8. Informationssuche

Stefan Deßloch

Informations-
systeme

Überblick

- Grundprobleme der Informationssuche
- Informationssuche
  - in Datenbanksystemen (strukturiert)
  - in Information-Retrieval-Systemen (unstrukturiert)
    - Indexierung
      - Probleme, Techniken
      - Ergebnis der Anfrageauswertung
- Systemgüte
  - Gütemaße (Precision, Recall)
- Einordnung von Suchhilfen
- Suchmaschinen im WWW
  - Anforderungen und Arbeitsweise
  - Suchstrategien
- Metasuchmaschinen
  - Anfrageverteilung
  - Ergebniszusammenstellung
- WWW-Anfragesprachen (WebSQL)

Grundprobleme der Informationssuche

- Nützlichkeit der Information in kritischer Weise abhängig von ihrer
  - Aktualität (currency)
  - Vollständigkeit (completeness)
- Wachstum der Informationsmenge verschärft das Problem
  - Ergänzung neuer Information (zeitgerechter Änderungsdienst)
  - Entfernung veralteter Information

  ➥ wünschenswertes Ziel oft unmöglich zu erreichen (Aktualität ↔ Vollständigkeit)
- Informationsflut - Beispiel: wissenschaftliche Publikationen
  - bis 1800: Verdopplung aller wiss. Publikationen alle 50 Jahre
  - 1800 - 1998: wissenschaftliche Zeitschriften: 10^2 ➔ 10^5
    aktuell: > 70 000 wiss. Zeitschriften, 15000 Artikel/Woche
    Institute for Scientific Information (Philadelphia) wertet 5000 der führenden wiss., techn. und med. Zeitschriften aus (Bibliometrik)
  - Momentan keine obere Grenze sichtbar

  ➥ „exponentielles" Wachstum: Bsp. Informatik

Grundprobleme (2)

- Warum?
  - Enorme Verstärkung der Forschungsanstrengungen

    ➥ in den letzten 150 Jahren: Verzehnfachung der Anzahl der Wissenschaftler alle 50 Jahre
  - Akademische Maßstäbe/Zwänge

    ➥ Publish or perish ?
  - Information als „Ware"

    ➥ Marktmechanismus, Selbstdarstellung, . . .
- Hilft das Web bei der Problemlösung?

  \+ + Orts- und Plattformunabhängigkeit, Suche zu jeder Zeit

  \- - Desorganisation, Mißinformation, Interpretationsprobleme
- Wachstum des Web
  - Faktor 10 pro Jahr
  - Wie lange kann es so weiter gehen?
  - Anzahl der Benutzer: Faktor 20? heute: ~ 50 Mio → 1 Mrd?
  - Wieviel mehr Daten (Text) als heute stellt der durchschnittliche Benutzer online ins Web: Faktor 20?

  ➥ 800 TBytes ASCII-Daten im Web?

Informationssuche – strukturierte Daten

- DB-Beispiel

ANGESTELLTER

PNR	NAME	TAETIGKEIT	GEHALT	ALTER
496	Peinl	Pförtner	2100	63
497	Kinzinger	Kopist	2800	25
498	Meyweg	Kalligraph	4500	56
...				

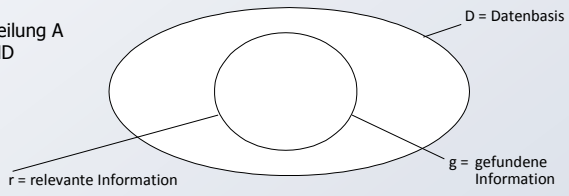
- Suche in DBS bei strukturierten Daten
  - Zeichen-/Wertvergleich:
    (TAETIGKEIT = 'PFOERTNER') AND (ALTER > 60)
  - exakte Fragebeantwortung: alle Sätze mit spezifizierter Eigenschaft werden gefunden (und nur solche)
  - Suche nach syntaktischer Ähnlichkeit:
    (TAETIGKEIT LIKE '%PF%RTNER')
    - LIKE-Prädikat entspricht der Maskensuche

Anfragen auf strukturierten Daten

- Genau festgelegter Bereich der Anfrage (Tabellen der FROM-Klausel)
- Verknüpfung verschiedener Tabellen ausschließlich über Werte
- Auswahl von Sätzen mit Hilfe von Prädikaten (WHERE-Klausel)

➥ Ergebnis liefert genau alle das Suchprädikat erfüllenden Sätze

```
SELECT *
FROM Angestellter P, Abteilung A
WHERE P.Anr = A.Anr AND
  P.Taetigkeit = 'Kopist'
```



Unstrukturierte Daten

- Daten in IRS
  - Objekte sind nur durch Dokumenttyp und Wert (D_i/W_k) beschrieben
  - Die Bedeutung der Objekte ist „inhärent"
  - Sie sind durch „lange" Werte repräsentiert und in speziellen Containern (Dateien) gespeichert
- Einsatzmöglichkeiten
  - Bibliotheken, Literaturrecherche, z. B. im Internet (WWW)
  - Informationsdienste, Informations-Broker: Chemie, Recht, Patente, . . .
- Beispiel - Call for Papers (Dokumenttyp)
- Was wird gesucht?

  Select *
  From ???
  Where ???
- Wie wird inhaltsbasiert (nach W_k) gesucht?
  - sequentiell: Volltextsuche
  - Indexnutzung
- Beschreibung der Dokumente durch Deskriptoren
  - geeignete Deskriptoren für CfP?

Die rasante Entwicklung des Web hat gerade im Datenbankbereich einerseits eine Vielzahl neuer Anwendungsfelder eröffnet, andererseits aber auch neue Herausforderungen geschaffen. Vor diesem Hintergrund soll der Workshop eine Plattform für Forscher und Praktiker bilden, um Ergebnisse der Datenbanktechnologie und -theorie mit Anforderungen und Erfahrungen aus Internet-Anwendungen abzustimmen. Im einzelnen sollen folgende Themenstellungen sowie damit verbundene Bereiche behandelt werden: . . .

Suche auf unstrukturierten Daten

- Art der Suche
  - Anfragen relativ unscharf (Ergebnis nicht "eindeutig bestimmt")
  - Suchbegriffe des Benutzers müssen „irgendwie" mit den Deskriptoren der Dokumente korrespondieren
  - Mehrdeutigkeit: bei Wörtern in Dokumenten und Anfragen (Synonym-, Homonymproblem usw.)
  - Ähnlichkeitssuche (nearest neighbor, best match, pattern matching usw.)
  - Ergebnisbewertung: Relevanzproblem (Precision, Recall)
- Verbesserung
  - Synonymsuche, unscharfe Suche usw. sollen unterstützt werden
  - Einsatz eines Thesaurus oder einer Ontologie* (Begriffssystematik): Festlegung von Begriffen und ihren Beziehungen zueinander

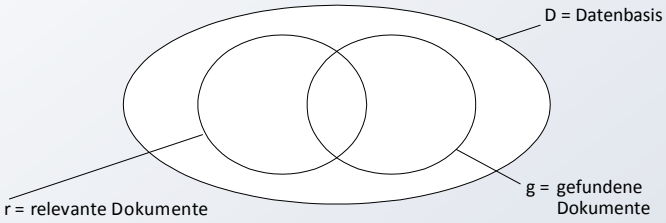
  ➥ „Verstehen natürlicher Sprache", Erkennen von Mehrdeutigkeiten sind „harte Probleme"

  Bsp.: "Time flies like an arrow – fruit flies like a banana" (Groucho Marx)

(*) Ontologie ist ursprünglich eine philosophische Disziplin, neuerdings ein Modebegriff der Informatik. Nach Meyers Enzyklopädischem Lexikon ist Ontologie die „Lehre von dem Wesen und den Eigenschaften des Seienden". In der Informatik ist sie die „formale Spezifikation eines bestimmten Gegenstandsbereichs in Form eines Begriffssystems".

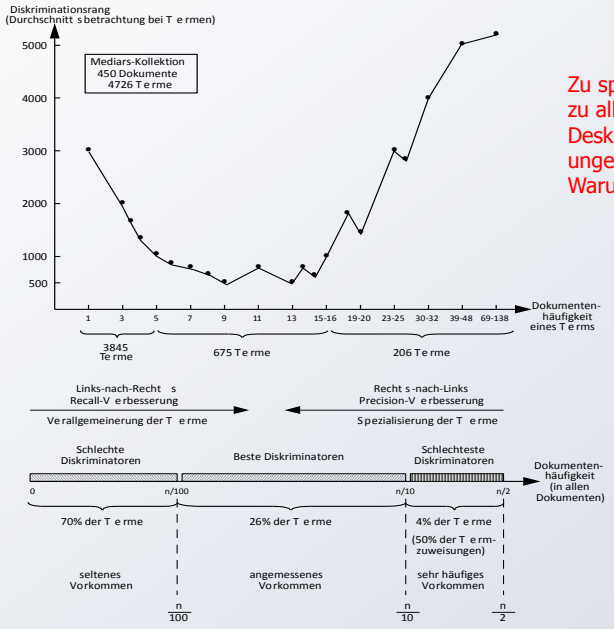
Anfragen in Information Retrieval Systemen

- Auswertung von Anfragen in IRS (und analog im Web)
  - Recall R bezieht sich auf die relevanten Informationen
  - Precision P bezieht sich auf die gefundenen Informationen



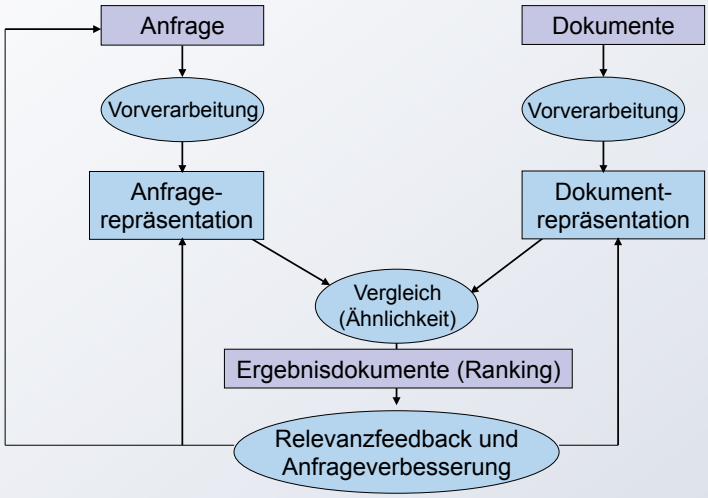
$$R = \frac{|r \cap g|}{|r|} \qquad P = \frac{|r \cap g|}{|g|}$$

Auswahl von Dokument-Deskriptoren



Zu spezielle und zu allgemeine Deskriptoren sind ungeeignet! Warum?

Information Retrieval - Prozeß



Text-Retrieval

- Wie wird in den Textquellen (DBs) gesucht?
- Dokument-Repräsentation
  - Dokument d = (d_1, ..., d_i, ..., d_n)
    - Wortschatz V = (t_1, ... t_k)
    - d_i: Gewicht des i-ten Terms t_i für d
    - die meisten d_i sind 0 für ein gegebenes Dokument
  - entsteht durch manuelle oder automatische Vorverarbeitung
- Automatische Vorverarbeitung
  - Entferne Stoppwörter: von, der, und, durch, ...
  - Wortstammbildung (erfordert "Sprachverstehen"):
    - Bäume → Baum
    - Suchbaumdurchläufe → Suche(n), Baum, durch, Lauf(en)
    - aber?
  - Formel zur automatischen Berechnung der d_i
    - term frequency (tf_i) = (#t_i in d)/$max_{t \in V}$(#t in d)
    - inverse document frequency (idf_i) = log(#Dok/#Dok mit t_i)
    - $d_i = tf_i \times idf_i$

Textsuche

- Anfrage-Repräsentation
  - Terme der Anfrage durch Benutzer vorgegeben, ggf. vom System erweitert (Synonyme, Hypernyme, ...)
  - $q = (q_1, ..., q_i, ..., q_n)$
    - q_i: Gewicht des i-ten Terms in q
  - Berechne q_i: typischerweise Nutzung von tf
- Einsatz von Ähnlichkeitsfunktionen (**Vektorraum-Modell**)
  - Dokumente lassen sich als Term-Vektoren auffassen: $d_i \in [0,1]^{|T|}$ mit d_{ij} = Gewicht des Terms t_j in d_i
  - Anfrage wird abgebildet auf Anfrage-Vektor: $q \in [0,1]^{|T|}$
  - Ähnlichkeitsmetrik zur Relevanzberechnung:

$$sim(d_i, q) := \frac{\sum_{j=1}^{t} (d_{ij} \times q_j)}{\sqrt{\sum_{j=1}^{t} d_{ij}^2} \times \sqrt{\sum_{j=1}^{t} q_j^2}}$$

  - Ranking nach absteigender Reihenfolge
  - Relevanz-Feedback durch den Benutzer

Indexierung

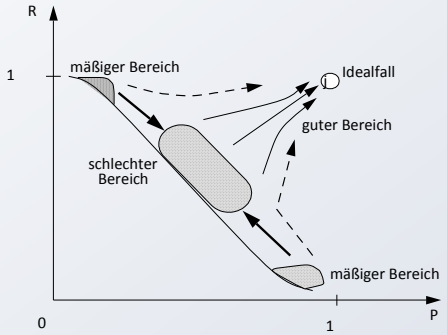
- Indexierung bei unstrukturierten Daten in IRS
  - Invertierung der gesamten Texte der Dokumente vom Typ D_i durch Bibliothekar oder durch spezielle Programme
  - Oft kontrollierter Wortschatz: Wertevorrat für Deskriptoren (Schlüsselwörter) ist begrenzt

  ➥ typischerweise: Anlegen eines Index pro Typ D_i als B*-Baum
- Indexeinträge
  - Deskriptor/Schlüsselwort t_i als Suchschlüssel
  - Daten im Eintrag für t_i (Blätter im B*-Baum)
    - Liste aller Dokumente mit zugeh. tf, für die $tf_i > 0$
  - "invertierte Liste"
- Suche mit Index
  - Ermittlung der Trefferlisten für jedes Schlüsselwort in der Anfrage
  - Kombination der Listen zur Relevanzberechnung für die Dokumente basierend auf der Ähnlichkeitsfunktion

Systemgüte

- Messung der Systemgüte

  empirische Ermittlung des R/P-Verhältnisses über Mengen von IRS-Anfragen



➥ Recall und Precision sind wesentliche IRS-Qualitätsmaße!

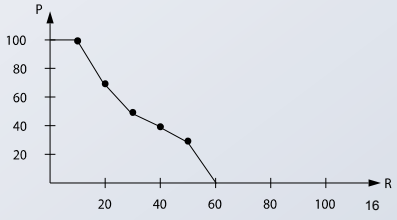
Bei Suchmaschinen im Web zusätzliche Kriterien: Überdeckungsgrad, ...

Retrieval-Bewertung - Recall und Precision

- Variation von R und P
  - R und P unterstellen, daß alle Dokumente in der Antwortmenge (g) überprüft werden
  - Typischerweise sind die Dokumente nach ihrem Relevanzgrad sortiert (Ranking); der Benutzer inspiziert die Liste „von oben nach unten"
  - R und P variieren mit dem Listendurchlauf
  - genauere Bewertung durch 11 Standard-Recall-Ebenen (P bei 0%, 10%, ..., 100% R)
- Beispiel für eine Anfrage
  - Menge der relevanten Dokumente in D für die Anfrage Q sei r_q mit $|r_q| = 10$
    $r_q = \{d_4, d_5, d_8, d_{25}, d_{49}, d_{64}, d_{70}, d_{75}, d_{101}, d_{199}\}$
- Retrieval-Algorithmus liefere die Menge g_q mit folgendem Ranking:

1. d199 ·	6. d8 ·	11. d47
2. d95	7. d929	12. d62
3. d49 ·	8. d747	13. d471
4. d12	9. d111	14. d123
5. d9	10. d25 ·	15. d4 ·

- Precision P bei 11 Standard-Recall-Ebenen



Retrieval-Bewertung (2)

- Durchschnittsbildung über mehrere Anfragen

  R_i = Recall-Ebene i, i = 0, ..., 10 (R_5 bezeichnet Recall-Ebene 50%)

  $P(R_i)$ = Precision bei Recall-Ebene i

  n_q = Anzahl der Anfragen

  $$P(R_i) = \sum_{j=1}^{n_q} \frac{P_j(R_i)}{n_q}$$

- Interpolation bei Recall-Ebenen

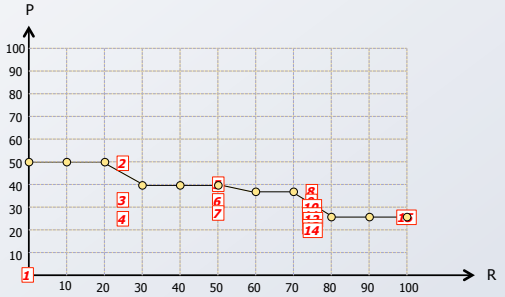
  $$P(R_i) = \max_{R_i \le R \le R_{i+1}} P(R)$$

Interpolationsbeispiel

- r'_q = $\{d_4, d_9, d_{95}, d_{747}\}$
- Ranking der gefundenen Menge g_q

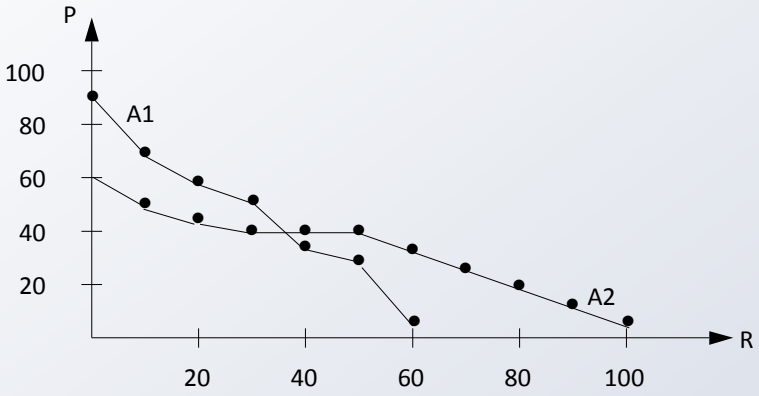
1.	d_{199}	6.	d_8	11.	d_{47}
2.	d_{95} ·	7.	d_{929}	12.	d_{62}
3.	d_{49}	8.	d_{747} ·	13.	d_{471}
4.	d_{12}	9.	d_{111}	14.	d_{123}
5.	d_9 ·	10.	d_{25}	15.	d_4 ·

- d_{95} hat Recall-Level 25% und Precision 50%



Vergleich

- Vergleich zweier (fiktiver) Retrieval-Algorithmen



➥ Welche Retrieval-Performance besitzen A1 und A2?

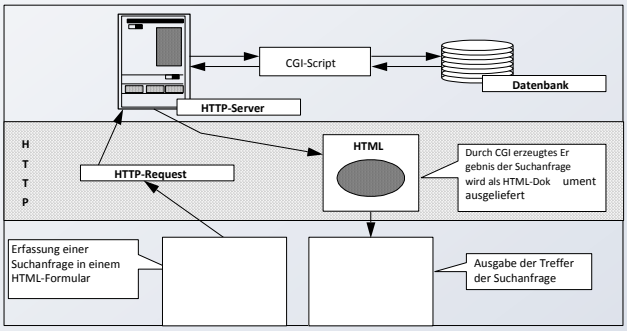
Weitere R/P-Variationen

Zusammenfassung zu einem „goldenen" Wert

- Average Precision at Seen Relevant Documents
  - Durchschnittsbildung mit den ersten n relevanten Dokumenten
  - Beispiel: P-Werte seien 1, 0.66, 0.5, 0.4, 0.3
    P_{avg} = (1 + 0.66 + 0.5 + 0.4 + 0.3) / 5 = 0.57
- R-Precision
  - n_R sei $|r_q|$;
    R-Precision wird an der Stelle n_R der Ergebnisliste bestimmt
  - Beispiel:
    n_R = 10; mit 4 relevanten Dokumenten bis zu dieser Position:
    R-Precision = 0.4

Suche im WWW

- WWW unterscheidet sich von DBS:
  - enormes und ständig wachsendes Datenvolumen
  - wesentlich breiteres Benutzerspektrum mit sehr heterogenen Rechnerkenntnissen
  - große Diversität der verwalteten Dokumente, die wenig Annahmen über eine einheitliche Struktur zuläßt
- Prinzipieller Ablauf der Suche



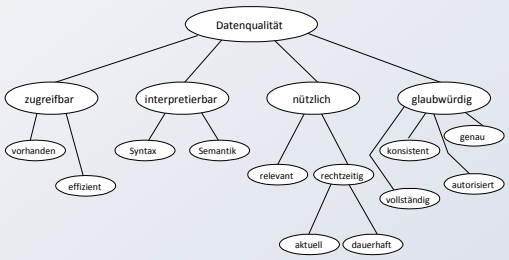
Einordnung von Suchhilfen

	generisch	domänenspezifisch
referenzbasiert	HTML / HTTP-Navigation	Hotlists, Themenkat aloge
wertebasiert	Suchmaschinen im engen Sinne	Informations-Broker

- Möglichkeiten der Suche im Web
  - ➥ Entwicklung der Suche im Web weist überraschende Analogien zur Entwicklung der DBS auf
- Referenzbasierte Suche (wie Netzmodell)
  - Problem: „getting lost in hyper space"
  - Verbesserung/Einschränkung: benutzerdefinierte Hotlists oder Bookmarks
- Wertebasierte Suche (wie Relationenmodell)
  - Problem: Interpretation bei Dokumentdiversität
  - Wesentliche Verbesserung/Einschränkung: Aufbereitung der Daten für bestimmte Kundenkreise

Einordnung von Suchhilfen (2)

- Beurteilung der Informationsqualität
  - differenzierter als durch Precision und Recall in IRS
  - Schema wurde am MIT entwickelt und berücksichtigt bei der Antwort
    - Zugreifbarkeit (Antwortzeitverhalten, Abdeckungsgrad)
    - Interpretierbarkeit (gleichförmige Ergebnisaufbereitung, semantische übereinstimmung)
    - Nützlichkeit (~ Precision)
    - Glaubwürdigkeit (Konsistenz, Vollständigkeit)



Einordnung von Suchhilfen (3)

- Kataloge
  - handselektierte Links, die nach Themen und innerhalb der Themen hierarchisch geordnet sind
  - Aktualität ist problematisch (da geringer Automatisierungsgrad)
  - Verbesserung: Erstellung und Wartung durch Informations-Broker
  - Beispiel: Database Systems & Logic Programming http://www.informatik.uni-trier.de/~ley/db/index.html

  ➥ Erstellung und Wartung erfordert viel manuelle Nacharbeit!!
- Suchmaschinen (Search Engines)
  - automatisierte Suche und Darstellung der „Such-Information"
  - Indexierung der Ergebnisse (Titel, Inhalt, URL)
  - Einsatz von „Spidern", „Crawlern" usw.

Informationsqualität der Suchhilfen*

Qualität / Suchhilfen	Zugreifbarkeit	Interpretier-barkeit	Nützlichkeit	Glaubwürdig-keit
HTML/HTTP-Navigation	schlecht	schlecht	schlecht	gut
Hotlists, Themenkataloge	gut	gut	mittel	mittel
Suchmaschinen im engeren Sinne	gut	schlecht	mittel	schlecht
Informations-Broker	gut(?)	gut(?)	gut(?)	gut(?)

(?) Kombination der automatischen Indexierung (Extraktion der Information aus Originaldokumenten) mit zusätzlichem Wissen (manuelle Nacharbeit)

(*) Jeusfeld, A., Jarke, M.: Suchhilfen für das World Wide Web: Funktionsweisen und Metadatenstrukturen, Wirtschaftsinformatik 39, 1997, S. 491-499.

Suchmaschinen im WWW - Anforderungen

- Suche im „gesamten" WWW
  - Suchbereich: prinzipiell alle Dokumente auf allen Servern
- Inhaltliche und strukturelle Suche
  - Welche Seiten gehören zu welchem Thema?
  - Welche Dokumente liegen auf Server mit URLx?
  - Welche Dokumente lassen sich von URLx aus erreichen?
- Korrektheit der Ergebnisse
  - URL-Listen sollen keine veralteten Links enthalten (referentielle Integrität!)
  - Ergebnisdokumente sollen thematisch „relevant" sein
  - ➥ Relevanzbegriff im Kontext von HTML und WWW
  - Problem: aus unstrukturierten oder semistrukturierten Daten ist die eigentliche Information zu bestimmen
  - ➥ Jeder Suchdienst hat einen anderen Ansatz zur Relevanzbestimmung
- Wenig Eingriffe in die dezentrale WWW-Organisation
  - autonome Organisation von Servern und dezentraler Aufbau des Web sind die Grundpfeiler für Einfachheit der Informationsbereitstellung
  - Strukturlosigkeit der Dokumente erschwert das Auffinden der relevanten Informationen

  ➥ stark verbesserte Suchmöglichkeiten durch verbindliche Beschreibung der HTML-Dokumente mit Metadaten

Probleme

- Indexierung durch Suchmaschinen im Web
  - Vollständigkeit/Erreichbarkeit aller Dokumente? (Abdeckungsgrad der relevanten Datenquellen)
  - Aktualität der Indexierung
  - Wie lange würde die Erstellung eines Index dauern? (bei 10^9 - 10^{12} Dokumenten und einer Zugriffszeit von 1 - 10^{-2} sec/ Dok.)
  - Wann werden neue Dokumente gefunden?
  - Wie zeitgerecht sind Änderungen in alten Dokumenten im Index repräsentiert?
  - . . .
- Verschärfung des Problems
  - Es gibt keinen kontrollierten Wortschatz (Thesaurus), wie in manchen gut gepflegten IRS üblich
  - Es werden Dokumente in vielen Sprachen gefunden (Multi-Kulti-Index?)
  - Autoren von Dokumenten verwenden Begriffe nicht nur unkontrolliert, sie streben manchmal sogar eine gezielte Überlistung der Verfahren zur automatischen Indexierung an

  ➥ gezielte und häufige Verwendung bestimmter Begriffe

Verbesserung der Web-Suche

- Schwierigkeiten bei der Suche
  - „Harvard" kommt in > Mio. Webseiten vor, Begriff auf Homepage von Harvard ist für Suchmaschinen „uninteressant"
  - Auf der Homepage von „IBM" kommt der Begriff Computer nicht vor

  ➥ Bei manchen Suchmaschinen werden solche Probleme manuell gelöst:

  Die „Harvard"-Seite kommt an die Spitze der Liste für „Harvard".

  Aber: Handverlesene Seiten bei unbegrenzter Anzahl von Fragen?
- Verbesserung
  - Verbot der Desinformation?
  - geplante Organisation der im Web bereitgestellten Information
  - Bereitstellung von Metadaten zu jedem Dokument

  ➥ verbesserte Suchergebnisse, geringerer Zeitaufwand!

PageRank

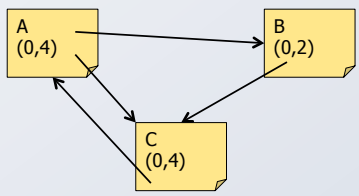
- Verfahren entwickelt von Larry Page (einer der Google-Gründer)
  - Grundidee: Ausnutzen der durch Hyperlinks gegebenen Konnektivität zur Relevanzbewertung von Webseiten
    - Modell: ein Benutzer surft im Web und besucht mit einer geg. Wahrscheinlichkeit eine zufällig gewählte Seite oder verfolgt zufällig einen Link auf der aktuellen Seite.
      - kehrt nicht über "Back"-Links auf eine frühere Seite zurück
    - PageRank(a) gibt die Wahrscheinlichkeit an, mit der sich der Surfer auf der Seite a befindet. Dieser Wert kann unabhängig von Suchanfragen ermittelt werden und wird dann bei der Berechnung der Relevanz für die Suchergebnisse zusätzlich berücksichtigt.
  - PageRank ist
    - proportional zur Anzahl der eingehenden Hyperlinks und zur Relevanz der Seiten, die den Link enthalten
    - umgekehrt proportional zur Anzahl der Links, die in den referenzierenden Seiten enthalten sind
- Alternative Interpretation
  - Wenn eine Seite einen Link zu einer anderen Seite enthält, drückt sie damit eine "Empfehlung" für sie aus.

PageRank (2)

- Formel zur (iterativen) Berechnung
  - C(a) = Anzahl der ausgehenden Links in a
  - in p_1 bis p_n wird auf a verwiesen
  - q = Wahrscheinlichkeit für Sprung auf "zufällige" Seite

$$PR(a) = q / N + (1 - q) \sum_{i=1}^{n} PR(p_i) / C(p_i)$$

- Beispiel (hier mit q = 0)



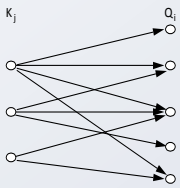
- Effiziente Berechnung
  - formal: Matrixmodell mit Übergangswahrscheinlichkeiten
  - massive Parallelisierung
    - MapReduce Programmierparadigma

Der HITS-Algorithmus

- Nutzung des Wissens, das in den Hyperlinks steckt
  - Annahme: Hyperlink ist eine Empfehlung der Seite, auf die er zeigt (Ausnahmen: strukturelle Links (zurück zur Homepage) oder Reklame)
  - Programm Clever von IBM: Unterscheidung von Webseiten als Quellen und als Knotenpunkte
  - Quellen sind Seiten über ein spezielles Thema, auf die zahlreiche andere Webseiten verweisen (und ihnen damit eine gewisse Glaubwürdigkeit zusprechen); beispielsweise die Homepage von Amnesty International zum Thema Menschenrechte
  - Knotenpunkte sind Seiten, die eine große Anzahl von Quellen auflisten, wie z. B. die Rubrik „Meine Lieblingslinks" einer persönlichen Homepage oder professionelle Portale
- Quellen und Knotenpunkte als lose und nicht zentral geplante Ordnungsstruktur des Web
  - Eine renommierte Quelle Q_i ist eine Seite, auf die von einer großen Anzahl von Knotenpunkten verwiesen wird
  - Ein Knotenpunkt K_j ist gut, wenn er auf eine große Anzahl von beachtenswerten Quellen verweist
  - ➥ Es wird versucht, die Klassifikationsarbeit der Web-Benutzer auszunutzen

Der HITS-Algorithmus (2)

- Suchalgorithmus mit Bewertung von Quellen und Knotenpunkten
  1. Finden einer Liste von Kandidatenseiten mit herkömmlichen Verfahren
  2. Anfängliche Grobbewertung von Q_i und K_j
  3. Neubewertung der Q_i: eine Quelle gilt als „gut", wenn viele „gute" Knotenpunkte auf sie verweisen
  4. Neubewertung der K_j: die neue Note von K_j ist umso besser, je besser die bisherigen Noten der Q_i sind, auf die er verweist



  - schnelle Konvergenz: bei 3000 Primärseiten 5 Iterationen, bis sich die Bewertungen kaum noch ändern
  - Endergebnisse sind im wesentlichen unabhängig von den anfänglichen Schätzwerten. Man könnte sogar alle Anfangswerte auf 1 setzen

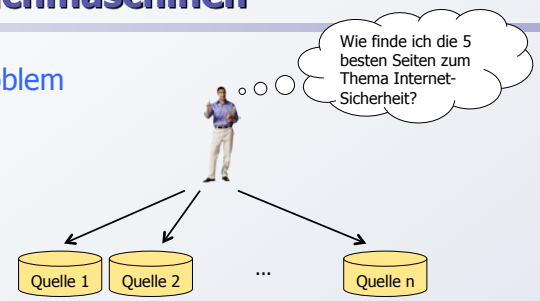
Leistungszahlen von Google (2008)

- Wieviele Seiten im Web? *
  - Google-Crawl(ohne Duplikate)
    - 1998: 26 x 10^6 Seiten
    - 2000: > 10^9 Seiten
    - 2008: > 10^{12} Seiten
    - Wachstum heute: > 10^9 Seiten pro Tag
  - Nicht alle indexiert
    - Kalenderseiten, automatisch generierte Inhalte, ...
- In 2004
  - > 200 Millionen Anfragen pro Tag
  - > 20 TB indexierte Seiteninhalte
  - > 20 000 PCs im Suchmaschinencluster

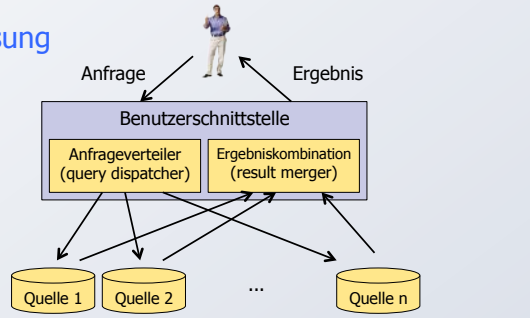
(*) http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

Metasuchmaschinen

- Das Problem

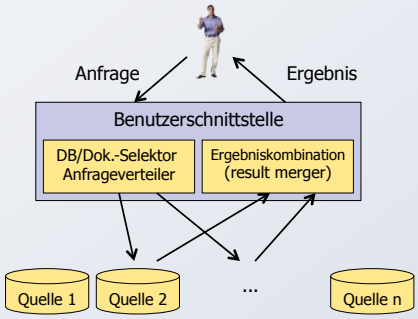


- Eine Lösung



Metasuchmaschinen (2)

- Einige Beobachtungen
  - Die meisten Textquellen liefern keinen nützlichen Beitrag für eine gegebene Anfrage
  - Eine Anfrage an eine nutzlose Textquelle
    - erzeugt unnötigen Netzverkehr
    - verschwendet lokale Ressourcen für die Anfrageauswertung
    - erhöht die Kosten für die Ergebniszusammenstellung
  - Das Aufsuchen von zu vielen Dokumenten von einer (nützlichen) Quelle ist ineffizient
- Eine effiziente Metasuchmaschine



Informations-systeme
Grundprobleme
Informations-suche
Systemgüte
Suchhilfen
Suchmaschinen
Meta-Suchmaschinen
WWW-Anfragesprachen

Metasuchmaschinen (3)

- Einsatzbeispiel
  - Anfrage: Internet-Sicherheit

  - Suchergebnisse (URLs): t1, t2, t2 ... p1, p2, p3 ..
  - Ergebnisliste: p1, t2, p2, ...
    - Duplikateliminierung, Relevanzbestimmung, Ranking
- DB-Selektion
  - Auswahl potentiell nützlicher DBs für eine gegebene Anfrage
  - Potentiell nützliche DBs enthalten potentiell nützliche Dokumente. Solche Dokumente
    - besitzen eine globale Ähnlichkeit über einem vorgegebenen Schwellwert und
    - sind unter den m Dokumenten mit der größten globalen Ähnlichkeit

    ➥ Es ist Wissen über jede DB vorab erforderlich, um die DB-Selektion vornehmen zu können.

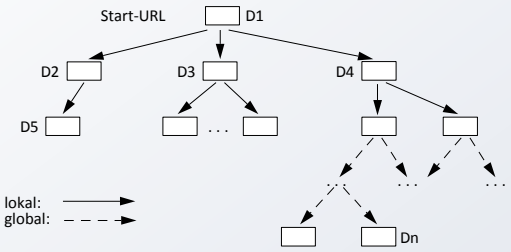
WWW-Anfragesprachen

- DB-Anfragesprachen erlauben deklarativen und navigierenden Zugriff auf eine DB
  - mengenorientierte Deklaration durch Prädikate
    - ➥ Was wird gesucht?
  - Navigationsmöglichkeiten durch Information aus DB-Schema
    - ➥ Wie kann etwas aufgefunden werden?
  - Ziel: deklarative Anfragen um WWW-spezifische Eigenheiten erweitern
- WebSQL*
  - Beschreibung des Web durch ein relationales DB-Schema mit zwei Relationen:
    1. Document (url, title, text, type, length, modif) zur Beschreibung der Dokumente selbst,
    2. Anchor (base, label, ref) für die Hyperlinks zwischen den einzelnen Dokumenten.
  - SQL-ähnliche Sprache
    - mit inhaltlicher Spezifikation von Dokumente durch Attribute/Attributwerte
    - mit Pfadnotation zur Spezifikation, nach welchen Dokumenten gesucht werden soll
- Pfadnotationen (Web-spezifischer Aspekt)
  - erlauben die Suchtiefe einzuschränken
  - nur lokale URLs: → *
  - Einbezug globaler URLs: ⇒*
  - Einsatz regulärer Ausdrücke: → → (⇒ *)
    (alle Pfade, die durch zwei lokale Links und anschließend durch beliebig viele globale Links beschrieben sind)

  ➥ Versuch der Kostenbewertung bei solchen Anfragen

(*) A.O. Mendelzon, G.A. Mihaila, T. Milo: Querying the World Wide Web, in: Proc. 4th Int. Conf. on Parallel and Distributed Information Systems (PDIS), 1996.

WWW-Anfragesprachen (2)



- Anfragebeispiel:
  - Es sollen alle Dokumente gefunden werden, die auf dem WWW-Server der Uni KL liegen und von deren Einstiegsseite erreicht werden können. Im Text der Dokumente soll „Datenbanksysteme" vorkommen. Als Ergebnis sollen URL und Titel der betreffenden Webseiten ausgegeben werden:

```
SELECT      d.url, d.title
FROM        Document d
SUCH THAT 'http://www.uni-kl.de' -> * d
WHERE       d.text CONTAINS 'Datenbanksysteme'
```

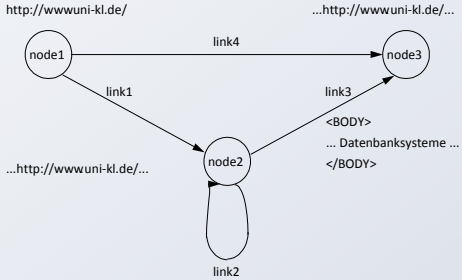
  - SELECT- und FROM-Klausel wie in SQL
  - SUCH THAT schränkt FROM-Klausel ein
  - WHERE-Klausel: Spezifikation von Suchbedingungen, die sich auf die reine Schlagwortsuche in einem HTML-Dokument beschränken

WWW-Anfragesprachen (3)

- Spezifikation der Suche
  - Angabe einer Start-URL: Einstieg für die Navigation
  - Fehlen eines Startdokumentes: Rückgriff auf Suchdienste durch WebSQL
    - Indexserver liefern Listen von URLs, die (Teile der) Suchbedingung (WHERE-Klausel) erfüllen
    - Kombination der Suchbreite von Indexservern mit der zusätzlichen Funktionalität von WebSQL

WWW-Anfragesprachen (4)

- World Wide Web Query System mit der Sprache W_3QL
  - Nutzung von semistrukturierten Dateien (HTML-, Latex-Format) und Verwendung von deren Strukturelementen als Suchkriterium
  - Web wird als gerichteter Graph interpretiert (mit Dokumenten als Knoten und HyperLinks als Kanten)
  - Anfrage beschreibt Teilgraph mit seinen Eigenschaften, der durch die Suche gefunden werden soll
  - Nutzung von Indexservern, wenn keine Start-URL angegeben wird
  - Ergebnisse werden als Views (wie in relationalen DBS) bezeichnet, da sie eine Sicht auf das WWW darstellen
  - Mit UNIX-Kommandos und -Programmen können Listen aus den zurückgelieferten Ergebnissen erzeugt werden
- Beispiel für einen gesuchten Graphen in W_3QS



WWW-Anfragesprachen (5)

- Anfragesprache W_3QL^*
  - In einer Anfrage werden Startknoten und beteiligte Knoten mit den Links genau spezifiziert
  - Suchbedingungen für Dokumente, z. B. solche, in deren BODY das Wort Datenbanksysteme vorkommt
  - Ausgabe beliebiger Textmarkierungen und Eigenschaften (z.B. URL und Titel)
- Anfragebeispiel

```
SELECT      node3.url, node3.title
FROM        node1, link1, (node2, link2), link3, node3; node1, link4, node3
WHERE       node1 in {http: / / www.uni-kl.de};
 node2 in {http: / / www.uni-kl.de};
 node3 in {http: / / www.uni-kl.de};
 node3: PERLCOND
 `node3.body = ~ / ^ Datenbanksysteme/ `
```

- Spezifikationen von
  - Ausgabe in SELECT
  - Teilgraph in FROM: Klammern geben Schleifen an, Teilgraphen werden durch Semicolon getrennt
  - den zu erfüllenden Eigenschaften der einzelnen Knoten in WHERE-Klausel
  - zusätzliche Bedingungen in WHERE: Beschreibung mit Perl-Syntax

(*) D. Konopnicki, O. Shmueli: W_3QS: A Query System for th World Wide Web, in: Proc. 21st Int. Conf. on Very Large Data Bases (VLDB), pp. 54-64, 1995.

Zusammenfassung

- Grundprobleme der Informationssuche
  - Aktualität
  - Vollständigkeit
  - Wachstum der Informationsmenge
- Informationssuche - unstrukturierte Dokumente
  - unscharfe Suche
  - Suchqualität: precision, recall
  - Einsatz von IR-Techniken
    - Ähnlichkeitssuche - Synonymsuche, Einsatz von Wörterbüchern
    - Indexierungstechniken - Vorverarbeitung (z.B. Wortstammbildung),
    - Auswahl und Gewichtung von Suchbegriffen/Termen
    - Einsatz von Ähnlichkeitsfunktionen
    - Bildung einer Rangfolge der Ergebnisse
- Suche im Web
  - Einordnung von Suchhilfen
  - Suchmaschinen vs. Meta-Suchmaschinen
  - Anfragesprachen für das Web