Seminar: Algorithmen für das WWW

Die Suchmaschine Google

Volker C. Schöch
Institut für Informatik
Freie Universität Berlin
vschoech@inf.fu-berlin.de

19. Juni 2001

Zusammenfassung

Google ist die erste Suchmaschine, die ihre Wurzeln an einer Uni hat. Sie wurde ursprünglich von S. Brin und L. Page aus Stanford lediglich als *Proof of Concept* implementiert [BP98]. Den gesuchten Beweis für die implementierten Konzepte erbrachte sie aber so überzeugend, dass sie innerhalb weniger Monate zu einer der populärsten universellen Suchmaschinen wurde. Unter den von der Suchmaschine Google zu Testzwecken implementierten Konzepten sind insbesondere zwei Verfahren zum Ranking von Suchergebnissen, auf die in diesem Papier näher eingegangen wird: *PageRank* (Abschnitt 6) und *Anchor Text* (Abschnitt 7).

1 Problemstellung

Gegenüber klassischen Problemen aus dem *Information Retrieval* (IR) stellt das Web eine besondere Herausforderung dar. Es ist nicht nur vollkommen inhomogen in Struktur und Inhalt, sondern wirft allein durch seine schiere Größe eine ganze Klasse neuer Probleme auf. Ein Beispiel aus [BP98] macht das besonders deutlich: Im November 1997 hat nur eine der damals aktuellen Top-4 Suchmaschinen auf die Eingabe ihres Namens ihre eigene Homepage unter den ersten 10 Ergebnissen gelistet.

Einer universellen Suchmaschine bietet sich heute in etwa folgende Situation:

- Es gibt mehrere Milliarden Webpages (Verdoppelung alle 3–6 Monate).
- Mehrere hundert Millionen verschiedene Terme müssen indiziert werden.
- Pro Tag müssen hundert Millionen Suchanfragen beantwortet werden.

Trotz dieser Probleme bietet WWW auch einen großen Vorteil, der es eben erst zum *Web* macht: Die strukturelle Organisation als Hypertext.

- Aus der Struktur des Hypertext und der Benennung von Links lassen sich aussagekräftige Metainformationen ableiten.

2 Geschichte

Google ist hervorgegangen aus der ersten wissenschaftlichen Veröffentlichung, die sich explizit auf die Besonderheiten von IR im WWW bezog [BP98]. Vorher war Suchmaschinen-Technik im großen Maßstab ausschließlich von kommerziellen Interessen getrieben. Daher orientierten sich die Entwicklungen an dem zugrundeliegenden Geschäftsmodell (Banner-Werbung) und technische Details waren (und sind) in der Regel nicht öffentlich zugänglich.

Das Papier von Brin und Page wurde Ende 1997 geschrieben. Soweit möglich, habe ich Zahlen und Fakten für dieses Papier neu recherchiert, um den aktuellen Stand (Juni 2001) widerzuspiegeln. Wer sich für umfangreiches, aktuelles Zahlenmaterial interessiert, wird bei *Search Engine Watch* [Wat] fündig.

3 Definitionen

Um Suchmaschinen zu beurteilen und Suchergebnisse und Dokumente zu charakterisieren, haben sich bestimmte Begrifflichkeiten eingebürgert. Die folgenden Begriffe finden sich in verschiedenen Fachgebieten mit jeweils etwas unterschiedlicher Bedeutung. Diese Definitionen beziehen sich spezifisch auf den Kontext von Web-Suchmaschinen.

Precision. Anteil (hoch-)relevanter Suchergebnisse (unter den Top 10). Die Präzision macht also eine Aussage darüber, wieviele irrelevante Dokumente („falsche Alarme“) die Suchmaschine liefert.

Recall. Anteil der gefundenen relevanten Dokumente an der Gesamtheit aller relevanten Dokumente. Eine abgeschwächte Definition versteht Recall einfach als Gesamtzahl der gefundenen Dokumente. Der Recall (Übersetzung unmöglich) sagt noch nichts über die Güte der Treffer aus. Im Zusammenhang mit dem WWW hat Recall meist eine nachgeordnete Bedeutung, weil es i. d. R. eine „praktisch unendliche“ Zahl relevanter Dokumente zu einer Suchanfrage gibt.

Relevance. „Bedeutsamkeit“ eines Dokuments in einem bestimmten Kontext. Die Relevanz eines Dokumentes wird relativiert durch die Relevanz anderer Dokumente zur selben Anfrage. Nur die „besten“ Dokumente sollen in den Top 10 Suchergebnissen erscheinen, da es zehntausende von Dokumenten geben kann, die „irgendwie“ relevant sind.

Ranking. Mechanismus zur Relevanzbewertung von Suchergebnissen. Suchergebnisse werden in der Regel in absteigender Reihenfolge Ihres Ranking-Wertes sortiert dargestellt. Wie dieser Wert berechnet wird, ist oftmals entscheidend für den Erfolg einer Suchmaschine – und für den Erfolg einer Website, die für den jeweiligen Ranking-Algorithmus optimiert ist. Der konkrete Algorithmus ist deshalb eines der bestgehüteten Betriebsgeheimnisse eines jeden Suchmaschinenbetreibers.

Authority. Eine Webseite, die kompetente, aktuelle und verlässliche *Informationen* zu einem bestimmten Thema enthält. Kleinberg geht in [Kle99] davon aus, dass man gute Authorities an der großen Anzahl eingehender Kanten (Backlinks) von guten Hubs erkennen kann, wobei er nur den durch die jeweiligen Suchergebnisse gegebenen Teilgraph des WWW betrachtet.

Hub. Eine Webseite, die viele gute *Links* für ein bestimmtes Thema bietet. Hubs sind in gewisser Weise komplementär zu Authorities: Nach Kleinberg ([Kle99]) zeichnen sich gute Hubs durch eine große Anzahl ausgehender Kanten (Links) zu guten Authorities aus, wobei wieder nur der durch die jeweiligen Suchergebnisse gegebene Teilgraph des WWW betrachtet wird.

4 Spezielle Features von Google

Die Suchmaschine Google sticht offenbar in besonderer Weise aus der Masse der existierenden Suchmaschinen hervor und wurde so ohne kommerzielle Hintergedanken vom Geheimtip zu einer der populärsten Suchmaschinen überhaupt. Besondere Merkmale sind:

Einfache Bedienung. Die Google Suchseite kann weder Emails verschicken noch Kaffee kochen. Es gibt einfach einen Suchschlitz und zwei Buttons. Eingegebene Suchbegriffe werden grundsätzlich mit logischem *and* verknüpft. Daher lässt sich die Suchmaschine auch von Laien „ intuitiv“ bedienen.

Auch für Fortgeschrittene. Obwohl sie im Allgemeinen selten benötigt wird, unterstützt Google durchaus eine differenzierte Anfragesprache. Der interessierte User wird auf einer zusätzlichen Seite von einem aufwendigeren Formular dabei unterstützt, Anfragen in dieser Sprache zu formulieren. Wer die Sprache bereits beherrscht, kann sie ohne Umwege auch in dem einfachen Suchschlitz verwenden.

Übersichtliche Präsentation der Suchergebnisse. Die Anzeige der Treffer bietet alle wichtigen Informationen zur Beurteilung der Dokumente, ist aber gleichzeitig knapp und übersichtlich. Seiten, die vom gleichen Host kommen, sind optisch gruppiert.

I'm Feeling Lucky. Der eine der beiden Such-Buttons schickt die Suche ab und liefert ganz traditionell eine Liste mit Suchergebnissen zurück. Der andere Button heißt *I'm Feeling Lucky* und bringt einen direkt zu dem Treffer mit dem höchsten Rank. Das ist nützlich, wenn man genau weiß, wo man hin will, und wenn es sich dabei um eine sehr populäre Seite – z. B. die Homepage eines großen Unternehmens – handelt.

Über eine Milliarde URLs indiziert. Die Datenbasis von GOOGLE ist – soweit es von der Öffentlichkeit beurteilt werden kann – im Vergleich zu den Indizes der anderen Suchmaschinen z. Zt. mit Abstand die größte: 705 Millionen Seiten wurden besucht. Da GOOGLE auch URLs aus Links in den Index aufnimmt, umfasst der Index insgesamt über 1,3 Milliarden verschiedene URLs.

Ranking - das Erfolgsgeheimnis. Aufgrund der enormen Größe des WWW ist „gutes" Ranking entscheidend für den tatsächlichen Nutzen einer Suchmaschine. Ein guter Ranking-Algorithmus muss einer ganzen Reihe von Anforderungen gerecht werden und bietet sich insofern als Forschungsgegenstand an. Aus solchen Forschungen ist GOOGLE hervorgegangen. Von daher finden in GOOGLE besonders interessante Ranking-Verfahren Anwendung, auf die ich in Abschnitt 5 näher eingehe.

Index plus Cache. GOOGLE indiziert Seiten nicht nur, sondern verwaltet eine eigene Kopie von jeder besuchten Seite. Dadurch werden insbesondere weiterführende wissenschaftliche Arbeiten auf der bekannten Datenbasis – also unter realistischen *und* kontrollierten Bedingungen – ermöglicht.

Ähnliche Seiten finden. Zu jedem Treffer gibt es einen Link, der eine Liste von „ähnlichen Seiten" erzeugt. Da die Ähnlichkeit von Seiten aus der Linkstruktur des WWW abgeleitet wird, werden so auch relevante Seiten gefunden, die nicht notwendigerweise die eingegebenen Schlüsselworte enthalten. Neben anderen haben sich Dean und Henzinger mit Algorithmen befasst, die das leisten können [DH99].

Sprachbarrieren überwinden. Seit kurzem bietet GOOGLE die automatische Übersetzung von HTML-Dokumenten aus dem Englischen und Französischen ins Deutsche sowie aus einer ganzen Reihe von anderen Sprachen ins Englische an. Dieses Feature, das sich ausdrücklich in der Erprobungsphase befindet, ist vielleicht ein Zeichen dafür, dass das GOOGLE-Team immer noch kreativ und experimentierfreudig ist. Vielleicht ist es aber auch nur ein Zeichen dafür, dass GOOGLE aus dem wissenschaftlichen Umfeld heraus und in den Sog des Wettrüstens zwischen kommerziellen Suchmaschinen geraten ist.

5 Ranking

5.1 Was ist Ranking?

Angesichts der Größe und Inhomogenität des WWW reicht für eine erfolgreiche Suche eine bloße Trennung zwischen relevanten und nicht-relevanten Seiten nicht aus. Entscheidend ist die Reihenfolge bei der Präsentation der Treffer, denn nur wenn das erste Dutzend der angezeigten Treffer schon hilfreiche Suchergebnisse enthält, hat die Suchmaschine einen echten praktischen Nutzen.

Um dieser Problematik gerecht zu werden, arbeiten Suchmaschinen für das WWW seit jeher mit verschiedenen Heuristiken, um die Treffer zu einer Suchanfrage bewerten und in eine Reihenfolge bringen zu können. Beispielsweise wurden oder werden folgende Heuristiken von manchen Suchmaschinen angewandt:

- Je mehr Begriffe aus der Suchanfrage im Titel einer Seite auftauchen, desto relevanter scheint die Seite für die jeweilige Anfrage zu sein. Analog kann man dies für gewisse Meta-Tags wie *description* und *keywords* annehmen.
- Je häufiger ein Suchbegriff innerhalb einer Seite vorkommt, desto relevanter scheint diese Seite für diese Anfrage zu sein. Dabei werden i. d. R. die verschiedenen Stellen des Auftretens – z. B. Titel, Meta-Tags, Überschriften, Fließtext – unterschiedlich gewichtet.
- Je kürzer eine URL ist, desto bedeutsamer scheint die dazugehörige Webseite zu sein.

5.2 Eigenschaften guter Ranking-Algorithmen

Es war das erklärte Ziel der wissenschaftlichen Arbeit, aus der GOOGLE entstanden ist, „gute“ Ranking-Algorithmen zu finden und in einer realistischen Anwendung zu erproben [BP98]. Gute, d.h. praxisgerechte Ranking-Algorithmen zeichnen sich durch folgende Eigenschaften aus:

Geschwindigkeit. Die Antwortzeit einer Suchmaschine ist – zusammen mit einem guten Ranking – eines der wichtigsten Kriterien für die Nutzer-Akzeptanz. Daher müssen zeitaufwendige Berechnungen offline im Voraus vorgenommen werden. Alle Algorithmen, die online bei der Bearbeitung einer Suchanfrage laufen, müssen extrem schnell sein.

Skalierbarkeit. Das WWW übertrifft schon jetzt im Umfang alle praktisch relevanten Datenbanken, und die Anzahl der Dokumente im Web verdoppelt sich etwa alle 3–6 Monate. Ähnliches gilt für die Nutzerzahlen des WWW. Damit eine Suchmaschine in Zukunft noch nutzbar bleibt, müssen die verwendeten Algorithmen extrem gut skalieren.

Spamresistenz. Es gibt bei Suchmaschinen immer zwei Gruppen von Interessenten: Die Suchenden und die Gefundenen. Für viele Websites ist die Anzahl der Besucher gleichbedeutend mit Umsatz und Geschäft. Daher setzen die Betreiber dieser Websites alles daran, die Ranking-Algorithmen der großen Suchmaschinen gut kennen zu lernen und ihre Seiten darauf zu optimieren. Für Spitzenpositionen in der Trefferliste sind oft auch sehr merkwürdige Mittel und Wege recht – klassische Beispiele sind „Text in Hintergrundfarbe“ oder auch spezielle „Brückenseiten“, die menschliche Internetnutzer nicht zu Gesicht bekommen. Im Extremfall führt das zum sog. *Index-Spamming*: Die „Treffer“ einer Suchmaschine werden unbrauchbar, weil nicht wirklich relevante Seiten als erstes aufgeführt werden, sondern solche, die den Index am erfolgreichsten manipuliert haben. Um das zu verhindern, sollte ein guter Ranking-Algorithmus schwer zu manipulieren – also *spamresistent* – sein.

Plausibilität. Das einzige, was für den Anwender letztlich zählt, ist die subjektive Zufriedenheit. Um das zu erreichen, müssen die Prinzipien, nach denen eine Suchmaschine das Ranking der Treffer durchführt, dem Anwender plausibel und sinnvoll erscheinen. Ein theoretisch perfekt durchdachtes Ranking nützt nichts, wenn der Anwender das Ergebnis nicht nachvollziehen kann.

5.3 Interpretation der Suchanfrage

Eine Vorstufe zum Ranking ist die Entscheidung darüber, welche Seiten überhaupt als Treffer erkannt werden. Schon hier legt GOOGLE einen strengen Maßstab an, um möglichst viele irrelevante Seiten von vornherein auszuschließen. Danach wird eine erste Sortierung vorgenommen, die berücksichtigt, wie genau die Treffer der gegebenen Suchanfrage entsprechen.

1. Es werden nur Webseiten berücksichtigt, auf denen *alle* eingegebenen Suchterme vorkommen.
2. Seiten, auf denen die Suchterme in der *Eingabe-Reihenfolge* vorkommen, werden stärker gewichtet.
3. Seiten, auf denen die Suchterme in *räumlicher Nähe* zueinander vorkommen, werden stärker gewichtet.

5.4 Interpretation der Hypertext-Struktur

Erst mit der Vorstellung von GOOGLE wurde es unter den Suchmaschinen populär, auch Informationen zum Ranking zu nutzen, die implizit durch die Hypertext-Eigenschaft des WWW gegeben sind. GOOGLE verwendet hier zwei Verfahren, die nachweisbar besonders plausible und hilfreiche (also „gute“) Resultate liefern.

PageRank. PageRank macht sich den gerichteten Graphen zu Nutze, der von Referenzen (Links) im WWW aufgespannt wird. Auf diese Weise wird die „allgemeine Bedeutsamkeit“ einer Seite unabhängig von einer konkreten Suchanfrage bestimmt (s. Abschnitt 6).

Anchor Text. Anchor Text meint die Auswertung von Link-Bezeichnungen für die Indizierung einer Seite. So wird erreicht, dass die gewichteten Index-Terme einer Seite diese inhaltlich angemessen repräsentieren (s. Abschnitt 7).

5.5 User-Feedback

Es ist naheliegend, dass nicht ein Ranking-Verfahren alleine universell optimale Ergebnisse liefern kann. Ebenso ist sofort einsichtig, dass es bei der Implementierung eines jeden einzelnen Verfahrens sowie bei der Kombination dieser Verfahren zur Berechnung eines Gesamt-Rank-Wertes eine Unzahl von Parametern gibt, die zunächst einmal recht willkürlich festgelegt werden.

Einen systematischen Weg zur Bestimmung der optimalen Parameter hat auch das Team von Google noch nicht gefunden. Um zumindest gewisse Anhaltspunkte über den Einfluss der Parameter zu gewinnen, bekommen ausgewählte, vertrauenswürdige Nutzer eine spezielle Ergebnisseite mit einem Formular, das ein Ranking durch den Menschen erlaubt. Alle so gesammelten Informationen werden gespeichert. Dadurch wird es möglich, bei Veränderungen von Parametern den Einfluss auf alle gespeicherten Rankings zu rekonstruieren und auf diese Weise das Verhalten des Ranking-Algorithmus' an die menschlichen Einschätzungen anzunähern.

6 PageRank

Das *PageRank*-Verfahren heißt nicht etwa so, weil es den Rank von „Seiten" ermittelt, sondern weil es zuerst in einer Arbeit von L. Page et al. vorgestellt wurde [PBMW98]. Es gibt auch eine Online-Präsentation zu diesem Papier unter [Pag] sowie eine kurze, aber gut verständliche Einführung in [BC01].

6.1 Warum PageRank

Die Link-Struktur des WWW kann man als (gerichteten) Graphen auffassen. Dieser Graph enthält wertvolle, allgemein zugängliche, objektive Informationen über die Bedeutung einer einzelnen Webseite. Mit dem PageRank-Verfahren wird aus dieser Information eine Zahl („der PageRank") berechnet, die für eine Sortierung der Suchergebnisse verwendet werden kann.

Die Idee stammt ursprünglich aus der Zitat-Analyse wissenschaftlicher Literatur. So gab es bereits Spekulationen über den nächsten Nobelpreis-Gewinner, die allein darauf beruhten, auszuzählen, wie oft die Arbeiten der Kandidaten in anderen Arbeiten zitiert wurden [San95].

Es sollte klar sein, dass das WWW von wissenschaftlicher Literatur grundsätzlich verschieden ist. Insbesondere gibt es keinen übergreifenden thematischen Kontext. Ferner sind Veröffentlichungen im WWW fast kostenlos und vollkommen unkontrollierbar, so dass ein gewisser „Wildwuchs" in der Natur der Sache liegt. Eine Übertragung der Konzepte aus der wissenschaftlichen Zitat-Analyse in das WWW ist daher nicht trivial.

6.2 Verwandte Algorithmen

Das Webquery-System setzte diese Idee bereits 1995 für die Bewertung von Webseiten ein [CK97]. Allerdings wurde bei Webquery nur eine simple Zählung aller „Backlinks" vorgenommen. Dieses System ist mit etwas Aufwand relativ leicht zu manipulieren.

Der Companion-Algorithmus und der Cocitation-Algorithmus zur Bestimmung von „ähnlichen" Seiten gehen analog vor [DH99]. Hier wird die Linkstruktur zumindest über zwei Ebenen von Knoten analysiert, was die Manipulation durch einen einzelnen Webseiten-Betreiber schon erheblich erschwert.

Kleinbergs Algorithmus zur Bestimmung und Bewertung von Hubs und Authorities geht noch einen Schritt weiter: Gewichte werden rekursiv berechnet, jedoch nur innerhalb der Treffermenge einer konkreten Suchanfrage [Kle99]. Die Berechnung der Hub- und Authority-Gewichte muss daher online für jede einzelne Suchanfrage erfolgen, was potentiell die Antwortgeschwindigkeit beeinträchtigt.

6.3 Wie funktioniert PageRank

PageRank setzt diese Ideen konsequent fort: Für jede Seite in der Datenbasis, die eine möglichst große Probe des gesamten WWW repräsentieren sollte, wird aus der Struktur ihrer Referenzen (Backlinks) iterativ ein globaler „Bedeutsamkeitswert" berechnet.

PageRank basiert, ähnlich wie die vorgenannten Algorithmen, auf folgenden Annahmen:

- Die Autoren von Webseiten machen implizit eine Aussage über ihre (subjektive) hohe Meinung von gewissen anderen Webseiten, auf die sie mit Links verweisen.
- Die Gesamtheit der subjektiven Wertschätzungen von Web-Autoren kann man als objektive Bewertung einer Seite auffassen.
- Je mehr Links auf eine bestimmte Seite verweisen, desto „ bedeutender“ scheint diese Seite zu sein (z. B. Netscape Homepage).
- Je weniger Links eine Seite enthält, desto „bedeutender“ ist jeder einzelne Link.
- Je „bedeutender“ eine Seite ist, desto bedeutender sind die auf ihr enthaltenen Links (z. B. Seiten, die von Yahoo referenziert werden).
- Je „bedeutender“ die Links sind, die auf eine bestimmte Seite zeigen, desto „bedeutender“ scheint diese Seite zu sein.

Es wird deutlich, dass der PageRank-Algorithmus iterativ arbeiten muss. Vereinfacht lässt sich der Algorithmus wie folgt beschreiben:

1. Jeder Knoten (Seite) wird mit einem Startwert initialisiert. Grundsätzlich können die Startwerte beliebig gewählt werden, da der Algorithmus in (praktisch) jedem Fall konvergiert. Allerdings hat die Wahl der Startwerte wesentlichen Einfluss darauf, wie schnell eine akzeptable Konvergenz erreicht wird. Da man den PageRank gerne auch als Wahrscheinlichkeitsmaß auffassen möchte, initialisiert man die Knoten z. B. mit $\frac{1}{Anzahl\,Knoten}$.
2. Aus den Gewichten der Knoten werden die Gewichte der *ausgehenden* Kanten (Links) bestimmt als $\frac{Gewicht\,des\,Knotens}{Anzahl\,Links}$.
3. Aus den Gewichten der *eingehenden* Kanten (Backlinks) werden die Knotengewichte neu berechnet als $\sum Kantengewichte$.
4. Dieses Verfahren wird ab 2. sooft wiederholt, bis die Knotengewichte konvergiert sind bzw. bis eine hinreichende Annäherung erreicht ist.

Eine stabile Zuordnung von Knoten- und Kantengewichten in einem kleinen Beispielgraphen zeigt Abb. 1.

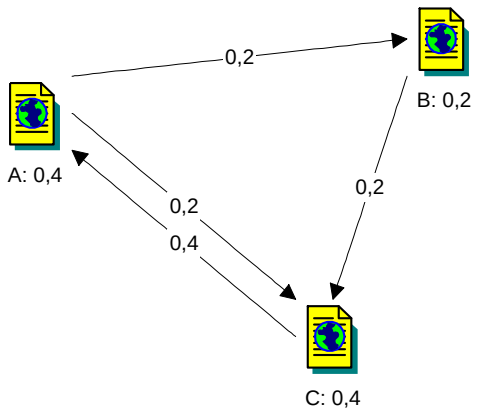

Abbildung 1: Ein Linkgraph mit Kanten- und Knotengewichten im Gleichgewicht.

6.4 PageRank mathematisch

6.4.1 Vereinfachte Definition

Aus der verbalen Beschreibung lässt sich eine (etwas vereinfachte) mathematische Definition ableiten.

Sei u eine Webseite und F_u die Menge der Seiten, die von u gelinkt sind. B_u sei die Menge der Seiten, die Links auf u enthalten. $N_u = |F_u|$ sei der Anzahl der Links in u. Da nicht alle Seiten Links enthalten, brauchen wir noch einen Faktor $c < 1$, damit die Summe des Rank-Wertes aller Webseiten konstant bleibt. Dann definieren wir den vereinfachten PageRank R' wie folgt:

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} \tag{1}$$

6.4.2 Vereinfachte Berechnung

Mathematisch betrachtet entspricht der PageRank dem dominanten Eigenvektor der $n \times n$–Matrix, die durch die initialen Kantengewichte gegeben ist ($n = |Knoten|$). Zur Erinnerung: Der *dominante Eigenvektor* ist der Eigenvektor mit dem größten *Eigenwert*.

$$\begin{aligned} Ap = \lambda p &\Rightarrow p \text{ Eigenvektor zu A} \\ \lambda \to max &\Rightarrow p \text{ dominanter Eigenvektor} \end{aligned} \tag{2}$$

Bezogen auf den Beispielgraphen von Abb. 1 könnte man mit folgender Matrix starten:

$$A = \begin{array}{c|ccc} & \text{A} & \text{B} & \text{C} \\ \text{A} & 0 & 0 & {}^1/_3 \\ \text{B} & {}^1/_6 & 0 & 0 \\ \text{C} & {}^1/_6 & {}^1/_3 & 0 \end{array} \tag{3}$$

Nach einer ausreichenden Zahl von Iterationen findet man den dominanten Eigenvektor p:

$$\begin{aligned} \lambda &= 1 \\ p &= \begin{pmatrix} 0.4 \\ 0.2 \\ 0.4 \end{pmatrix} \end{aligned} \tag{4}$$

Zur Approximation des dominanten Eigenvektors gibt es in der mathematischen Literatur Standardverfahren (z. B. bei [HH91]).

6.4.3 Rank Sinks und Rank Source

Die vereinfachte Definition des PageRank hat einen entscheidenden Nachteil: Es können sog. *Rank Sinks* auftreten. Darunter versteht man eine Anordnung von Seiten, in der der Rank mit jeder Iteration kumuliert, aber kein Rank wieder abgegeben wird. Es handelt sich dabei um zyklisch verlinkte Seiten mit Eingang aber ohne Ausgang (Abb. 2).

Um den Verlust von Rank in Rank Sinks zu kompensieren, führen Page et al. *Rank Sources* ein. Eine Rank Source ist ein Vektor E, der jeder Seite bei jeder Iteration einen gewissen konstanten „Bonus“ gibt. Wenn E nur positive Werte enthält, muss der Normalisierungsfaktor c nun noch kleiner gewählt werden, damit die Summe des im System vorhandenen Rank konstant bleibt. In der Praxis wählt man c in der Größenordnung von 0.85.

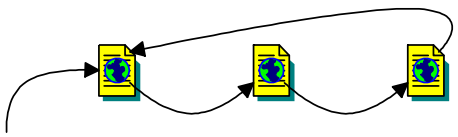

Abbildung 2: Eine Schleife, die Rank kumuliert („*Rank Sink*“).

6.4.4 Vollständige Definition des PageRank

Unter Berücksichtigung des Rank Source Vektors E können wir jetzt den PageRank definieren:

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} + cE(u) \tag{5}$$

In Matrix-Schreibweise haben wir die rekursive Formulierung:

$$\begin{aligned} R_i &= c(AR_{i-1} + E) \\ \|R\|_1 = 1 \Rightarrow R_i &= c(A + E \times \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}) \cdot R_{i-1} \\ \Rightarrow R_i &\rightarrow \text{Eigenvektor von } (A + E \times \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}) \end{aligned} \tag{6}$$

6.5 Random Surfer Model

Eine Veranschaulichung und Rechtfertigung für diese Definition des PageRank ist das sog. *Random Surfer Model*. Dieses Modell beschreibt die Vorstellung, dass ein Webuser seine Session auf einer willkürlich gewählten Webseite beginnt, einigen Links folgt und nach einer gewissen Zeit willkürlich eine neue Seite wählt. Wenn man den PageRank als Wahrscheinlichkeitsverteilung versteht, beschreibt die *Rank Source* die Wahrscheinlichkeit, mit der der Surfer zu einem gegebenen Zeitpunkt eine bestimmte Seite als neuen Startpunkt wählt – auf diese Weise bleibt der Surfer nicht in einem *Rank Sink* gefangen. Der *PageRank* einer Seite beschreibt dann die Wahrscheinlichkeit, mit der sich der Surfer zu einem gegebenen Zeitpunkt auf dieser Seite befindet.

6.6 Dangling Links

Ein weiteres Problem, dass sich in der Praxis stellt, beruht auf der Tatsache, dass man immer nur einen Ausschnitt des WWW lokal abbilden kann. Daher enthält die Datenbasis i. d. R. auch Links zu Seiten, die ihrerseits nicht in der Datenbasis vorhanden sind. Page et. al. [PBMW98] lösen dieses Problem sehr pragmatisch: Diese sog. *dangling links* werden für die Berechnung des PageRank entfernt und hinterher wieder eingefügt. Das Kantengewicht, das ja stets unter Berücksichtigung der Gesamtzahl der auf einer Seite vorhandenen Links berechnet wird, wird dadurch nur unerheblich beeinflusst. Nach dem Wiedereinfügen der dangling links kann man aus den bereits berechneten Ranks der in der Datenbasis erfassten Seiten den PageRank der nicht erfassten Seiten approximieren.

6.7 Bewertung von PageRank

Ziehen wir nun zur Beurteilung des PageRank-Algorithmus' die in Abschnitt 5.2 aufgestellten Kriterien heran.

Geschwindigkeit. Da der PageRank einer Seite unabhängig ist von einer konkreten Suchanfrage, kann er im Voraus offline berechnet werden. Insofern spielt die Effizienz des Algorithmus eigentlich eine untergeordnete Rolle. Dass die Implementierung von Page et al. [PBMW98] schnell genug ist für den praktischen Einsatz, wird durch Zahlen belegt: Auf einer Datenbasis von 25 Mio. erfassten Seiten und insgesamt 75 Mio. indizierten URLs benötigt eine „durchschnittliche" Workstation für eine Iteration etwa 6 Minuten – obwohl zu jedem Zeitpunkt nur die Hälfte des Graphen im Hauptspeicher gehalten werden kann. Dazu trägt wesentlich die sorgfältige Optimierung auf technischer Ebene bei; so wurden z. B. Plattenzugriffe konsequent linear angeordnet. Auf diese Weise ist es möglich, den gesamten Prozess zur hinreichenden Annäherung des PageRank für alle 75 Mio. URLs unter den genannten Bedingungen in nur 5 Stunden durchzuführen.

Skalierbarkeit. Eine genaue Berechnung des dominanten Eigenvektors im n-dimensionalen Raum mit $n \approx 25 \cdot 10^6$ ist in akzeptabler Zeit nicht realisierbar. Man begnügt sich daher mit einer hinreichend genauen Approximation. Als „hinreichend genau“ sehen Page et al. eine Gesamtabweichung pro Iteration von weniger als 100 an. Das entspricht durchschnittlich weniger als $4 \cdot 10^{-6}$ pro Seite bei einem durchschnittlichen PageRank von 1. Um diese Genauigkeit zu erreichen, sind nach Angaben der Autoren bei 161 Mio. URLs 45 Iterationen notwendig, für doppelt so vielen URLs werden 52 notwendige Iterationen angegeben. Damit kann man die Effizienz des verwendeten Algorithmus auf etwa $O(\log n)$ in der Eingabelänge abschätzen. Diese Abschätzung lässt sich untermauern durch theoretische Betrachtungen, die darauf beruhen, dass der vom WWW aufgespannte Graph einen hohen Expansionsfaktor aufweist [PBMW98]. Daher kann man davon ausgehen, dass der PageRank-Algorithmus auch auf extrem großen Datenmengen sehr gut skaliert.

Spamresistenz. Der PageRank einer einzelnen Webseite wird theoretisch durch den PageRank jeder anderen Seite im betrachteten Graphen beeinflusst. Dadurch ist es für einen einzelnen Webseiten-Betreiber sehr schwierig, den PageRank bestimmter Seiten zu manipulieren.

Plausibilität. Der Erfolg gibt GOOGLE recht: Offenbar werden die von GOOGLE erzeugten Suchergebnisse von vielen Anwendern als zutreffend und hilfreich empfunden. Die User-Feedback Strategie (vgl. Abschnitt 5.5) hilft dabei, diesen Aspekt des Rankings zu bewerten und zu verbessern.

7 Anchor Text Indizierung

Der Crawler von GOOGLE hat zum gegenwärtigen Zeitpunkt etwa 705 Mio. Seiten besucht. Der Index umfasst aber 1,3 Mill. verschiedene URLs. Diese Differenz erklärt sich dadurch, dass GOOGLE URLs aus Links mit den Termen indiziert, die den Link beschriften. Diesen Text nennt man *Anchor Text*.

Die Indizierung von Anchor Text birgt aber auch für tatsächlich besuchte Webseiten einen großen Nutzen.

7.1 Motivation

Man stelle sich die Suchanfrage „IBM“ vor. Mit hoher Wahrscheinlichkeit wird diese Anfrage durch die IBM Homepage optimal beantwortet. Allerdings

- könnte die IBM Homepage überwiegend grafisch gestaltet sein und kaum indizierbaren Text enthalten.
- kommt auf der IBM Copyright Seite der Term „IBM“ extrem häufig vor – diese Seite ist für die Suchanfrage aber wahrscheinlich eher uninteressant.
- könnte ein Mitbewerber von IBM den Term „IBM“ gezielt einsetzen, um Suchmaschinen zu „spammen“ mit dem Ziel, bei einer entsprechenden Anfrage möglichst weit oben in den Suchergebnissen zu erscheinen.

In diesem Fall würde ein klassisches Indexing und Ranking, das lediglich der Termfrequenzen innerhalb der indizierten Seite berücksichtigt, zu einem unbefriedigenden Suchergebnis führen.

7.2 Arbeitsweise des Anchor Text Verfahrens

Wie schon beim PageRank-Verfahren wird davon ausgegangen, dass der Autor einer Webseite andere Seiten implizit bewertet *und kommentiert*, indem er diese Seiten mit Links referenziert. Ferner unterstellt man, dass die Gesamtheit subjektiver Kommentare aller Webseiten-Autoren eine objektive Beschreibung der referenzierten Seiten ergibt. Wenn man nun die Labels der Backlinks einer Seite beim Indizieren dieser Seite mit berücksichtigt, dann spiegeln die gewichteten Terme im Index den Gehalt einer Seite wesentlich besser wider, als wenn lediglich Begriffe von der Seite selbst indiziert werden.

Um zum Beispiel zurückzukommen: Die Vielzahl von Links auf die IBM Homepage wird dazu führen, dass bei Anwendung des „Anchor Text“-Verfahrens eben diese Seite als besonders relevant eingestuft wird – und das ist genau das gewünschte Verhalten. :-)

Anchor Text Indexing wurde erstmals vorgestellt von McBryan [McB94].

8 System-Architektur

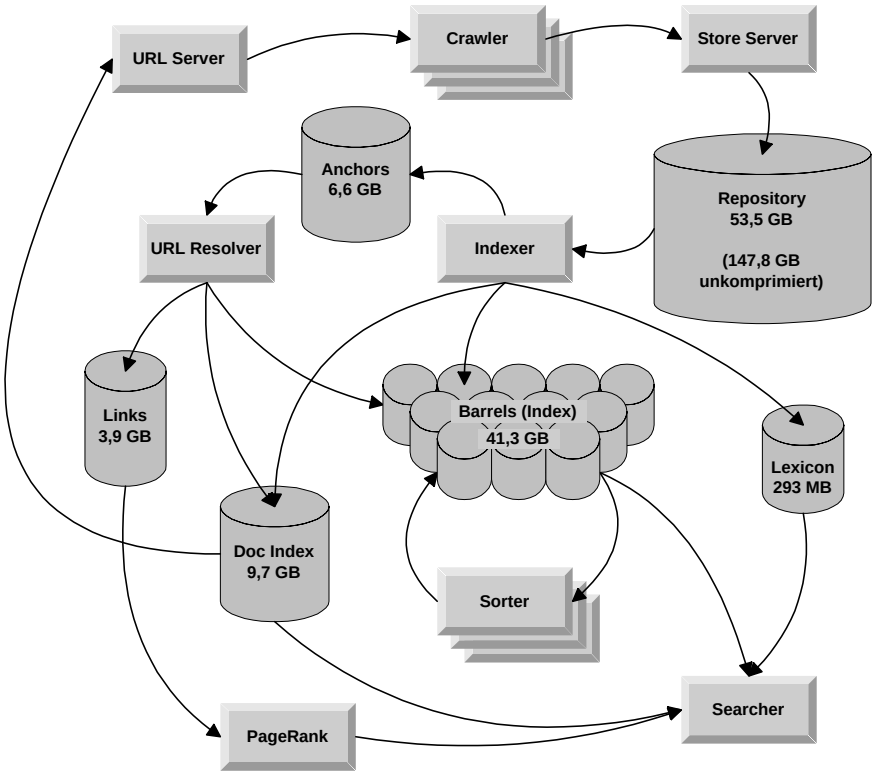

Abbildung 3: Die Google Systemarchitektur im Überblick. Größenangaben beziehen sich auf die Datenbasis von Nov. 1997 mit ca. 25 Mio. besuchten Seiten und 75 Mio. indizierten URLs.

Die System-Architektur und Implementierung von Google wird von Brin und Page detailliert erläutert [BP98]. Im Rahmen dieses Papiers möchte ich nur einen kurzen Überblick geben und einige besonders interessante Punkte herausgreifen.

8.1 Überblick

Abb. 3 gibt einen Überblick über die System-Komponenten der Google-Architektur und ihr Zusammenspiel. Die verschiedenen Prozesse und Speicher sind auf verschiedene Prozessoren bzw. Einzelrechner verteilt, so dass das Gesamtsystem hochgradig parallel arbeitet. Für das Design der Architektur hatte optimale Performance die oberste Priorität.

Crawler. Die *Crawler* besuchen systematisch alle Seiten, die ihnen vom *URL Server* vorgegeben werden.

Store Server und Repository. Jeder besuchte Seite wird vom *Store Server* komprimiert und im *Repository* abgelegt.

Indexer. Der *Indexer* von Google entspricht der Komponente, die in anderen Systemen als „Parser“ geläufig ist. Er liest die Seiten aus dem *Repository* und verarbeitet sie weiter:

- Links werden extrahiert und in die *Anchors* Datei geschrieben.

- Anhand der URL wird mit Hilfe des *Doc Index*' die *docID* bestimmt. Falls das Dokument noch nicht im Index ist, wird eine neue docID zugewiesen.
- Bedeutungstragende Terme werden extrahiert und mit Hilfe des *Lexikons* die *wordID* bestimmt. Neue Terme werden in das Lexikon eingetragen und erhalten eine neue wordID. Aus der wordID und einigen Meta-Informationen wie Ort des Auftretens, Schriftgröße, etc. wird ein *Hit* konstruiert.
- Die *docID* wird mit den dazugehörigen *Hits* in Form eines zunächst teilweise sortierten *Forward Index* in den *Barrels* abgelegt.

Barrels. Um einem hohen Grad an Parallelität und Prozess-Lokalität entgegen zu kommen, ist der Index auf viele relativ kleine *Barrels* aufgeteilt.

Sorter. Die *Sorter* arbeiten kontinuierlich daran, den nach der *docID* sortierten *Forward Index* in einen nach *wordID* sortierten *Inverted Index* umzuwandeln. Dabei werden gleichzeitig noch Platz- und Zugriffsoptimierungen vorgenommen. In anderen Architekturen wird diese Systemkomponente auch „Indexer“ genannt.

URL Resolver. Der *URL Resolver* berechnet aus den relativen Links, die in der *Anchors* Datei stehen, absolute URLs und ermittelt die dazugehörigen *docIDs*. Aus dieser Information wird ein Link-Graph konstruiert und in der *Links* Datei abgelegt.

PageRank. Mit Hilfe der *Links* Datenbank berechnet der *PageRank*-Prozess für jede im System vorhandene *docID* den dazugehörigen PageRank (vgl. Abschnitt 6).

Searcher. Während das gesamte übrige System kontinuierlich daran arbeitet, die Datenbasis zu aktualisieren und zu erweitern, stellt der *Searcher* die Funktionalität zur Verfügung, die man eigentlich von einer Suchmaschine erwartet: Er verarbeitet Suchanfragen, erzeugt Trefferlisten und sortiert sie nach Relevanz (vgl. Abschnitt 5).

8.2 Performance-Engpässe und Optimierungen

Es ist wesentliches Ziel des GOOGLE-Projekts, gigantische Datenmengen handhabbar zu machen und die Grenzen von Skalierbarkeit und Performanz auszuloten. Die Arbeiten an diesem Projekt führten zu einer großen Anzahl von innovativen und originellen Ideen. Eine kleine Auswahl der von Brin und Page [BP98] beschriebenen Konzepte möchte ich hier kurz vorstellen.

Disk Seeks. Während Prozessoren immer schneller und Hauptspeicher immer größer und billiger werden, stagniert die mittlere Plattenzugriffszeit bei etwa 10 ms. Da die von Google verarbeiteten Datenmengen auch für sehr große Hauptspeicher einfach zu umfangreich sind, spielen Plattenzugriffe für die Performanz des Gesamtsystems eine wesentliche Rolle. Daher wurden alle Algorithmen und Datenstrukturen konsequent daraufhin optimiert, Disk Seeks zu vermeiden und Daten möglichst linear zu lesen.

BigFiles. Der Umfang der Daten stellt nicht nur für den Hauptspeicher ein Problem dar. Vielmehr überschreiten die von GOOGLE benötigten Dateigrößen bereits die Grenzen von Dateisystemen und Festplatten. GOOGLE arbeitet deshalb auf dem eigenen virtuellen Dateisystem *BigFiles*. Mit dieser Komponente können sehr große virtuelle Dateien auf physikalischer Ebene Datei-, Platten- und Dateisystem-Übergreifend verwaltet werden.

Lexicon. Das Lexikon ist eine der am häufigsten benötigten Datenstrukturen. Es war daher ein wichtiges Optimierungsziel, die Daten im Lexikon so kompakt zu codieren, dass das komplette Lexikon jederzeit im Hauptspeicher gehalten werden kann.

Ein weiterer Engpass ergibt sich daraus, dass das Lexikon von vielen verschiedenen, parallel arbeitenden Prozessen benötigt wird – und zwar nicht nur zum Lesen, sondern insbesondere auch zum Schreiben. Bei klassischem Reader/Writer-Ausschluss würde diese Konstellation zwangsläufig zu Wartezeiten bei den schreibenden Prozessen führen. Im GOOGLE-System wird das Problem umgangen, indem für die schreibenden Prozesse ein hinreichend großer Puffer bereitgestellt wird. Eine eigene Systemkomponente – der sog. *DumpLexicon*-Prozess – kümmert sich dann darum, die Informationen aus den Schreibpuffern ohne Verklemmungen und Verzögerungen in das eigentliche Lexikon zu integrieren.

9 Grenzen der Google-Architektur

Nach Brin und Page [BP98] arbeitet das Gesamtsystem auf ausreichend viel Hardware in ungefähr linearer Zeit. Dies wird durch konsequente Parallelisierung ermöglicht. Bei einer Datenbasis deutlich jenseits der 100 Mio. Webseiten erwarten die Autoren für ihr System jedoch einen sprunghaften Anstieg der Komplexität, weil damit alle möglichen Beschränkungen auf Betriebssystem-Ebene überschritten werden könnten – z. B. der Adressraum des Speichers, die maximale Anzahl offener Dateien, die maximale Anzahl von IP-Sockets, usw.

Leider ist keine Bewertung dieser Situation aus heutiger Sicht verfügbar, denn offensichtlich ist es mit GOOGLE gelungen, die genannten Beschränkungen zu überwinden. Schließlich beansprucht GOOGLE heute für sich, über 700 Mio. Seiten in der Datenbasis zu haben. Einen Anteil an der Überwindung der kritischen Beschränkungen haben sicherlich neue oder individuell angepasste Versionen von Linux, die z. B. eine Vergrößerung des adressierbaren Speichers um mehrere Größenordnungen gegenüber früheren Versionen bieten. Hier hat es sich möglicherweise ausgezahlt, dass GOOGLE von Anfang an für Solaris und Linux konzipiert wurde, denn in das freie Betriebssystem lassen sich beliebige Erweiterungen und Spezialisierungen integrieren.

Literatur

[BC01] Krishna Bharat and Bay-Wei Chang. Web Search Engines: Algorithms and User Interfaces. tutorial at CHI 2001, Seattle, April 2001.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998. `http://www-db.stanford.edu/pub/papers/google.pdf`.

[CK97] J. Carriere and R. Katzman. WebQuery: Searching and visualizing the Web through connectivity. In *Proceedings of the 6th International WWW Conference*, 1997. `http://www.cgl.uwaterloo.ca/Projects/Vanish/webquery-1.html`.

[DH99] Jeffrey Dean and Monika Rauch Henzinger. Finding related pages in the world wide web. *WWW8 / Computer Networks*, 31(11-16):1467–1479, 1999. `http://www.research.digital.com/SRC/personal/monika/papers/monika-www8-%1.ps.gz`.

[HH91] Günther Hämmerlin and Karl-Heinz Hoffmann. *Numerische Mathematik*, volume 7 of *Grundwissen Mathematik*. Springer-Verlag, Berlin Heidelberg, 1989, 1991.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999. `http://www.cs.cornell.edu/home/kleinber/auth.pdf`.

[McB94] Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. In O. Nierstarsz, editor, *Proceedings of the first International World Wide Web Conference*, page 15, CERN, Geneva, May 1994. `http://www.cs.colorado.edu/~mcbryan/mypapers/www94.ps`.

[Pag] Lawrence Page. Pagerank: Bringing order to the web. online Powerpoint presentation at `http://hci.stanford.edu/~page/papers/pagerank/index.htm`.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, 1998. `http://stanford.edu/~backrub/pageranksub.ps`. There is an online Powerpoint presentation of this paper available [Pag].

[San95] N. Sankaran. Speculation in the biomedical community abounds over likely candidates for nobel. *The Scientist*, 9(19), October 1995. `http://www.the-scientist.com/yr1995/oct/index_951002.html`.

[Sch] Volker Schöch. Bookmarks zu Suchmaschinen. *Hier findet man alle Quellen aus diesem Papier übersichtlich zusammengestellt.* `http://bscw.gmd.de/pub/german.cgi/0/27304193`.

[Wat] Search Engine Watch. Search Engine Ratings, Reviews and Tests. *Aktuelle Daten und Fakten.* `http://searchenginewatch.internet.com/reports/index.html`.