Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

Information Engineering 1: Information Retrieval

Kapitel 6

Web Search

M. Braschler

(basierend u.a. auf Material von J. Savoy)

"Themenkarte" Websearch

- Fallstudie CHerCHer
- Das Web in Zahlen
- Suche im Web
- Entwicklung und Anatomie von Suchmaschinen
- Spidering
- Indexierung Webseiten / Anfrageverarbeitung
- Evaluation
- Spreading Activation (SA)
- Indegree
- PageRank
- HITS (Kleinberg Model)
- Link-Analyse
- PageRank & Indegree

Lernziel Kapitel

- Web Search ist der Use-Case in IR, den alle kennen (Google...)
- Sie sollen verstehen,
  - .. inwiefern die bereits behandelte Theorie in diesem Umfeld zum Zuge kommt
  - .. was die Besonderheiten des Web Search sind
  - .. inwieweit die speziellen Erkenntnisse auf andere Umfelder übertragen werden können

Fallstudie: CHerCHer

- Das fiktive Schweizer Suchmaschinenunternehmen CHerCHer betreibt eine Suchmaschine, die vollständig aus Werbeeinnahmen finanziert werden soll. Die CHerCHer-Technologie basiert auf preiswerter PC-Hardware. Ein CHerCHer-PC ist fähig, Indizes mit 100 Millionen Webseiten zu verarbeiten. Durch Clustering der PC's wird es möglich, enorme Datenmengen verarbeiten zu können. Neben den Such-PC's ist eine Anzahl von PC's vorhanden, die sich ausschliesslich um gepufferte Antworten von gebräuchlichen Anfragen kümmern.
- Der CHerCHer-Spider crawlt das gesamte Web ab. Die folgenden Folien zeigen das voraussichtliche Budget und Unterhaltskosten für das kommende Jahr.

Ursprüngliches Beispiel von D. Hawking, CSIRO zitiert nach J.Savoy.

Technische Daten von CHerCHer

- Indexgrösse: 30 Milliarden Webseiten
- Durchschnittliche Seitengrösse: 15 Kbytes
- Ertrag pro Anfrage: 0.25 Cents
- Durchschnittliche Anzahl Anfragen pro Tag: 20 Millionen
- Spitzenwert Anzahl Anfragen pro Tag: 100 Millionen
- Durchschnittliche Dauer einer vollständigen Anfrage: 0.5 sek.
- Dauer um eine gepufferte Anfrage zurückzuliefern: 0.001 sek. (Cluster)
- Anteil von Anfragen, die gepuffert werden: 35%
- Kosten für einen Standard-PC: €300 (jährliche Leasingkosten)
- Netzwerkkosten: €20 per Terabyte
- Spidering-Budget: €1 Million
- Fixkosten (z.B. Gehälter, Miete, Ferrari für CEO): €2.5 Millionen

Probleme / Fragen

- Frage 1: Wie viel Netzwerkkosten fallen bei einem vollständigem Spidern an?
  - **$30*10^9$ Webseiten · 15 Kb = $450*10^9$ Kb → 450 TB = €9000**
- Frage 2: In was für einem Intervall kann gespidert werden um innerhalb des Budgets zu bleiben?
  - **1 Million / 9000 = 111/Jahr oder ca. alle 3 Tage**
- Frage 3: Wie viele PC's werden gebraucht, um einen Cluster aufzubauen, der die gesamte Datenmenge verarbeitet werden kann?
  - **1 PC = 100 Mill Webseiten, Total 30 Mrd Webseiten → 300 PC/Cluster**
- Frage 4: Wie viele PC's werden gebraucht um mit ungepufferten Anfragen in Spitzenzeiten umgehen zu können?
  - **100M · 65% · 0.5 sec = 32.5 Msec → 377 Tage für 1 Cluster**
    Wir brauchen 377 Clusters = 377 * 300 PC's = 113,100 PC's
  - **Google Schätzung schon im Jahr '06: bis >450,000 PCs**

Probleme / Fragen

- Frage 5: Wie viele PC's werden für die gepufferten Anfragen in Spitzenzeiten gebraucht?
  - **100M · 35% · 0.001 sec = 35 Ksec → 9.7 h für 1 Cluster**
    wir brauchen 1 Cluster = 300 PCs
- Frage 6: Wie hoch sind die Hardwarekosten?
  - **113,400 PC · € 300 = € 34M**
- Frage 7: Wie hoch ist der voraussichtliche Erlös?
  - **€0.0025 · $20 \cdot 10^6$ = €50K/day → 360days = €18M**
- Frage 8: Was kostet der Strom?
  - **113,400 Server à 8 kWh pro Tag schlucken 27 GWh/Monat: kostet irgendwo in der Grössenordnung von 4-5 Millionen CHF (Stand 2021)**
- Frage 9: Wie hoch ist der voraussichtliche Verlust für dieses Jahr?
  - **(34M + 1M + 4M + 2.5M) - 18M = €-23.5M**

Probleme / Fragen

- Frage 10: Was wird der Gewinn/Verlust für CHerCHer sein, wenn folgenden Vorschläge umgesetzt werden?
- Verwende grössere und teurere (€3000 pro Jahr) PC's für gepufferte Anfragen, um den Anteil der gepufferten Anfragen auf 50% zu steigern. (Aber: realistisch?)
  - **100M · 50% · 0.001 sek. = 50 Ksek. → 13.89 h für 1 Cluster**
    wir brauchen 1 Cluster = 300 PCs → €0.9 M
    100M · 50% · 0.5 sek. = 25 Msek. → 290 Tage für 1 Cluster
    wir brauchen 290 · 300 PCs = 87,000 PCs → €26.1 M
    Anfrageverarbeitungskosten: €26.1+0.9= € 27M (-7M)
- Einführung einer Anfrageoptimierung, die die Dauer einer Anfrageverarbeitung auf 0.33 sek. kürzt.
  - **100M · 65% · 0.33 sek. = 21.45Msek. → 249 Tage**
    wir brauchen 249 · 300 PCs = 74,700 PCs
    Kosten → €22.41 M (anstatt €33.93M → -€11.5M)

Probleme

- Frage 11: Was könnte die Motivationen für CHerCHer sein um:
  - die Qualität der Suchresultate zu verbessern?
  - häufigere Aktualisierungen des Indexes vorzunehmen?

Das Web in Zahlen..

- Anzahl der Internet-User
  - 61 mio. in 1996
  - 147 mio. in 1998
  - 604 mio. in 2002
  - 1,802 mio. in 2010
  - 2,405 mio. in 2012
  - 4,930 mio. in 2020/Q3

- Weltweite Verteilung der Internet-User (Penetration in %)

  728M Europa (87%)
  2556M Asien (60%)
  333M Nordamerika (90%)
  29M Ozeanien (68%)
  468M Lateinamerika (72%)
  632M Afrika (47%) (war nur 9% im Jahr 2010!)
  185M Naher/mittlerer Osten (71%)

(www.internetworldstats.com)

Das Web in Zahlen…

- Volumen der verfügbaren Daten

  5% “surface web”

  95% “deep web” (dynamisch)

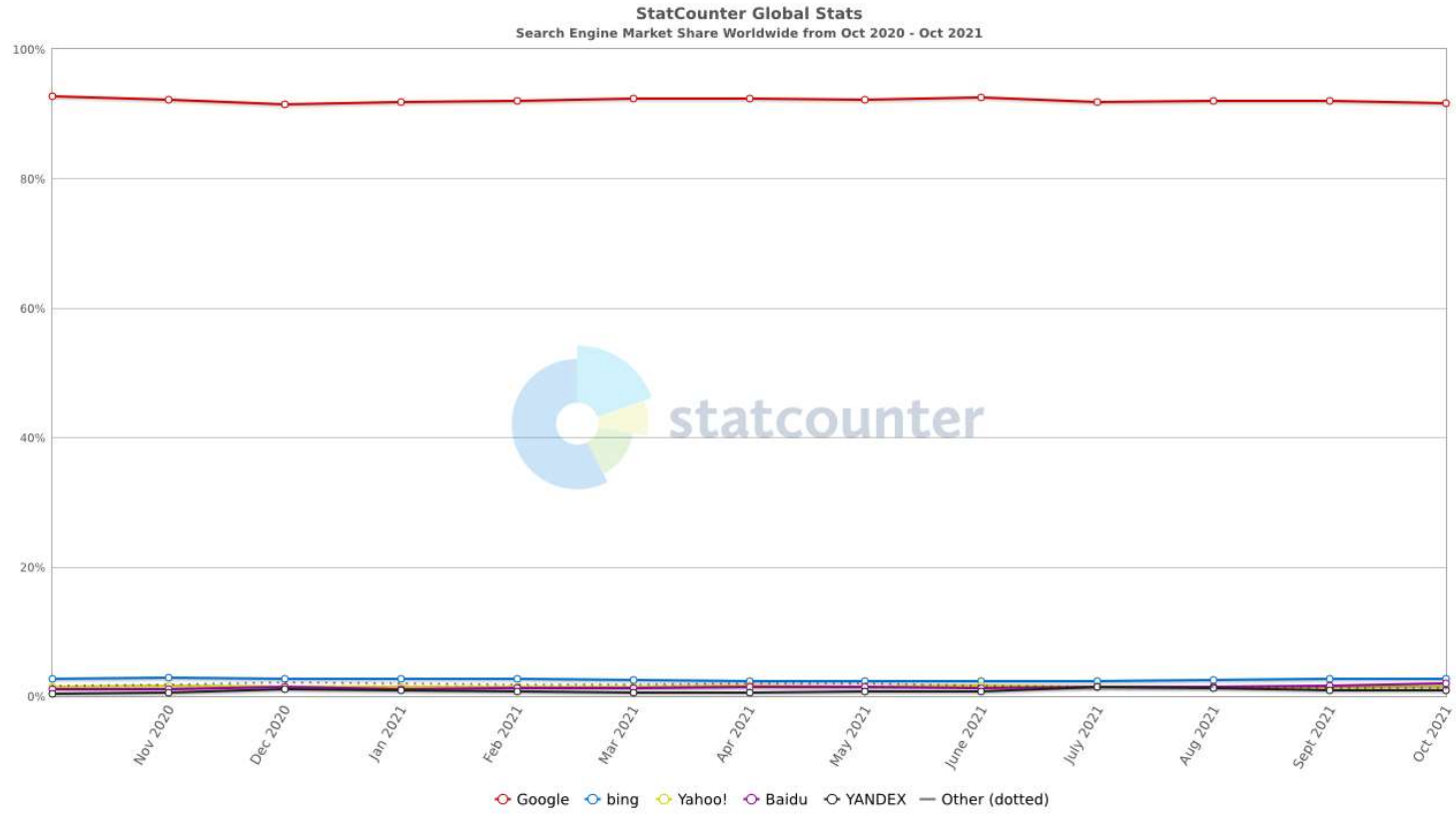
  unbekannt: “dark web”

- Schätzung von 2015: Google-Index des “surface web” ~14.5 Milliarden Seiten
- Schätzungen variieren dramatisch, oft zwischen 5 und 60 Milliarden Seiten

Marktanteil

Globaler Marktanteil Google Websuche: ~92%

Aber: Marktführer in RU: Yandex, in CN: Baidu, in JP: Yahoo



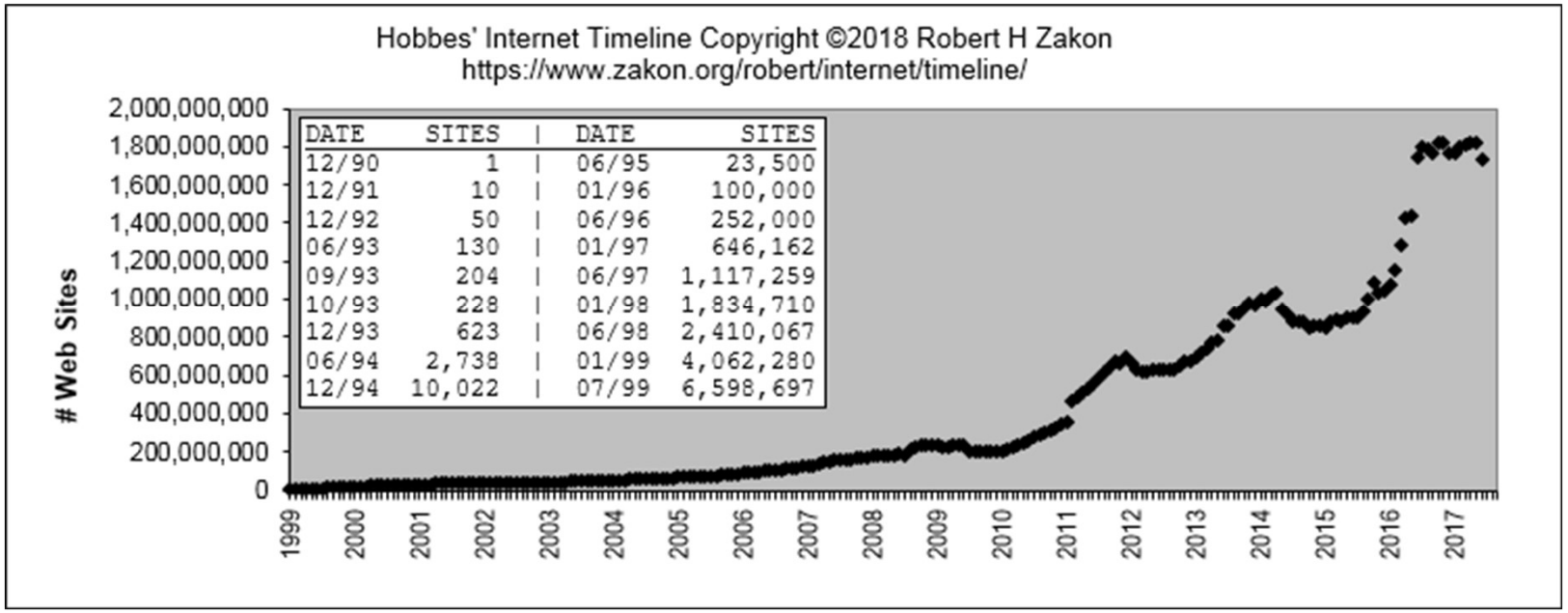
Quelle: Statcounter.com, Oktober 2021

Das Web in Zahlen…

Anzahl der Websuchen pro Tag (2019)

→ Google verarbeitet 3.5 Milliarden Suchen/Tag

(internetlivestats, 2019)



DATE	SITES	DATE	SITES
12/90	1	06/95	23,500
12/91	10	01/96	100,000
12/92	50	06/96	252,000
06/93	130	01/97	646,162
09/93	204	06/97	1,117,259
10/93	228	01/98	1,834,710
12/93	623	06/98	2,410,067
06/94	2,738	01/99	4,062,280
12/94	10,022	07/99	6,598,697

Das Web in Zahlen…

- Zusammensetzung des “surface web”

Typ	% (totale Dateigrösse)
Bilder	23.2%
HTML	17.8%
PHP	13.0%
Adobe PDF	9.2%
Videos	4.3%
Komprimiert/Archiv	3.7%
Audio	2.3%
Programme	1.4%

Nicht nur Google!

- Verschiedene spezialisierte Suchdienste
  - Allgemein
  - Nachrichten
  - Shopping
  - Für Kinder
  - Fachspezifisch (Medizin, öff. Behörden, Juristisch, QA, Reisen, …)
  - Bilder/audio/video
  - Metasearch
  - Länderspezifisch
  - Eigene Suchmaschine für eine Website
  - Enterprise Search (Web+Emails+Memos+…)

Nicht nur Google!

- ABER: Überschätzen Sie Web Search nicht!

- Benutzer gehen vermehrt direkt zur gesuchten Websites, als über den Umweg eines Suchdienstleisters.
- IDC behauptet, dass 70% aller Anfragen direkt bei den gesuchten Websites abgesetzt werden (→ Bedeutung Enterprise Search)

  (Quelle: "Who Owns the Web? Guess Again“, Frank Gens. 21/03/2007, http://blogs.idc.com/ie/?p=92)

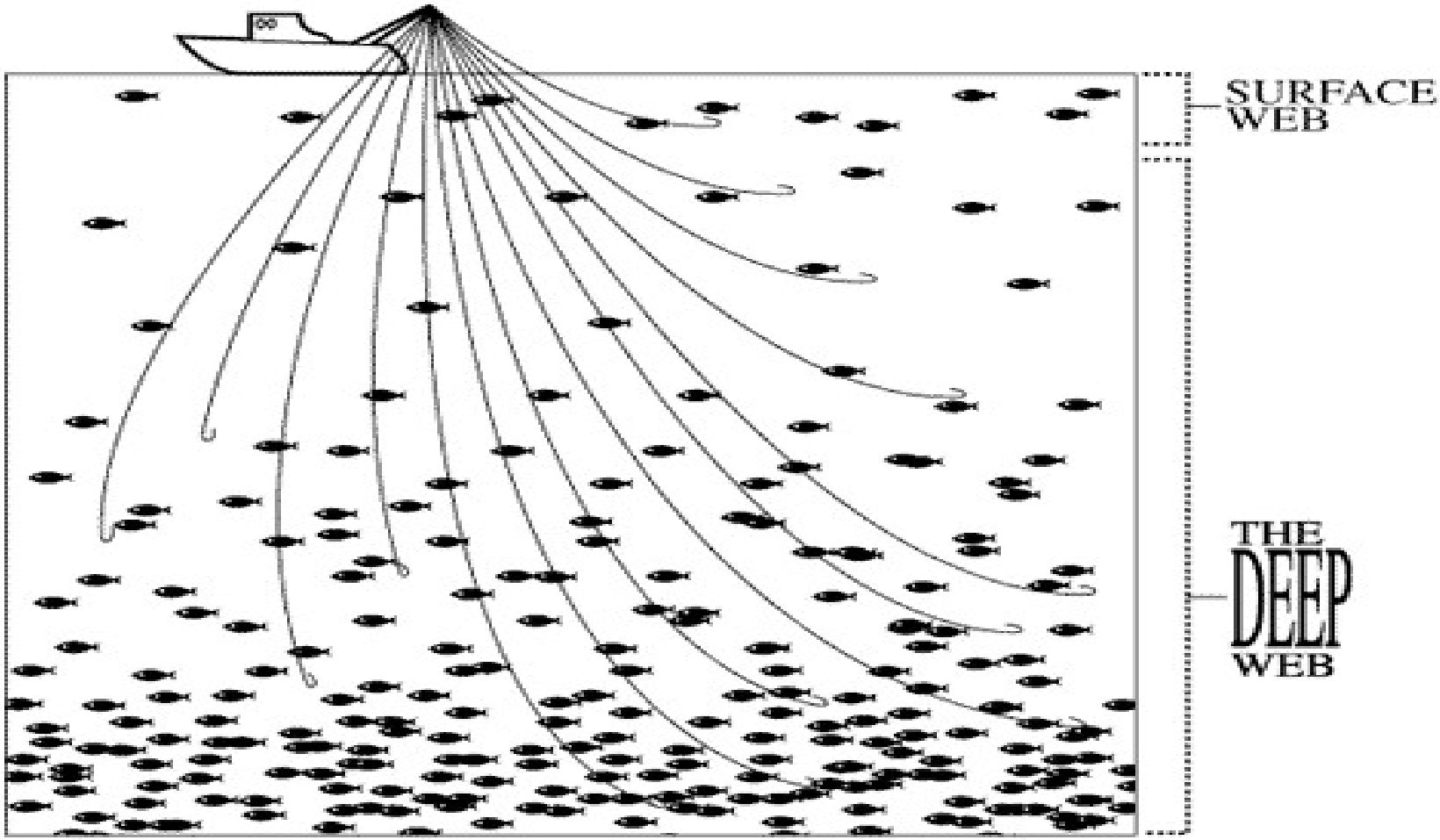
Klassisches IR

- Korpus:
  - Wohldefinierte, statische Dokumentenkollektion
  - Zentral, and einem bekannten Ort, gespeichert
- Kollektion ist homogen, typischerweise von hoher Qualität
- Ziel: Dokumente zurückgeben, die Information enthalten, welche relevant ist in Hinsicht auf das Informationsbedürfnis des Nutzers
- Klassische Definition von Relevanz
- Für jede Query Q und jedes Dokument D der Kollektion existiert nur ein Relevanzscore RSV(Q, D)
- Der Score ist ein Durchschnitt über alle User U und Kontexte C
- Wir optimieren also den RSV(Q, D) statt RSV(Q, D, U, C)
- → User und Kontext ignoriert, akzeptabel in einem kontrollierten Setting

Suche im Web

- Korpus: Das öffentliche zugreifbare Web besteht aus statischem und dynamischen Inhalt. Die Daten im Web sind sehr unbeständig (40% monatliche Änderung), unstrukturiert, teilweise von schlechter Qualität (Spam!) und/oder heterogen.
- Leute kommen auch zu Google oder Bing um im Web zu navigieren (schwierig den Anbieter direkt zu finden)
- Ziel: Finde qualitativ hoch stehende Resultate (nicht unbedingt Dokumente!), die für den Benutzer relevant sind.
- Resultat:
  - Statische Seiten (Dokumente) z.B. Texte, mp3, Photos, Videos ...
  - Resultat ist nicht ausbeuteorientiert!
  - Ggf. Probleme mit "dynamischen Seiten": diese werden bei einer Anfrage durch den Benutzer generiert. Die Daten werden dynamisch aus einer Datenbank geladen und dargestellt. → „Deep Web“

Deep Web



Quelle: Rapporto tra Deep Web e Surface Web, Silvia Panzavolta, http://www.indire.it/content/index.php?action=read&id=1303

Suche im Web

- Bedürfnis (Einteilung nach A. Broder, damals bei Altavista)
  - **informationell**
    - man will etwas lernen (~40%) z.B. “Was ist das Semantic Web?”
  - **navigational**
    - man will zu einer gewissen Webseite (~25%) z.B. ”Tschechien Bahn”
  - **transaktional**
    - man will etwas tun (~35%)
    - Zugriff auf einen Service z.B. “Wetter in Zürich”
    - Downloads, z.B. “Oberflächenfotos vom Mars”
    - Shop, z.B. “iTunes”
  - **Graubereiche**
    - Finde einen guten Hub, z.B. “Automiete in Seattle”
    - Erkundungssuche “see what’s there”

Entwicklung von Suchmaschinen

- **Erste Generation** – nur einfache Webseiten, Textdateien
  - Worthäufigkeit, Sprache
  - AltaVista, Lycos, Excite
- **Zweite Generation** – „off-page“, web-specific data
  - Hyperlink- oder Verbindungsanalyse
  - Click-through data (Angeklickte Resultate)
  - Ankertext (wie Leute auf diese Webseite verweisen)
  - Google (1998) mit PageRank-Algorithmus
- **Dritte Generation** – Beantwortet “das Bedürfnis hinter der Anfrage” (immer noch im Fluss, die heutigen Systeme setzen diese Idee zunehmend um) (nimmt stärkere Rücksicht auf die Unterscheidung informationell/navigational/-transaktional)

Anatomie von Suchmaschinen

- **Spider** (Crawler oder Roboter) – bildet den Korpus
  - Sammelt die Daten rekursiv
    - Für jede bekannte URL, hole die Webseiten, parse diese und extrahiere die neuen URL's.
  - Zusätzliche Daten von direkten Einträgen und anderen Quellen.
  - Verschiedene Suchmaschinen haben unterschiedliche Grundsätze – kleine Übereinstimmung unter den Korpora
- Der **Indexer** – verarbeitet die Daten (inverted files)
  - Verschiedene Grundsätze, welche Worte indexiert oder gestemmt werden, Phrasenunterstützung, Grossschreibung, Unicodeunterstützung etc.
- **Anfrageverarbeitung** – akzeptiere Anfrage und liefere Resultat zurück
  - Front end – macht Anfragereformulierung – Stemming, Grossschreib-Regeln, Boolean-Optimierung, Zusammensetzung, etc.
  - Back end – findet passende Dokumente und rangiert diese

Spidering

- Starte mit einer umfassenden Menge von URLs, von welchen die Suche (Spidering) gestartet wird (seeds S0).
- Speichere die Dokumente in D und die Hyperlinks in E. Dabei ist sowohl D als auch E ein eigenständiger Datenbehälter.
- Während dem Crawling wird eine Liste Q von URLs intern unterhalten.
- Wir extrahieren eventuell eine URL von Q oder fügen eine URL Q hinzu (Funktionen Dequeue() und Enqueue())
- Wir bezeichnen
  - u, v als eine URL
  - d(u) die zugehörige Webseite

Spidering

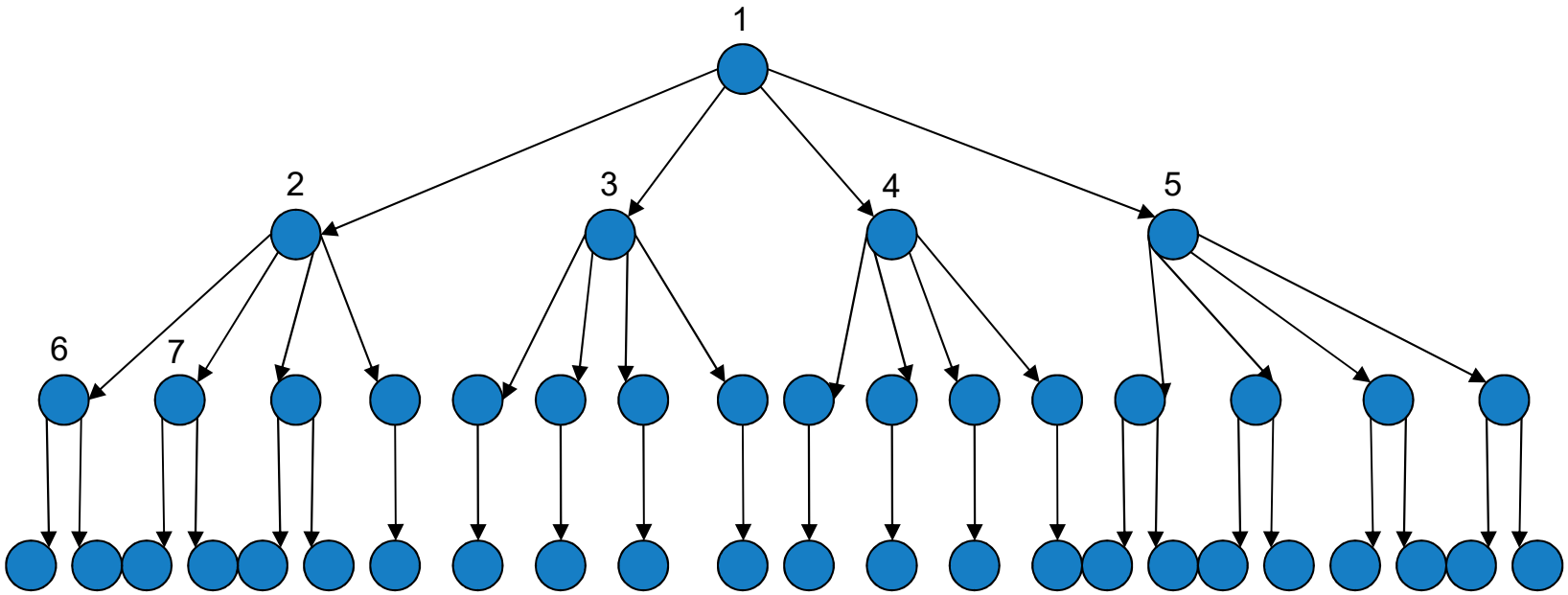
- Unser einfacher Webcrawler-Algorithmus:

Simple-Crawler (S_0, *D*, *E*)

```
Q ← S0;
while Q ≠ Ø do
  u ← Dequeue(Q);
  d(u) ← Fetch(u);
  Store (D, (d(u), u);
  L ← Parse(d(u));
  for each v in L do
      Store (E, (u,v));
      if ¬((v ∈ D) v (v ∈ Q)) then Enqueue(Q,v);
```

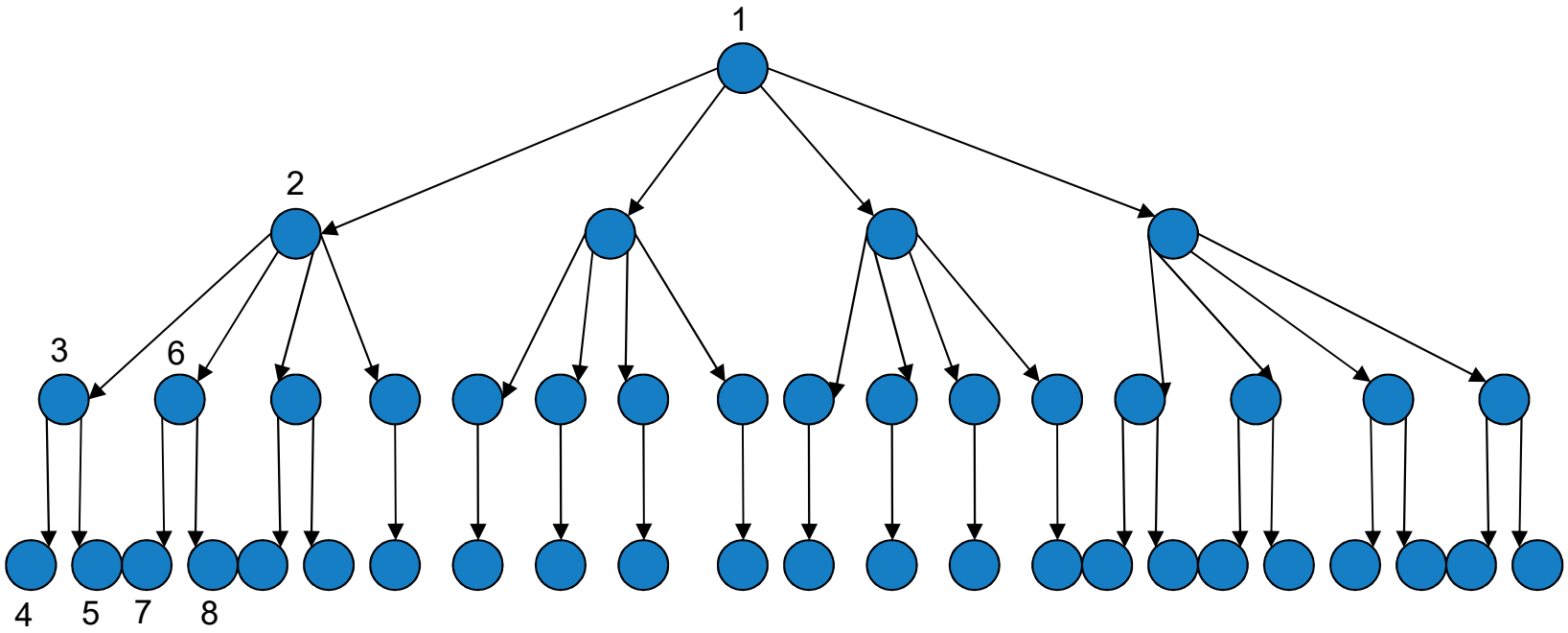
Spidering

Breitensuche (Breadth-first)



Spidering

Tiefensuche Suche (Depth-first)



Spidering (Erforschungsstrategie)

- Tiefen- und Breitensuche sind Suchalgorithmen um einen Knoten in einem Graphen zu suchen.
- Tiefensuche erfordert Speicher nur für die Tiefe (d) mal den Verzweigungsgrad (b) (O(bd)). → Algorithmus benötigt aber zu viel Zeit, um nur einem Zweig nachzugehen (Wie tief kann ein Ast sein?).
- Breitensuche erforscht von der Rootwebseite gleichmässig nach aussen. Dieser Algorithmus erfordert Speicher für alle Konten des Graphen von früheren Ebenen ($O(b^d)$).
- Kompromisslösung nötig
- Wie neue Links zu Q hinzugefügt (Enqueue()) und extrahiert (Dequeue()) werden hängt von der jeweiligen Suchstrategie ab.
  - FIFO („first in first out“ → wird am Ende von Q angehängt) → Breitensuche
  - LIFO („last in first out“ → wird am Anfang von Q angehängt) → Tiefensuche

Spidering (Erforschungsstrategie)

- Heuristisches Anordnen von Q ergibt einen fokusorientierten Crawler, der dadurch die Suche auf interessante Seiten ausrichtet.
- Der Spider kann auf eine bestimmte Webseiten begrenzt werden
  - Lösche Links zu anderen Seiten von Q
- Der Spider kann auf bestimmte Ordner begrenzt werden (z.B. Homepagesuche)
  - Lösche Links, die nicht im festgelegten Ordner liegen.
- Der Spider kann auf bestimmte Domänen begrenzt werden (Region, Land, Sprache), unter Annahme von genauen Themenbeschreibungen oder eine Menge von vorgegebenen und interessanten Webseiten .
  - Sortiere Q nach Ähnlichkeit (z.B. Kosinus). Als Sortierkriterien kann die ganze Webseite und/oder Ankertext des Themas verwendet werden.
  - Themenzuordnung durch Verfolgung und Entdeckung

Spidering (Erforschungsstrategie)

- Beobachten aller „in-degree“ und „out-degree“-Werte jeder Webseite.
- Beispiel:
  - Sortiere Q so, dass populäre Seiten (mit vielen in-coming Links) bevorzugt werden.
  - Sortiere Q so, dass zusammenfassende Seiten (mit vielen out-going Links) bevorzugt werden.
  - Verwende den Google-Algorithmus → PageRank
- Aber: Problem mit Start-Set S0 ("seeds") der URLs. Wie wählen?

Spidering (Praxis)

- Praktische Betrachtungen
  - Die Objekte D und E müssen auf einer Disk gespeichert werden (Grösse, Recovery,…)
  - Wir müssen die Schlaufe abrechen bevor Q = Ø terminiert (Zeit, Speichereinschränkungen)
  - Die Zeit, um ein Dokument herunterzuladen, ist unbestimmt (der Crawler kann nicht stur abwarten → gleichzeitiges herunterladen ist die übliche Lösung)
  - Indiziere alle neuen Dokumente sofort (nicht trivial, da Behinderung des Spidering- und Suchprozesses)
  - Eventuell wird den Webseitenbetreiben direkt erlaubt, ihre Webseiten anzugeben, die gespidert werden sollen.

Spidering (Praxis)

- Praktische Betrachtungen
  - Das Web enthält viel redundante Information, und der Linkgraph enthält Zyklen
  - URLs müssen in kanonische Form gebracht werden, um die Zyklen zu erkennen (trailing /, /index.html, …)
  - Duplizierte Seiten, Mirrors, …
  - Near-Duplicates (interessantes Forschungsproblem – das Erkennen von Near-Duplicates ist inhärent ein quadratischer Aufwand. Alternative: Fingerprints)

Spidering (Praxis)

- Praktische Betrachtungen
  - Crawler sollten respektvoll mit den verfügbaren Serverressourcen umgehen, und die Restriktionen der Webseitenbetreibern beachten. (Roboterausschluss, zwei Möglichkeiten)
    - Robots exclusion protocol (www.robotstxt.org): Umfasst Restriktionen, die für gewisse Verzeichnisse gesetzt werden können.
    - Roboter META-Tag: Individuelle Dokumenten-Tags, um Indexierung oder Verfolgung auszuschliessen.
  - Webseitenadministratoren fügen ein “robots.txt”-File in das Rootverzeichnis ihrer Webseiten (z.B. www.foo.bar/robots.txt)
  - Probleme mit Links, die mit JavaScript, eingebettetem Flash etc. erstellt worden sind.

Spider (Anker)

- Extrahiere Ankertext (zwischen <a> und) für jeden Out-Link
- Normalerweise beschreibt der Ankertext das Dokument sehr gut, auf welches es zeigt (→ Schlüsselworte der Webseite, Ultra-Kurz-Zusammenfassung).
- Füge Ankertext hinzu, damit die Zielwebseite mit zusätzlichen Schlüsselworten versehen wird:
  - Software-Gigant
  - IBM
- Sehr effektiv, um relevante Information zu finden (wird u.a. von Google verwendet).
- Nicht immer hilfreich («Click here», «forward»)
- Google bietet teilweise Suchresultate, die nur aufgrund des Ankertexts zustande gekommen sind.
- → «Google Bombing»

Indexierung Webseiten

- Die exakte Indexierungsstrategie ist ein Geschäftsgeheimnis und variiert von Unternehmen zu Unternehmen. (Was erwarten Sie aber für eine Tendenz?)
- Extremfall: Man kann die Beschreibung jeder Seite auf die Kollektion der zugehörigen Ankertexte beschränken.
  - Riesige Reduktion der Dokumentenmenge (z.B. NTCIR-5 Webtrack von 1.5TB auf 80MB)
  - Geeignet, um Webseiten zu indexieren, die keinen Text enthalten (z.B. Multimedia-Daten)
- SPAM ist ein Problem, sowohl beim Spidern, als auch beim Indizieren.

- !!! Es gibt Probleme bei der Speicherung des Index (muss verteilt werden – Konsequenzen?)

Indexierung Webseiten

- HTML-Struktur kann genützt werden
- <TITLE>-Tag
- <H1>…<H5>, <STRONG>, <B>, <I>, ….
- <META>-Tags

- Wird nicht konsistent genutzt!

Anfrageverarbeitung

- Grosse Anzahl von Anfragen (in Millionen oder Milliarden pro Tag)
- Eine kleine Anzahl von häufigen Anfragen (80/20 Regel)
  - Periodische Themen (“chat”, “britney spears”)
  - Kurzlebige Themen (“tsunami”) → Kann mit “vorberechneten” Resultaten abgedeckt werden. Auch bei politisch sensiblen Resultaten (→ gerichtlich relevante, Zensur?)
    - Denken Sie daran: Suchresultate können grossen Einfluss ausüben
- Wenn die nächste Resultatseite angefragt wird (sehr häufige Anfragen): vorberechnen
- Falls die Anfrage gesucht werden muss → Cluster von PC’s
- Kann als Spezialfall von verteiltem IR angesehen werden (Probleme?)

Evaluation

- Intuition: Das Wachstum des Web macht es immer schwerer, relevante Seiten zu finden (Ausschuss nimmt zu)
- Aber: NEIN, siehe Suchparadox. Die “Precision@1” nimmt zu, wenn das Volumen zunimmt.
- (Einschränkung: es gibt bedeutend weniger Aussagen zum Thema “Ausbeute” – das ist auch ein Artefakt der verwendeten Testkollektionen)

WT10g

- Der erste “substantielle” Web-Korpus, 1.69M dicht verlinkte Seiten (TREC 9/10)
- 50 Topics/Queries, aus Search Engine-Logs abgeleitet
  - Irrelevant, relevant, hochgradig relevant
  - Die Betrachtung von “hochgradig relevant” ist wichtig [Gord99, Voor01]
- 145 Homepage-Suchanfragen (in TREC-10)
- Aber: gross genug? Genügend Links?

Evaluation mit WT10g

- 50 klassische Topics, aus SE-Logs
  - <title> Vikings in Scotland?
    <desc> What hard evidence proves that the Vikings visited or lived in Scotland?
    <narr> A document that merely states that the Vikings visited or lived in Scotland is not relevant. A relevant document must mention the source of the information, such as relics, sagas, runes or other records from those times.
  - <title> halloween?
    <desc> When, where, and how did Halloween evolve?
    <narr> A relevant document will discuss the origin of Halloween and the original customs of Halloween. Modern day trick-or-treating stories are not relevant.
- 145 Homepage-Suchen
  - “Information Technology Institute” (www.iti.gov.sg)
  - “Digital realms” (www.drealms.co.uk)

Erkenntnisse

Homepage-Suchen (145 Anfragen in TREC-10)

Okapi system	MRR
+ SMART stemmer	0.261
No stemming	0.274
No stemming + proximity	0.367
+ URL length	0.653
Nostem+prox+URL	0.693

J. Savoy, Y. Rasolofo: Report on the TREC-10 Experiment: Distributed Collections and Entrypage Searching. Proceedings TREC'10, Gaithersburg (MD), 2002, 586-595

.GOV-Kollektion

- .GOV corpus, 1.25M Seiten, Crawl der .gov-Domain, erstellt für TREC 11/12
- 50 Standardsuchen
- 150 Site-Suchen
- 50 Anfragen für "topic distillation" (“Dossier zusammenstellen”)
- Nicht wirklich repräsentativ, aber interessant
- Realistischer Web-Graph

Evaluation (.GOV)

- 50 Standardanfragen, aus SE-Logs
  - <title> intellectual property
    <desc> Find documents related to laws or regulations that protect intellectual property
    <narr> Relevant documents describe legislation or federal regulations that protect authors or composers from copyright infringement, or from piracy of their creative work. These regulations may also be related to fair use or to encryption.
- 150 Site-Suchen
  - “US agriculture changes 20th century”
  - “US passport renewal”
  - “white house west wing history”

Evaluation (.GOV)

- 50 Anfragen für “topic distillation” (TREC-12, 2003)
  - <title>cotton industry
    <desc> Where can I find information about growing, harvesting cotton and turning it into cloth?
  - Ein Dossier zusammenstellen (nicht *alle* Antworten)
  - Nicht nur eine einzelne Site (→ Site-Suche)
- Erfolgsfaktoren
  - Web-Struktur (<`title`>, Meta)
  - Ankertext
  - Indegree
  - URL (Länge)

ClueWeb – 1 Milliarde Dokumente

- Experimente in Hinblick auf Big Data-Phänomene werden erst mit sehr grossen Testdatensets möglich.
- ClueWeb mit ~1 Milliarde Dokumenten/Webseiten (multilingual)
- Topics aus Logs von Websuchdiensten hergeleitet

- Zum Teil andere Form der Evaluation, da keine klassichen Relevance-Assessments

Suchmaschinen (Schlussfolgerung)

- Spidering ist ein wichtiger Aspekt
- Ranking ist der Schlüssel (eine gute Antwort in den Top 5/10)
- Verwendung von verschiedenen Merkmalen → präzisionsorientiert!
  - Webseiteninhalt (+tags)
  - Meta tags + <title>
  - Ankertext
  - Hyperlink-Struktur
  - URL (Länge, Text, Struktur)
- Stemming / Stoppwortliste verbessert nicht immer das Resultat im gewünschten Sinne
- Die Geschwindigkeit ist ein Schlüsselelement
- Wirtschaftlichkeit! → Wie teuer darf ein Suche sein?