

Vorlesung Netzbasierte Informationssysteme

Struktur und Erschließung des Web

Prof. Dr. Adrian Paschke

Arbeitsgruppe Corporate Semantic Web (AG-CSW)
Institut für Informatik, Freie Universität Berlin
paschke@inf.fu-berlin.de
<http://www.inf.fu-berlin.de/groups/ag-csw/>





Crawling

- Lynch, C. (1995). Networked Information Resource Discovery: An Overview of Current Issues (Invited paper). IEEE Journal on Selected Areas of Communications, 13(8):1505–1522:

"information discovery is a complex collection of activities that can range from simply *locating a well-specified digital object on the network* through lengthy iterative research activities which involve the *identification of a set of potentially relevant networked information resources*, the *organization and ranking resources in this candidate set*, and the *repeated expansion or restriction of this set* based on characteristics of the identified resources and exploration of specific resources."

- Das Web ist
 - Verteilt
 - Dezentral organisiert
 - Dynamisch
- Resource Discovery Problem:
Wo sind Informationsquellen von Interesse
- Lösungsidee für das Web:
 - Automatisches Navigieren über Seiten
 - Indexierung der gefundenen Seiten
 - *Crawler* (auch *Spider*, *Robot*, *Worm* etc.)

- Eines der ersten Systeme: *WebCrawler* [Pinkerton94]
- Zwei Funktionen
 - Indexierung des Web
 - Automatische Navigation nach Bedarf
- WebCrawler in 94:
 - 50000 Dokumente von 9000 Quellen indexiert
 - 6000 Anfragen täglich
 - Updates wöchentlich
- Suchmaschinen 11/04: [Searchenginewatch.com]
- Google geschätzt 9/05: 24 Milliarden Seiten

Search Engine	Reported Size	Page Depth
Google	8.1 billion	101K
MSN	5.0 billion	150K
Yahoo	4.2 billion (estimate)	500K
Ask Jeeves	2.5 billion	101K+

- Das Web als traversierbarer Graph von Seiten die über Links als Kanten verbunden sind
 - `<a>`, `<link>`, `<meta>`, ``, `<object>`, `<frameset>`
 - FTP-Server, Adressen in nicht-HTML Dokumenten
 - ...

```
<p class=up><a href="http://www.fu-berlin.de/">Freie
Universit&auml;t Berlin</a><br> <a href="http://www.math.fu-
berlin.de/">Fachbereich Mathematik und Informatik</a></p>
<h1>Institut f&uuml;r Informatik</h1> <p class=langchange><a
href="http://www.inf.fu-berlin.de/index_en.html">Homepage in
English</a>.</p>
```

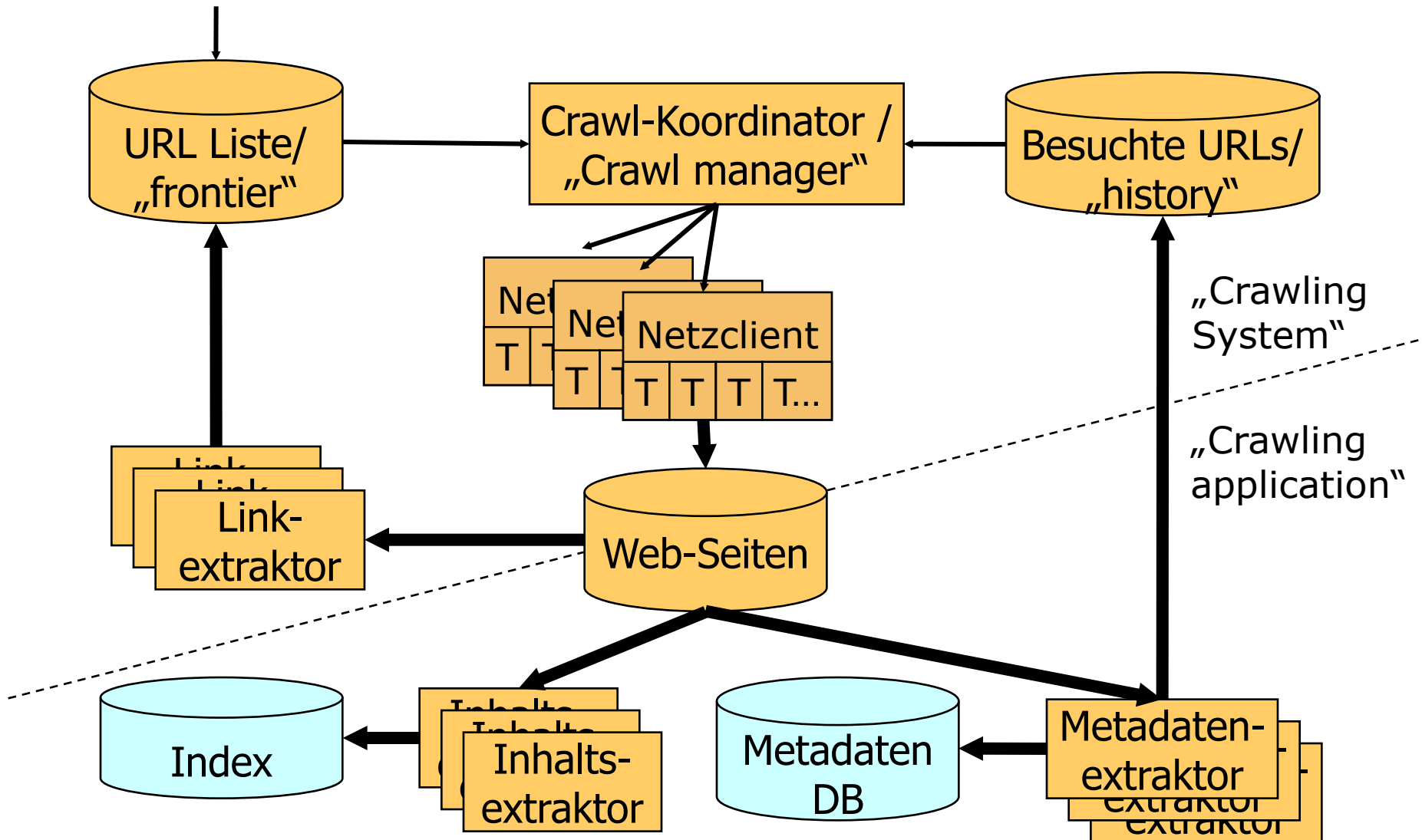
```
<frame SRC="content.html" NAME="content"
FRAMEBORDER="no">
<frame SRC="content.html" NAME="content"
FRAMEBORDER="no" NORESIZE
SCROLLING="auto">
<frame SRC="content.html" NAME="content"
FRAMEBORDER="no" NORESIZE SCROLLING="auto"
MARGINWIDTH="20" MARGINHEIGHT="20">
```

```
<table WIDTH=100% BORDER=0> <tr> <td> <img SRC="/pics/inf-
logo-klein.gif" ALT="Institutslogo" ALIGN=LEFT> </td> <td>
<small> <a HREF="http://www.fu-berlin.de/">Freie Universit&auml;t
Berlin</a>, <a HREF="http://www.math.fu-berlin.de/"> Department
of Mathematics and Computer Science </a> </small> <h1> Institute
of Computer Science</h1>
```

1. URL-Liste mit unbesuchten URLs initial füllen
2. Nehme URL aus Liste und teste
 - schon besucht?
 - passender Medientyp (html/ps/pdf/gif/...)?
 - andere Kriterien (Ort/...)?
3. hole Seite
4. extrahiere URLs und schreibe sie in URL-Liste
5. extrahiere und indexiere Seiteninhalt
6. extrahiere und speichere Metadaten
7. gehe nach 2

„Crawling loop“

Einfache Architektur

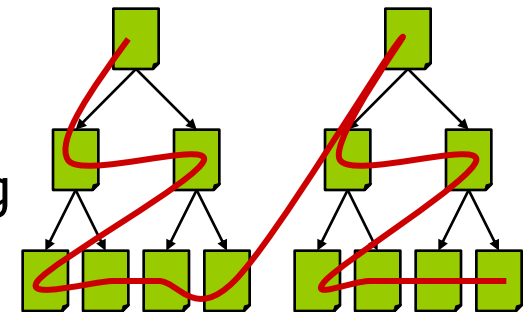
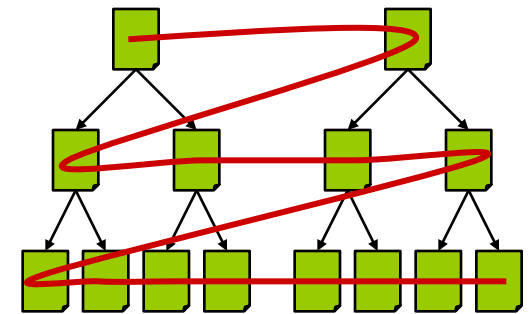
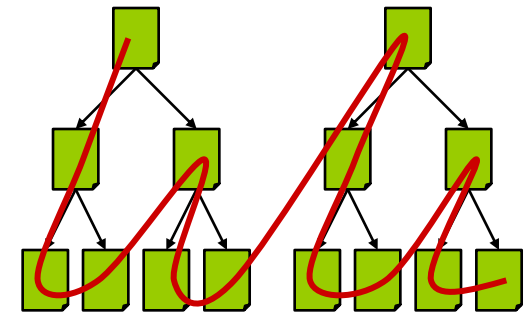


- URL-Liste / Frontier
 - Größe: Annahme: 7 Links pro Seite ->
 - Frontier wächst schnell
 - Frontier wird groß
 - Duplikate: Keine URLs doppelt
 - Serielle Suche teuer
 - Hash-Table mit URL als Schlüssel auch teuer

- Welche Links verfolgen?
 - `<a>`, `<link>`, `<meta>`, ``, `<object>`, `<frameset>` etc.?
- Im Web notierte URLs sind gar keine eindeutigen Schlüssel -> URL Normalisierung notwendig
 - `HTTP://www.UIOWA.edu` -> `http://www.uiowa.edu`.
 - `http://myspiders.biz.uiowa.edu/faq.html#` -> `http://myspiders.biz.uiowa.edu/faq.html`
 - `http://dollar.biz.uiowa.edu/%7Epant/` -> `http://dollar.biz.uiowa.edu/~pant/`
 - `http://dollar.biz.uiowa.edu` -> `http://dollar.biz.uiowa.edu/`
 - `http://www.foo.com/index.html` -> `http://www.foo.com/`
 - `http://dollar.biz.uiowa.edu/~pant/BizIntel/Seeds/./Seeds.dat` -> `http://dollar.biz.uiowa.edu/~pant/BizIntel/Seeds.dat`.
 - `http://www.foo.com:80/` -> `http://www.foo.com/`
- Viele weitere möglich, Heuristiken auch andersherum gültig

Designoptionen / Entnahme/Erweiterung der URL-Liste

- Durch Ordnung der Frontier wird die Crawl-Strategie bestimmt
 - Depth-First
"Enge" Suche in die Tiefe einzelner Sites
 - Breadth-First
"Breite" Suche über viele Sites, übliches Vorgehen
 - Breadth-First pro Site,
Nicht mehr beliebig, aber "breit" genug



- Best-first: Crawler versucht in „gute Richtung“ zu crawlen
 - Es gibt eine Vorgabe in Form einer Anfrage
 - Repräsentiert als Vektor von Termen
 - Crawler repräsentiert Seite als Vektor von Termen
 - Crawler ermittelt Ähnlichkeit der Vektoren
 - Alle auf der Seite gefundenen URLs erhalten Ähnlichkeit als Priorität
 - Frontier ist priorisierte Schlange
 - Crawl wird bei der nächsten „guten“ URL fortgesetzt
 - Weitere Prioritätsanhaltspunkte:
 - Entfernung von /
 - Angenommener Medientyp
 - Ankertext?

- Crawl-Koordinator
 - Schon gesehen?
 - Eigenschaften der URL
 - aus .de?
 - Verarbeitbarer Filetyp?
 - HTML
 - PDF, Postscript, Word
 - Excel?
 - MP3?
 - Serverzugriff zurückstellen?
 - Kurz vorher schon zugegriffen?
 - Schon zu viel von Server geholt?
 - Koordination mit weiteren Crawlern bei
 - Nebenläufigkeit
 - Verteilung

- Netzzugriffe
 - Wieviele Zugriffe parallel?
 - Welche Timeouts?
 - Umgang mit Fehlern
 - Verteilte Zugriffe?
- Erste Google-Versionen ca. 1997/8 (<http://google.stanford.edu>):
 - 3 Netzclients
 - je ca. 300 Verbindungen
 - mit 4 Clients ca. 100 Web Seiten/Minute crawlbar (144000/Tag, 6944 Tage für 1 Milliarde Seiten = 19 Jahre)
 - ca. 600Kb / Sekunde Netzlast

- Inhaltsextraktion
 - Welche Teile des Inhalts indexieren?
 - Überschriften
 - Nur Ankertexte
 - Titel
 - Gesamtdokument oder Teile davon?

Search Engine	Reported Size	Page Depth
Google	8.1 billion	101K
MSN	5.0 billion	150K
Yahoo	4.2 billion (estimate)	500K
Ask Jeeves	2.5 billion	101K+

- Metadaten ermitteln
 - Welche Metadaten speichern?
 - Titel
 - Besucht
 - <meta> Tag
 - Klassifikation?
 - Wann besucht
 - Quersumme?

- Framesets
- Unterschiedliche URLs für dieselbe Seite
Sitzungs-IDs, dynamisch erzeugte Pfade
- Errechnete Links ("Next year" auf einem Kalender)
- Dynamische Seiteninhalte (Javascript etc.)
- Fehlerhafte Seiten
- Transportprobleme durch Netz
- Transportprobleme durch Größe



Crawling aus Server-Sicht

- Crawler erzeugen Last beim Server
 - Verarbeitung der Anfragen
 - Auslieferung der Ergebnisse
- “Freundliche” Crawler versuchen das zu vermeiden
 - Keine fortlaufenden Anfragen zum Indexieren einer gesamten Site auf einen Schlag
 - Beachtung des Robot Exclusion Protokolls
 - Beachtung der <meta>-Tags zum Steuern von Robotern

Robots Exclusion Protokoll

- Definiert einen Mechanismus mit dem ein Server festlegt, ob er von einem Crawler besucht werden will
- Daten /robots.txt auf Server
- <http://www.inf.fu-berlin.de/robots.txt>:

```
# robots.txt for http://www.inf.fu-berlin.de/  
User-agent: *  
Disallow: /tec/net/  
Disallow: /tec/rechner/  
Disallow: /tec/software/packages/  
Disallow: /cgi-bin/  
User-agent: MOMspider/1.00  
Disallow: /cgi-bin/  
Disallow: /tec/software/packages/
```

- User-agent: bezeichnet den Roboter, für die die folgenden Regeln gelten sollen
 - Namen wie (s. <http://www.robotstxt.org/wc/active.html>)
 - Googlebot
 - Grapnel/0.01 Experiment
 - InfoSeek Robot 1.0
 - Platzhalter * für alle Roboter
- Bezeichnet jeweils einen Teil der Dokumentenraums, der nicht besucht werden soll
 - Eintrag
Disallow: /tec/net/
 - <http://www.inf.fu-berlin.de/tec/net> soll nicht besucht werden

- Alle Roboter ausschließen:
User-agent: *
Disallow: /
- Einzelne Roboter ausschließen:
User-agent: Roverdog
Disallow: /
- Einzelne Seiten schützen:
User-agent: googlebot
Disallow: cheese.htm
- Nur einen Crawler zulassen:
User-agent: webCrawler
Disallow:
User-agent: *
Disallow: /

<meta>-Element

- Das HTML <meta>-Tag kann ebenfalls zur Roboter-Steuerung genutzt werden

```
<html>
```

```
  <head>
```

```
    <meta name="robots"  
          content="noindex,nofollow">
```

```
    <title>...</title>
```

```
  </head>
```

- Verbreitung bei Robots unklar

<meta>-Element

- `index`: Diese Seite soll indexiert werden
- `noindex`: Diese Seite soll nicht indexiert werden
- `follow`: Die Links dieser Seite weiterverfolgen
- `nofollow`: Die Links dieser Seite nicht weiterverfolgen
- `all` = `index, follow`
- `none` = `noindex, nofollow`

- Keine Möglichkeit, Verhalten für bestimmte Crawler zu bestimmen
- Kein Zugriff auf `robots.txt` notwendig



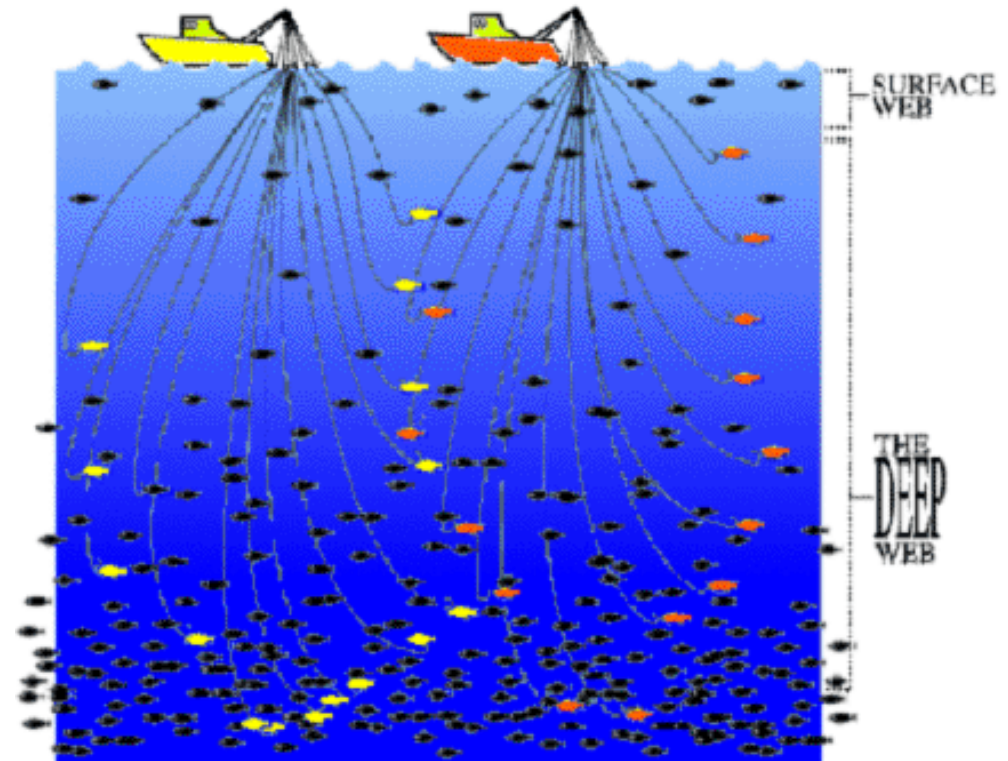
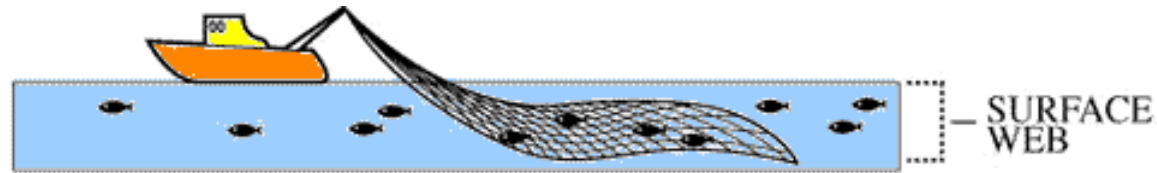
Das "Deep Web"

Michael K. Bergman. The Deep Web: Surfacing Hidden Value. The Journal of Electronic Publishing August, 2001. Volume 7, Issue 1 und <http://www.brightplanet.com/resources/details/deepweb.html>

He, B., Patel, M., Zhang, Z., and Chang, K. C. 2007. Accessing the deep web. *Commun. ACM* 50, 5 (May. 2007), 94-101. DOI=<http://doi.acm.org/10.1145/1230819.1241670>

"Deep Web"-Argumentation

- Traversierung des Web über Links führt nur zu einem Bruchteil der Informationen
- "Deep Web" wird von Datenbankinhalten gebildet
- Umfang 400-500 mal größer als "normales" Web
- 500 Mrd Dokumente vs. 1 Mrd Dokumente
- Zugriff aber nur durch Datenbankanfragen möglich



- 100 Sites analysiert
 - Schätzung der enthaltenen Datensätze oder Dokumente
 - Abfrage von Stichprobe von 10 Dokumenten zu Größenabschätzung durch Mittelwertbildung
 - Indexierung und Klassifizierung des Suchformulars
- Größenschätzung
 - Nachfrage bei Betreibern
 - Aussagen auf Site
 - Aussagen über Site in anderen Berichte
 - Zahlen bei Suchantworten, z.B. Treffer für "NOT sfgjsljffjd"
 - Ausschluss aus Untersuchung
- Schätzung: Durchschnittlich 74,4 MB pro Site

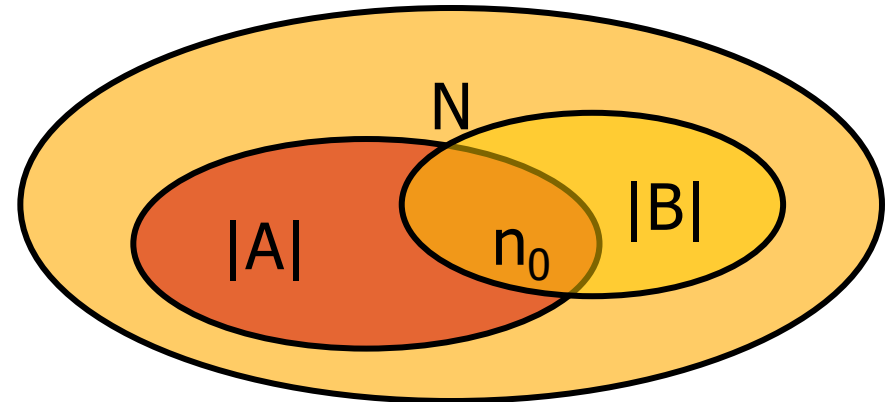
Größenschätzung Sites des Deep Web

Name	Type	Web Size (GBs)
National Climatic Data Center (NOAA)	Public	366,000
NASA EOSDIS	Public	219,600
National Oceanographic (combined with Geophysical) Data Center (NOAA)	Public/Fee	32,940
Alexa	Public (partial)	15,860
...
Subtotal Public and Mixed Sources		673,035
DBT Online	Fee	30,500
Lexis-Nexis	Fee	12,200
Dialog	Fee	10,980
Genealogy - ancestry.com	Fee	6,500
ProQuest Direct (incl. Digital Vault)	Fee	3,172
...
Subtotal Fee-Based Sources		75.469
Total		748,504

- Manuell und teilweise automatisch unterstützt:
 - 53220 URL-Hinweise aus anderen Sites
 - 45732 ohne Duplikate
 - 43348 noch zugängige
 - 17579 anscheinend suchbare
 - 13,6% davon nicht suchbar

Overlap analysis: Gesucht N - Größe des Deep Web

- n_A, n_B Abdeckung durch je eine Suchmaschine / ein Verzeichnis
- n_0 Überlappung
- $|A|, |B|$: Größe von A, B
- $p(A)$: Wahrscheinlichkeit, Seite von A gefunden wird
- $p(A \cap B) = p(A) * p(B)$
- $|A| = N * p(A), |B| = N * p(B), |A \cap B| = N * p(A \cap B)$
- $N = |A| * |B| / |A \cap B|$
- Da Verzeichnisse nicht zufällig: Untere Grenze



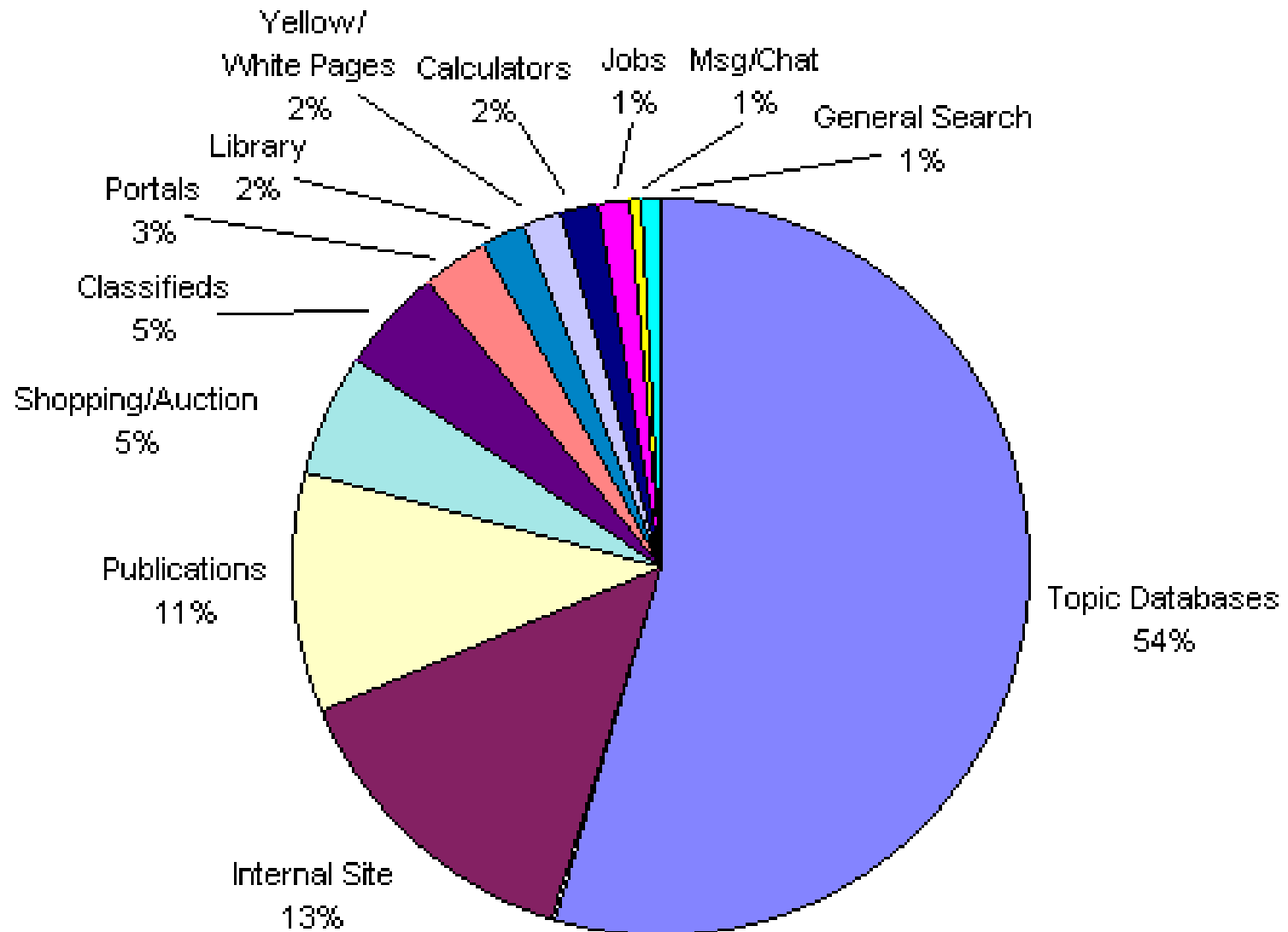
Schätzung Anzahl der Sites

DB A								Tot Est Deep Web
DB A	A no dups	DB B	B no dups	A+ B	Uniq.	DB Fract.	DB Size	Sites
Lycos	5,081	Internets	3,449	256	4,825	0.074	5,081	68,455
Lycos	5,081	Infomine	2,969	156	4,925	0.053	5,081	96,702
Internets	3,449	Infomine	2,969	234	3,215	0.079	3,449	43,761

- Schätzung: Ca. 100000 Deep Web Sites

- Inhaltsüberprüfung durch Anfragen aus 20 Gebieten
- Typanalyse durch Handauswertung von 700 Sites

Agriculture	2.7%	Law/Politics	3.9%
Arts	6.6%	Lifestyles	4.0%
Business	5.9%	News, Media	12.2%
Computing/Web	6.9%	People, Companies	4.9%
Education	4.3%	Recreation, Sports	3.5%
Employment	4.1%	References	4.5%
Engineering	3.1%	Science, Math	4.0%
Government	3.9%	Travel	3.4%
Health	5.5%	Shopping	3.2%
Humanities	13.5%	Law/Politics	3.9%



- Deep Web: 7500 Terabytes, Web: 19 Terabytes
- Deep Web: 550 Mrd Docs, Web: 1 Mrd Docs
- Mehr Traffic auf Deep Web Sites (50%)
- Mehr Wachstum im Deep Web
- Deep Web Sites mehr inhaltliche Tiefe und weniger inhaltliche Breite
- 95% des Deep Web frei zugänglich

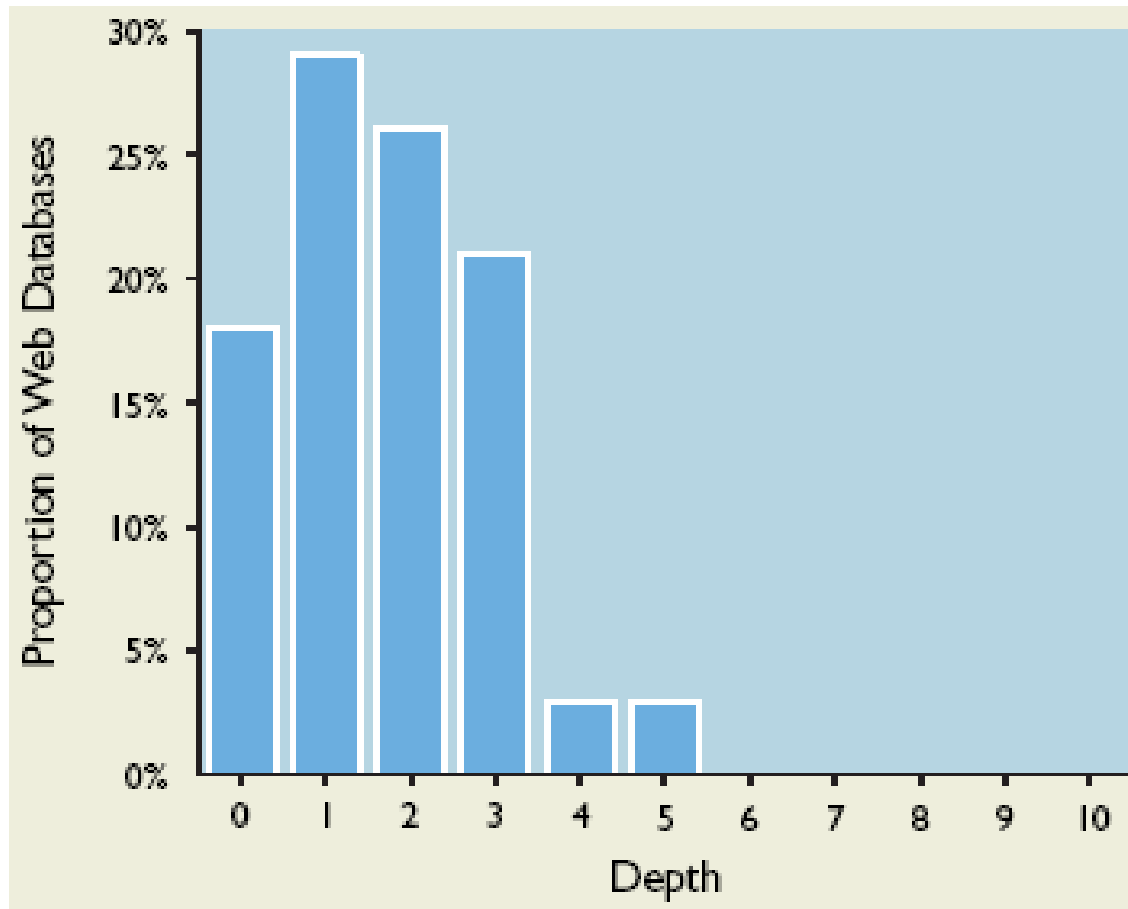
- Probleme:
 - Intention der Deep Web Studie
 - Erschließung?

- He/Patel/Szang/Chang: Überlappungsanalyse geht von Unabhängigkeit zwischen Indizes der Suchmaschinen aus
 - Das ist aber nicht gegeben
 - -> Deep Web Größe ist unterschätzt
- Vorgehen
 - 1000000 IP-Nummer auswählen
 - Auf Web-Server testen
 - Suchfelder ermitteln
 - Def. Deep Web Server: Server der über ein Suchformular Datenbankinhalte herausgibt

- #Suchformulare->#Datenbanken->#Deep Web Server
- Duplikate ausschließen
 - Suchfelder für „site search“, „login“ etc. herausnehmen
 - Formulare mit gleichem Ziel herausnehmen
 - Durch zufällige Anfragen gleiche Datenbanken ermitteln

[alle folgenden Abbildungen aus HePatelZhangChang2007]

- Wo befinden sich die Suchformulare des Deep Web?
 - 100000 IP Nummern in Tiefe untersucht



- Aus 1000000 IP Nummern 2256 Web Server ermittelt
- Davon 126 Deep Web Sites
- Mit 406 Suchformularen zu 190 Datenbanken

- Internet (IPv4) Adressraum = 2230124544 Nummern
- Hochrechnung aus Tiefenuntersuchung
 - 307000 Deep Web Sites
 - 450000 Datenbanken
 - 1258000 Suchformulare

- Vgl: 43000-96000 Deep Web Sites in Brightplanet Studie

- Abdeckung durch Suchmaschinen
 - Aus Datenbanken Ergebnisobjekte ermitteln
 - In Suchmaschinen anfragen
- Abdeckung durch Suchmaschinen ca. 1/3:
 - Google, Yahoo: 32%
 - MSN: 11%
 - Große Überlappung
- Abdeckung durch Deep Web Verzeichnisse: Gering

Verzeichnis	#	Abdeckung
completeplanet.com	70000	15,6%
lii.org	14000	3,1%
turbo10.com	2300	0,5%
invisible-web.net	1000	0,2%

- Brian Pinkerton. Finding What People Want: Experiences with the WebCrawler. Second International World-Wide Web Conference: Mosaic and the Web, Chicago, IL, October 17--20 1994.
<http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/pinkerton/WebCrawler.html>
- G. Pant, P. Srinivasan, and F. Menczer. Crawling the Web. In M. Levene and A. Poullovassilis, editors, Web Dynamics: Adapting to Change in Content, Size, Topology and Use. Springer-Verlag, 2004.
<http://citeseer.ist.psu.edu/579280.html>
- www.searchenginewatch.com
- The Web Robots Pages. www.robotstxt.org



Information Retrieval

V. Gudivada, V. Raghavan, W. Grosky and R. Kasanagottu.
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 Modelling and Text Retrieval.
1999. Siam.

- 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 Anfragen einfach zu beantworten sind
- Bei Suchmaschinen üblich:
 - Volltextindex gesammelter Seiten erstellen
 - Anfragen an den Volltextindex weiterleiten
 - Ergebnisse ordnen
 - Verweise auf Ursprungsdokumente an Nutzer ausliefern

- 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

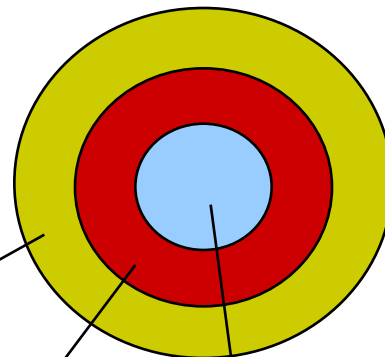
- Zwei Arten von Termen in IR
- *Objektive* Terme:
 - Außerhalb des eigentlichen Inhalts
 - Beispiele: Autorennamen, URL etc.
 - Einfach und klar zuzuordnen
- *Nichtobjektive* Terme / *Inhaltsterme*:
 - Beschreiben Informationen des Dokumenteninhalts
 - Schwierig zuzuordnen
 - Hauptaufgabe des Indexing

- *Indexing exhaustivity (Erfassungsgrad)*:
 - Grad zu dem Inhalt durch Indexing erfasst wird
 - Hohe Ausschöpfung: Viele Terme zugeordnet
 - Geringe Ausschöpfung: Weniger Terme zugeordnet
- *Term specificity (Detailgrad)*:
 - „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
- Fahrzeug Auto
- PKW BMW

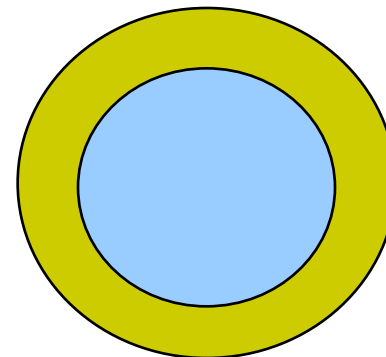
- *Recall (Nachweisquote)*:
Wie gut findet das System relevante Dokument wieder?

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

0,x



1,0



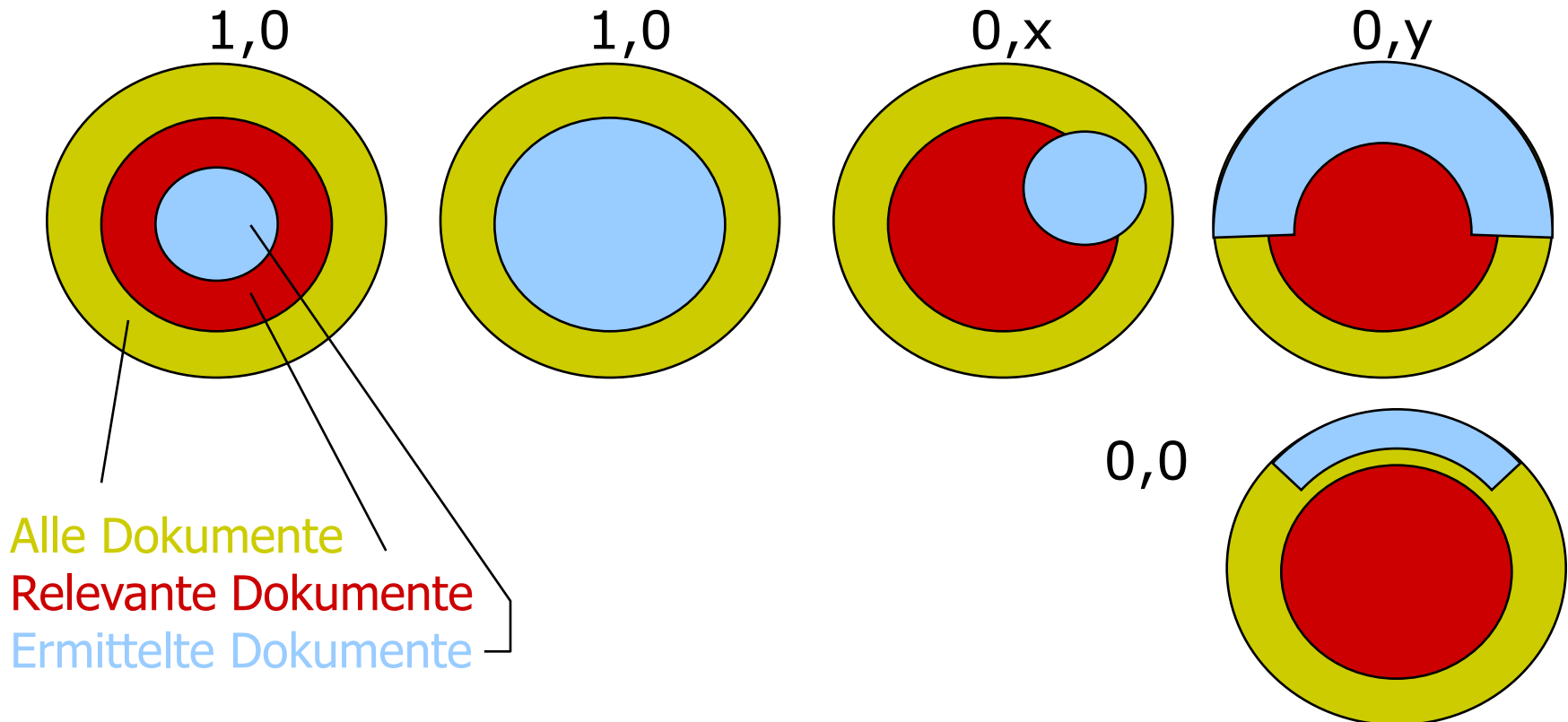
Alle Dokumente

Relevante Dokumente

Ermittelte relevante Dokumente

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

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



F-Measure

- Ein Wert um Güte auszudrücken?
 - Harmonisches Mittel zwischen Recall und Precision

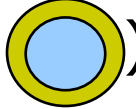
$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

- p=60%, r=75% -> $F_1=0,66$
- p=75%, r=60% -> $F_1=0,66$
- p=60%, r=60% -> $F_1=0,6$

- Allgemein:

$$F_a = \frac{(1+a) * \text{precision} * \text{recall}}{a * \text{precision} + \text{recall}}$$

- $F_{0,5}$: precision doppelt so wichtig
- F_2 : recall doppelt so wichtig

- Ziel Recall und Precision bei 1 () ($F_1=1$)
- 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“
- Messung basiert immer auf fester Dokumentenmenge mit händischer Relevanzbewertung
 - TREC Testsätze, <http://trec.nist.gov/>

- 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

- 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

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

- $T = (\text{„heute“}, \text{„ist“}, \text{„dienstag“}, \text{„vorlesung“}, \text{„nicht“})$
- 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	0	1	1
d_2	1	0	1	0
d_3	1	1	1	0

- 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)

Disjunktive Normalform DNF

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

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

- d_4 : „heute am diensttag ist die vorlesung nicht“
ist genauso bewertet wie
 d_5 : „heute ist die vorlesung nicht“
- Intuitiv aber ähnlicher der Anfrage

- Ähnlichkeitsmaß aus Anzahl der Übereinstimmungen

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

- $\text{sim}(d_4, q) = 3 > \text{sim}(d_5, q) = 2$

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

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

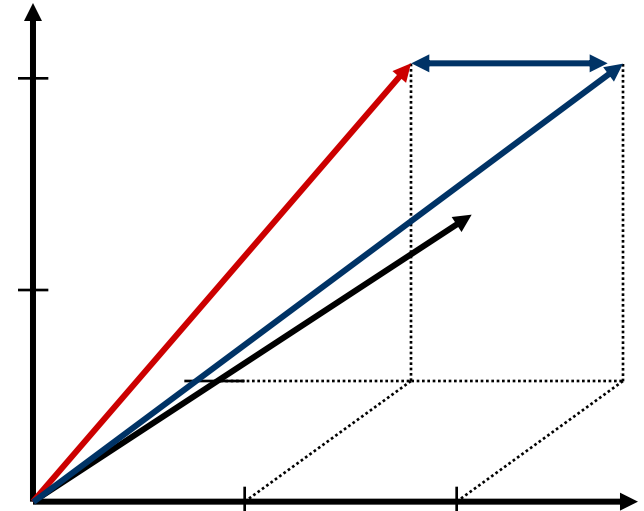
- Dokumente und Anfragen repräsentiert durch Vector in einem n-dimensionalen Raum
- Dimensionen durch Terme gegeben
- Gewichtet und normalisiert
- Relevanz durch Ähnlichkeitsmaß von Anfrage und Dokument gegeben und geordnet
- Sehr einfaches Modell
- Ausdrucksmächtigkeit boolescher Ausdrücke nicht vorhanden

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 der Punkte (Euklidische Distanz)

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

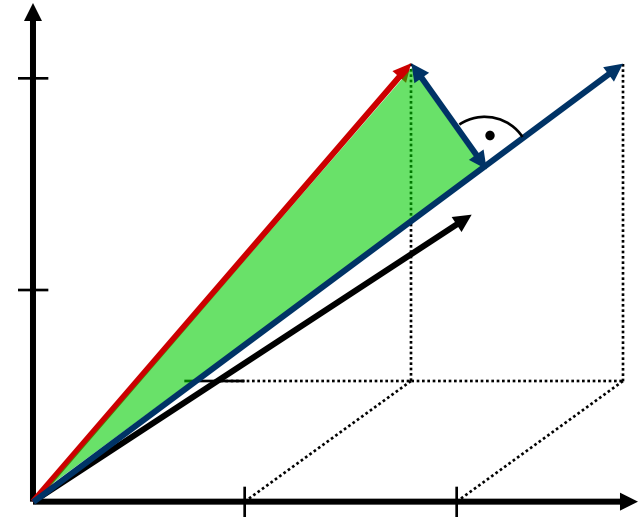
- Problem: Je mehr Terme, je größer der Abstand
- Dokumente haben mehr Terme als Anfragen
- Abstand immer groß
 - Zusätzliche Terme im Dokument „ziehen“ Punkt weiter weg
 - Häufige Terme „ziehen“ den Punkt weiter weg



- Skalarprodukt als Ähnlichkeitsmaß:

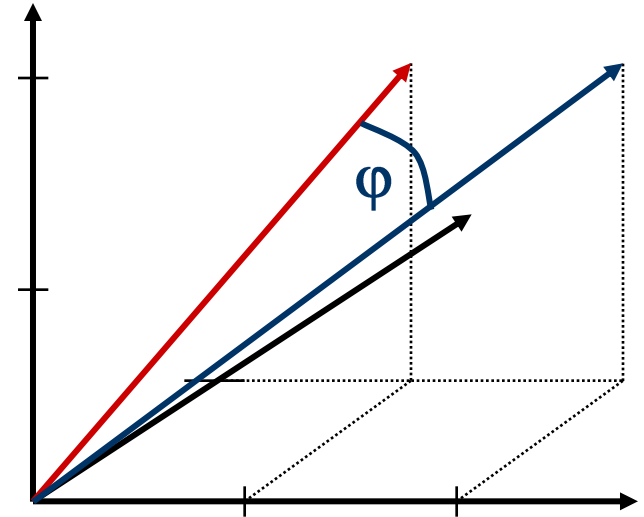
$$\text{sim}(d_j, q) =$$

$$w_{1,j} * q_1 + \dots + w_{n,j} * q_n$$



- Problem: Je mehr Terme, je größer der Vektor, je größer das Skalarprodukt
 - Zusätzliche Terme im Dokument „ziehen“ Vektor weiter weg

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



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

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

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 (Dokumentenfrequenz)*:
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

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

- Nutzer verfeinern Anfrage nach und nach
- Relevanz-Feedback: Nutzer bewerten Ergebnisgüte
- 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

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

- 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

- 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