräsentation

Term	Fundstellen
banana	(5,7)
cranb	(4,5); (6,4)
Hanna	(1,7); (8,2)
hunger	(9,4)
manna	(2,6)
meat	(7,6)
potato	(4,3)
query	(3,8)
rye	(3,3); (6,3)
sourdough	(5,5)
spiritual	(7,5)
wheat	(3,5); (6,6)

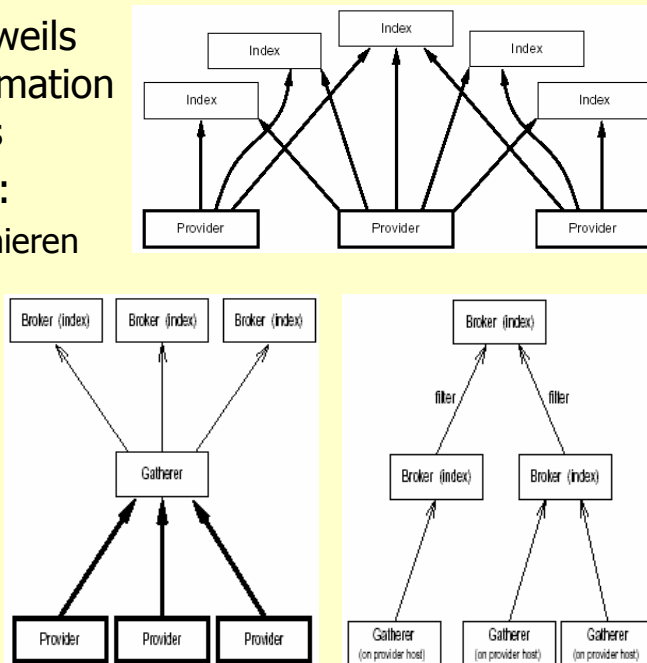
[55] © Robert Tolksdorf, Berlin

Collaborative Indexing

[56] © Robert Tolksdorf, Berlin

Harvest System

- Indexe lesen jeweils komplette Information aus Servern aus
- Harvest-System:
 - Gatherer extrahieren Quellen
 - Broker stellen Index her
 - Broker stellen beantworteten Anfragen
 - Kaskadierte Gatherer



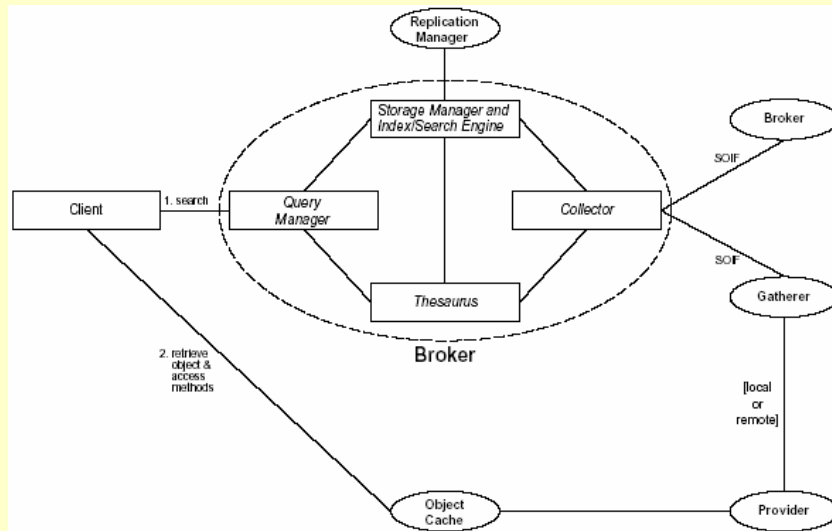
[57] © Robert Tolksdorf, Berlin

Reduzierte Netzlast / erhöhte Flexibilität

- Gatherer auf Server-Seite:
 - Reduzierte Kosten für Auslesen von Informationen
- Methoden:
 - Erstellung von Zusammenfassungen von Informationen
 - Indexierte Informationen komprimiert übertragen
 - Inkrementelles Update geänderter Seiten
 - Cache geholter Informationen
- Kaskadierte Broker:
 - Flexible Zusammenstellung zu themenspezifischen Indexen
 - Mögliche Filter für andere Broker

[58] © Robert Tolksdorf, Berlin

Systemarchitektur



- SOIF (Summary Object Interchange Format)
 - Objektive Metainformationen
 - Extrahierter Text

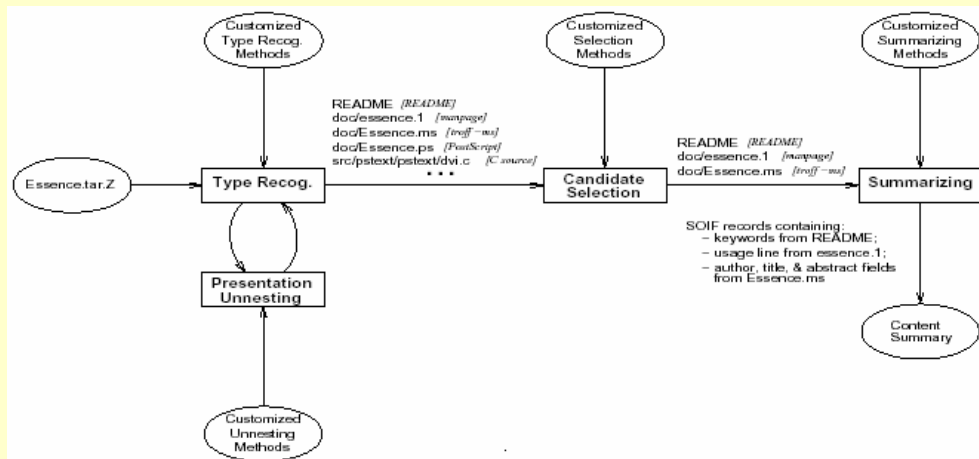
[59] © Robert Tolksdorf, Berlin

Gatherer Architektur

- Untersucht Objekte auf Server
 - aufgrund von Aufzählung in Konfiguration
 - durch Traversierung
- Essence System erstellt Zusammenfassungen
- Konfigurierbar:
 - Wie Typ erkannt wird (Endungskonvention, Inhalt etc.)
 - Welche Objekte indexiert werden (binary/source)
 - Wie Informationen extrahiert werden (<title> etc).
- Verschachtelte Objektrepräsentation (foo.tar.gz)

[60] © Robert Tolksdorf, Berlin

Essence



[61] © Robert Tolksdorf, Berlin

SOIF Beispiel

```
@FILE { http://harvest.cs.colorado.edu/harvest/user-manual/node99.html
update-time{9}: 793962520
description{27}: About this document ...
time-to-live{8}: 14515200
refresh-rate{7}: 2419200
gatherer-name{57}: Networked Information Discovery and Retrieval
gatherer-host{21}: bruno.cs.colorado.edu
gatherer-version{3}: 1.0
type{4}: HTML
file-size{4}: 2551
md5{32}: bea4c43ce6b976b3403c24f6a353f610
author{42}: Darren Hardy
Wed Feb 15 13:27:56 MST 1995
keywords{68}: about document drakos harvest html index latex manual
nikos this user
url-references{274}: user-manual.html
node98.html
node3.html
...
partial-text{601}: About this document ...
...
}
```

[62] © Robert Tolksdorf, Berlin

Broker Architektur

- Collector
 - Erfragt Updates bei Gatherer oder Broker
- Registry
 - Liste der gespeicherten Dokumente
- Storage Manager
 - Gespeicherte Dokumente (Filesystem) als SOIF
- Index/Search Engine
 - IR System für Dokumente
 - Unterschiedliche Systeme:
 - Glimpse: Volletextindex
 - Nebula: Standing queries
- Query Manager
 - Anfragen annehmen und an Index weiterleiten
 - Anfragen annehmen und an anderen Broker weiterleiten

[63] © Robert Tolksdorf, Berlin

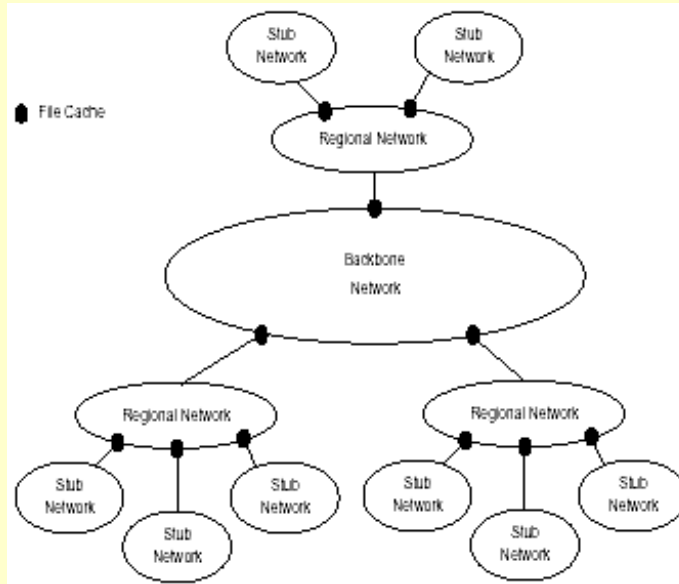
Replikation

- Broker können repliziert werden
- Replikation: Mehrere gleiche Kopien an unterschiedlichen Stellen bereitstellen
- Effekt: Bessere Erreichbarkeit, besser Skalierbarkeit
- Problem: Konsistenzerhaltung bei Änderungen
- Harvest:
 - Schwache Konstanz: Nach Änderungen werden sich die Replikate angleichen
 - Replikat gehört zu Gruppe
 - Gruppenmaster ermittelt notwendige Angleichungen
 - Replikate synchronisieren sich paarweise

[64] © Robert Tolksdorf, Berlin

Caching

- Objekte (von Servern) werden in Caches zwischengespeichert
- Cache ist hierarchisch organisiert



[65] © Robert Tolksdorf, Berlin

Caching-Strategie

- Anfrage nach Objekt geschieht über Cache
- *Hit*: Cache hat Objekt, *Miss*: Cache hat Objekt nicht
- Bei lokalen Miss: Anfrage an
 - Eltern und Nachbarcaches
 - an echo-Port des Herkunftsservers (mit Treffer-Antwort)
- Hit-Antwort von allen
 - vom schnellsten holen
- Hit-Antwort vom Herkunftsserver und Miss-Antworten von Caches
 - Wenn Herkunft langsamer als schnellster Eltern-Cache: Von Eltern-Cache holen lassen
 - Sonst: Von Herkunftsserver holen

[66] © Robert Tolksdorf, Berlin

Multimedia Indexing

- Große Anteile der im Netz verfügbaren Informationen sind kein Text und können nicht in einem Volltextindex gespeichert werden
- Beispiele:
 - Bilder – Fotos, Zeichnungen
 - Audio – Musik, Sprache
 - Video – Filme, Nachrichten
- Oftmals ist Information in mehreren Medien verteilt
 - Beispiel: Nachrichten, Bild und Ton
- Problem
 - Ermittlung von Indextermen
 - Entwurf von Ähnlichkeitsmaßen

[67] © Robert Tolksdorf, Berlin

[68] © Robert Tolksdorf, Berlin

Beispiel: Audio

Audioanfragen

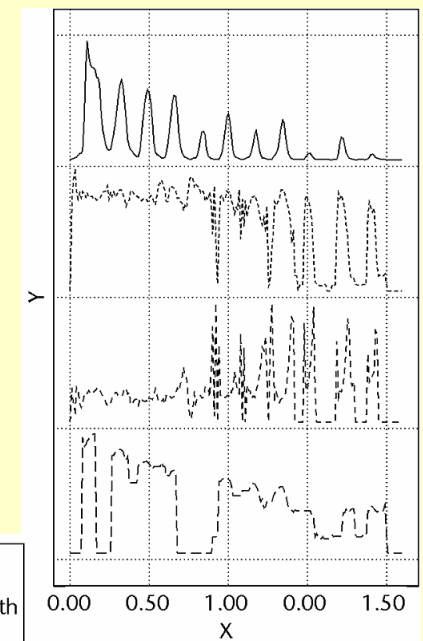
- Anfragbare Eigenschaften von Audio:
 - Ähnliche Sounds („wie eine Elefantenherde“, „Applaus“)
 - Akustische Eigenschaften (Laut, tief, schnell)
 - Subjektive Beschreibung (sanft)
 - Onomatopoeia („eine der zentral verwendeten Tropen im Chat ist die Onomatopoeia“) Lautmalerei (buzz, bumm, klingeling)

Anfragemöglichkeiten

- Stufen von Inhalts-Retrieval von Audio
 - Numerische Anfrage nach Sample-Daten (Text: Stringvergleich)
 - Enthaltensein bestimmter Audioteile unabhängig von deren Geschwindigkeit, Höhe etc. (Text: Ähnlichkeit von Zeichenketten)
 - Vorhandensein bestimmter Maße (Text: tf, etc)
 - Inhalt des Audiostücks (Text: Semantische Suche)

Ansatz

- Akustische Maße ermittelt
- Als N-Vektor repräsentiert
- Dimensionen
 - Lautstärke
 - Takt
 - Helligkeit (Brightness)
 - Bandbreite
 - Harmonie (Abweichung vom harmonischen Spektrum)



Ansatz

- Maße variieren über die Zeit, Dynamik erfasst durch:
 - Mittelmaße
 - Varianz
 - Autokorrelation
 - ...
 - Gewichtet durch Lautstärke

Property	Mean	Variance	Autocorrelation
Loudness	-54.4112	221.451	0.938929
Pitch	4.21221	0.151228	0.524042
Brightness	5.78007	0.0817046	0.690073
Bandwidth	0.272099	0.0169697	0.519198

[73] © Robert Tolksdorf, Berlin

Training

- Beispiele von Vektoren zur Bildung von Klassen genutzt
- Ziel: Klangeigenschaft durch mittleren Vektor μ ausgedrückt

$$\mu = (1/M) \sum_j a[j]$$

($a[j]$: Trainingsvektor, M : Anzahl Trainingsvektoren)

- Dazu: Kovarianzmatrix R

$$R = (1/M) \sum_j (a[j] - \mu)(a[j] - \mu)^T$$

[74] © Robert Tolksdorf, Berlin

Klassifikation neuer Sounds

- Klassifikation neuer Sounds a nach Euklidischem Abstand D vom Mittelvektor

$$D = ((a - \mu)^T R^{-1} (a - \mu))^{1/2}$$

- Schwellwert des Abstands legt fest, ob Sound in einer Klasse ist oder nicht
- Bei überlappenden Klassen die nächstliegende auswählen
- Likelihood L sagt aus, wie sehr ein Sound einer Klasse entspricht

$$L = \exp(-D^2/2)$$

[75] © Robert Tolksdorf, Berlin

Anfragen

- Auf Basis der Entfernung sind nun Anfragen möglich
 - Alle Sounds aus einer Klasse erfragen
 - 20 beste einer Klasse
 - Alle Sounds, die weniger als eine Beispielsound einer Klasse entsprechen
 - Sortierung von Sounds bezüglich einer Klasse
 - ...

[76] © Robert Tolksdorf, Berlin

Literatur

- Erling Wold, Thorn Blum, Douglas Keislar and James Wheaton. Content-Based Classification, Search, and Retrieval of Audio. IEEE Multimedia 3(3), 27-36.

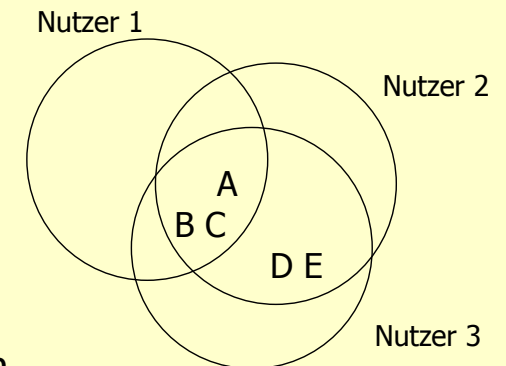
Collaborative Filtering

Collaborative Filtering

- Information Retrieval
 - basiert auf dem *Inhalt* von Dokumenten
 - benötigt Dokumente
 - repräsentiert Dokumente
 - optimiert auf Suchen/Finden von Dokumenten
- Information Filtering
 - basiert auf der *Bewertung* von Inhalten
 - benötigt keinen Zugang zu Dokumenten
 - repräsentiert Nutzer und Bewertungen
 - optimiert auf Filtern von Dokumenten in Strömen
- Collaborative Filtering
 - basiert auf Bewerten durch andere Nutzer

Grundidee

- Ähnliche Nutzer haben ähnliche Vorlieben
- Vorlieben eines Nutzers können genutzt um einem anderen einen Vorschlag zu machen
- Beispiel: A...E sind Produkte, Informationen etc.
- Nutzer 1, 2 und 3 ähneln sich, da sie alle A, B und C mögen/haben
- Für Nutzer 1 sind wahrscheinlich auch D und E relevant



Ähnlichkeit von Nutzern

- Nutzer bewerten Informationen/Produkte
- {1:sehr schlecht, 2: schlecht,...,6:gut, 7:sehr gut}

Nutzer	A	B	C	D	E
U	1	3	4	2	7
V	3		6	5	7
W	2	3	5	1	6

- Entfernung zwischen Vektoren:

$$d_{MSD}(U, V) = \frac{1}{|P_{U \cap V}|} \sum (U_x - V_x)^2$$

- $d(U, V) = 4,25$
- $d(V, W) = 4,75$
- $d(U, W) = 0,80$

[81] © Robert Tolksdorf, Berlin

Korrelation zwischen Nutzern

- Korrelationskoeffizient zwischen Nutzern

$$r_{Pearson}(U, V) = \frac{\sum_{x \in P_{U \cap V}} (U_x - \bar{U})(V_x - \bar{V})}{\sqrt{\sum_{x \in P_{U \cap V}} (U_x - \bar{U})^2 \sum_{x \in P_{U \cap V}} (V_x - \bar{V})^2}}$$

- $r_{uv} > 0$: Positive Korrelation
- $r_{uv} = 0$: Keine Korrelation
- $r_{uv} < 0$: Negative Korrelation

[82] © Robert Tolksdorf, Berlin

Empfehlungen

- Korrelation und Entfernung des Profils neuer Nutzer messen
- Ähnliche Nutzer ermitteln:
 - Korrelation über Schwellwert
 - Abstand unter Schwellwert
- Collaborative Bewertung ist das gewichtete Mittel der Bewertungen ähnlicher Nutzer

$$W_x = \bar{W} + \frac{\sum_{U \in \text{ähnliche Nutzer}} (U_x - \bar{U}) r(W, U)}{\sum_{U \in \text{ähnliche Nutzer}} |r(W, U)|}$$

- Höchstbewertete werden empfohlen

[83] © Robert Tolksdorf, Berlin

Probleme

- Anlaufprobleme:
 - Neue Nutzer finden keine Bewertungen vor
→ Erhalten keine/schlechte Empfehlungen
 - Neue Nutzer/Bewerter erhalten keine „Gegenleistung“ für ihre Bewertungen
→ Wie werden Nutzer zu Bewertungen gebracht?
 - Neue Produkte/Informationen sind nicht bewertet
→ Werden nicht empfohlen
- Abdeckungsproblem:
 - Bei vielen zu bewertenden Dingen ist Abdeckung gering
 - Wie erhält man viele Bewertungen?
 - Implizite Bewertungen durch Messungen
- Bisherige Systeme haben sich auf das Filtern von Usenet-News konzentriert

[84] © Robert Tolksdorf, Berlin

Literatur

- Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40,3 (1997), 77-87 <http://www.cs.umn.edu/Research/GroupLens/>
- Badrul M.Sarwar, Joseph A.Konstan, Al Borchers, Jon Herlocker, Brad Miller, and John Riedl. Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW), 1998.
- Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. 1992. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM 35(12):61-70.

[85] © Robert Tolksdorf, Berlin

Zusammenfassung

[86] © Robert Tolksdorf, Berlin

Zusammenfassung

- Information Retrieval
- Indexerstellung
- Multimedia Indexing
- Collaborative Filtering

[87] © Robert Tolksdorf, Berlin