licht, zählen HTML-Dokumente dennoch als schwach strukturierte Textdokumente. Textdokumente können jedoch von Information Retrieval Systemen nur immer dann optimal analysiert werden, wenn eine einheitliche und normierte Strukturierung vorhanden ist, die eine differenzierte Auswertung von Inhalten anhand von Strukturierungsregeln ermöglicht. Bei HTML-Dokumenten ist dies nicht der Fall, weshalb die Suchmaschinen unterschiedliche Verfahren der Dokumentenanalyse einsetzen müssen, um HTML-Dokumente *inhaltlich* erschließen zu können. Gleichfalls sind Gewichtungsverfahren erforderlich, die die einzelnen Dokumente *bezogen auf die Relevanz* zu einer Suchanfrage entsprechend sortieren. Eine Relevanz bezogene Sortierung bestimmt die Rangposition eines Dokuments innerhalb der Suchergebnisliste und wird *Ranking* genannt.

Links

Eigenschaften von HTML
- [www.teamone.de/selfhtml/tbae.htm]

HTML 4.01 Specification
- [www.w3.org/TR/1999/REC-html401-19991224/html40.txt]

Schema des Grundgerüsts einer HTML-Datei
- [www.teamone.de/selfhtml/tq.htm]

Allgemeine Regeln für HTML
- [www.teamone.de/selfhtml/bae.htm]

Respect Standards
- [www.w3.org/Talks/1999/0830-tutorial-unicode-mjd/slide99-0.html]

3 Funktionsweisen von Suchmaschinen

Eine genaue Kenntnis über die Funktionsweisen von Suchmaschinen schafft die Voraussetzung, Suchmaschinen auf technischer Ebene in Hinblick auf die Indexierung besser zu verstehen. Die einzelnen Prozesse und Verfahren, die Suchmaschinen einsetzen, um Informationen aus dem Internet zu generieren, sie in zulässige Daten und unzulässige Daten zu separieren, als auch Dokumente inhaltlich zu erschließen, bilden die technische Grundlage für die in Kapitel 6 bis 9 aufgeführten Empfehlungen der Website-Optimierung.

Eine Analyse der verschiedenen Verfahren, die zur Datenbeschaffung und dem Aufbau eines durchsuchbaren Datenbestands erforderlich sind, enthüllen die „Geheimnisse" der Suchmaschinen. Im Zuge der Darstellung der Funktionsweisen von Webrobots wird sehr deutlich, wie Suchmaschinen das Web durchsuchen und wie sie dabei Dokumente und andere Dateien im Internet finden.

Um Textdokumente in einen Datenbestand zu überführen, der durchsucht werden kann, müssen die Dokumente zunächst aufbereitet und in weiteren Prozessen analysiert werden. Im Mittelpunkt der nachfolgenden Betrachtungen stehen dabei die eingesetzten Methoden zur inhaltlichen Erschließung von Textdokumenten. Oder anders ausgedrückt, welche Verfahren setzen Suchmaschinen ein, um die natürliche Dokumentensprache als auch ihren Inhalt bzw. das Thema, das sie behandeln, erschließen zu können. Erst die Kenntnis der Methoden einer inhaltlichen Relevanzbewertung von Textdokumenten ermöglicht das Aufstellen von Handlungsanweisungen zur inhaltlichen Ausarbeitung von Dokumenten.

3.1 Webrobots und die Erfassung des WWW

Eine der wichtigsten Fragen bei der Erstellung eines Datenbestandes ist immer,

„... wie kann ein Datenbestand möglichst effizient und kostengünstig erzeugt und aktuell gehalten werden ...".

Grundsätzlich gilt die Regel, dass je überschaubarer und eindeutig identifizierbar ein Datenbestand ist, je homogener die Eingangsdaten sind und je besser abgrenzbar die technischen Rahmenbedingungen sind, desto eher ist es möglich, Daten organisiert und kostengünstig in einen Bestand aufzunehmen sowie über ihren Lebenszyklus hinweg periodisch zu aktualisieren.

Betrachten wir das WWW, so widersprechen die realen Gegebenheiten allen Anforderungen einer einfachen und effizienten Datengenerierung und Pflege. Im Internet befinden sich unterschiedlichste Dateiressourcen in einem weltweit verteilten Netzwerk. Dabei existieren vielfältige Datenquellen und Dateitypen, die in unterschiedlichen Programmiersprachen und Programmierstandards erstellt werden. Sie befinden sich nicht in einem eindeutig abgrenzbaren Systemumfeld. Darüber hinaus wächst die Anzahl der miteinander verbundenen Computersysteme, als auch die Menge der Dokumente und Ressourcen in einer bisher noch nie dagewesenen Geschwindigkeit. Die zeitnahe und vollständige Erfassung der Inhalte des World Wide Webs und damit verbunden die Erzeugung eines durchsuchbaren Datenbestands, ist somit für alle Recherchetools in Internet eine nicht ganz einfach Aufgabe.

Webkataloge begegnen dieser Problemstellung in der Form, dass sie die Verantwortung zur Generierung ihres Datenbestands sowie dessen Vollständigkeit grundsätzlich den Content-Anbietern überlassen. Der Aufbau des Datenbestands bei Webkatalogen erfolgt bekannterweise über die aktive Anmeldung einer Website. Die Webkataloge verfügen über keinen Mechanismus der dazu führt, Dokumente im Web selbstständig mit dem Ziel zu finden, sie in den Bestand aufzunehmen. Wird eine Website nicht aktiv bei einem Webkatalog angemeldet, erscheint sie auch nicht in dessen Datenbestand. Die damit verbundene außerordentlich geringfügige Erfassung des Internets, selbst durch sehr große Webkataloge wie Yahoo, ist eines der häufigsten Kritikpunkte an deren Suchergebnissen.

Diese unzureichende und unvollständige Erfassung des Internets stellt die Ausgangsüberlegung der Suchmaschinen für ihre Dienste dar. Erklärtes Ziel der großen Suchmaschinen ist es eben gerade, das WWW möglichst umfassend und vollständig zu erfassen und dabei Schritt zu halten, mit der Geschwindigkeit der Veränderungen an erfassten Dokumente sowie dem fortschreitenden Wachstum im Internet.

Die Aufgabe bedeutet zu aller erst ein System und Verfahren zu entwickeln das es ermöglicht, vorhandene und neu erzeugte Ressourcen im Internet umfassend zu identifizieren und über ihren Lebenszyklus hinweg auf Veränderungen erkennen zu können.

Die eingesetzte Lösung ist ein *Webrobot-System.* Häufig verwendete Synonyme für ein Webrobot-System sind auch *Webrobot, Robot, Web Wanderer, Web Crawler* oder auch *Spider*, die jedoch grundsätzlich die gleiche Art von System und Prozess beschreiben. Ein Webrobot ist diejenige Systemkomponente der Suchmaschinen, die für die Erfassung von neuen und veränderten Ressourcen im Internet verantwortlich ist.

Die Aufnahme in den Datenbestand einer Suchmaschine ist logischerweise die erste Voraussetzung, um bei Suchanfragen überhaupt berücksichtigt werden zu können. Da die Suchmaschinen versuchen, das Web zwar möglichst vollständig zu erfassen, dabei aber keine irrelevanten oder unerwünschten Inhalte zu speichern, verfügen die Webrobots über Systematiken, nicht erwünschte oder doppelte Inhalte, sowie technisch fehlerhaft erstellte oder mangelhaft angebundene Websites von der Indexierung auszuschließen.

Der Webrobot stellt folglich die erste Hürde für einen Content-Anbieter dar, der eine Website in den Datenbestand einer Suchmaschine aufgenommen haben möchte. Eine genaue Betrachtung der Arbeitsweisen von Webrobots scheint demzufolge ratsam. Hierdurch werden Fehler bei der Aufnahme vermieden und Websites können in Hinblick auf die Verwaltungsfunktionen der Webrobots optimiert werden.

3.1.1 Arbeitsweisen von Webrobots im Überblick

Ein *Webrobot* ist bei großen Suchmaschinen ein im Internet global verteilt arbeitendes Software- und Hardwaresystem, das das Internet konstant auf neue oder veränderte Dokumente und Ressourcen hin überprüft. Da ein Webrobot aus verschiedenen Hardware- und Softwarekomponenten bestehen kann, wird auch der Begriff des Webrobot-Systems verwendet. Im Client-Server-Modell entspricht der Webrobot dem *Client* und der Host einer Website dem *Server.*

Zur Überprüfung, ob sich im System erfasste Dokumente auch im Original verändert haben, werden alle erfassten Ressourcen in periodischen Abständen vom Webrobot-System wiederholt besucht und analysiert. Neue Ressourcen werden durch die Verfolgung von Hyperlink-Verweisen aus bereits indexierten Dokumenten erkannt und erfasst.

Webrobots sind von ihrer technischen Konzeption her grundsätzlich nicht nur auf das HTML-Dateiformat beschränkt. Über Einstellungsmöglichkeiten des Serversystems kann genau bestimmt werden, welche Dokumententypen von einem Robot erfasst werden sollen. Gleichfalls können Robots auch auf unterschiedlichen Anwendungsprotokollen arbeiten und neben HTTP-Servern auch FTP-, GOPHER-, WAIS- und NEWS-Server besuchen. Eine Restriktion auf bestimmte Dokumente- und / oder Protokolle erfolgt mittels Einstellungen mit dem Zweck, durch eine Beschränkung auf ausgesuchte Dateitypen eine Homogenität der Eingangsdaten und damit verbunden, einen hohen Eiffizienzgrad bei der Verarbeitung von Daten zu erreichen.

Betrachtet man das Gesamtsystem einer Suchmaschine, stellt das Webrobot-System grundsätzlich eine vom Retrievalsystem und Query-Processor eigenständige Systemkomponente dar. In Abhängigkeit der Gesamtkonfiguration des Suchmaschinensystems ergeben sich jedoch teilweise unterschiedliche Funktionen und Prozesse, die von den beiden Systembereichen Robotsystem und Retrievalsystem prozessual überlappend ausgeführt werden können. Überschneidungen kommen u.a. bei der Filterung und Datennormalisierung von Eingangsressourcen vor, die systemtechnisch grundsätzlich durch beide Systemkomponenten vorgenommen werden können.

Betrachtet man einen Webrobot in diesem Zusammenhang nicht nur als Softwareapplikation die Dateien im Internet sucht, sondern als ein System, das aus mehreren Komponenten und jeweils verschiedenen Funktionen besteht, wird der Begriff Webrobot fälschlicherweise mit dem *Gatherer* gleichgesetzt. Ein Gatherer ist eine eigenständige Systemkomponente innerhalb eines Webrobot-Systems, die den reinen Prozess des Sammelns von Dokumenten im WWW übernimmt. Zum Zweck einer deutlichen Abgrenzung der einzelnen Systemkomponenten kann ein Webrobot-System in vier Komponenten unterteilt werden:

Gatherer	sammelt Dokumente im WWW,
Loader	organisiert die auszuführenden Aufträge,
URL-Datenbank	verwaltet alle gespeicherten URL's,
Checker	wendet unterschiedliche Filter an.

Wie die einzelnen Komponenten zusammenwirken, kann anschaulich am Beispiel der Aktualisierung bereits erfasster Dateien dargestellt werden:

1. Aus einem bestehenden Bestand wird eine Liste der zu besuchenden URL's erstellt.
2. Diese Liste wird dem Loader übergeben, der die URL's entsprechend der Auslastung der einzelnen Gatherer verteilt und deren Abarbeitung überwacht.
3. Die Gatherer richten HTTP-Requests an die WWW-Server und übergeben die zurückgelieferten Daten an den Checker.
4. Werden nicht mehr existierende URL's erkannt, erfolgt eine Löschungsmeldung an die URL-Datenbank.
5. Der Checker hat die Aufgabe, über die Weitergabe der Eingangsdaten an das IR-System zu entscheiden. Er wendet unterschiedliche Filter auf die Eingangsressourcen an und gibt nur diejenigen Dokumente an das Retrievalsystem zur Indexierung weiter, die eine System individuelle Filterkette fehlerfrei durchlaufen haben.
6. In den Dokumenten gefundene neue Hyperlinks behandelt das System gesondert. Über sie können neue Ressourcen im WWW ausfindig gemacht werden. Mittels HTTP-Request besteht die Möglichkeit sie sofort auf ihre Existenz hin zu überprüfen. Je nach Systematik erfolgt entweder sofort ein Download (vollständiges Laden) der betreffenden Datei oder die neu gefundenen URL's werden zunächst an die URL-Datenbank zur Aufnahme und späteren vollständigen Indexierung übergeben.
7. Diejenigen Ressourcen, die den Systemvorgaben entsprechen, übergibt der Checker an das Information Retrieval System (IR-System) der Suchmaschine. In einem gesonderten Prozess werden dann die akzeptierten Dokumente vom IR-System analysiert und in den Datenbestand aufgenommen.

Nachfolgende Darstellung verdeutlicht die Systematik und das Zusammenwirken der vier Systemkomponenten.

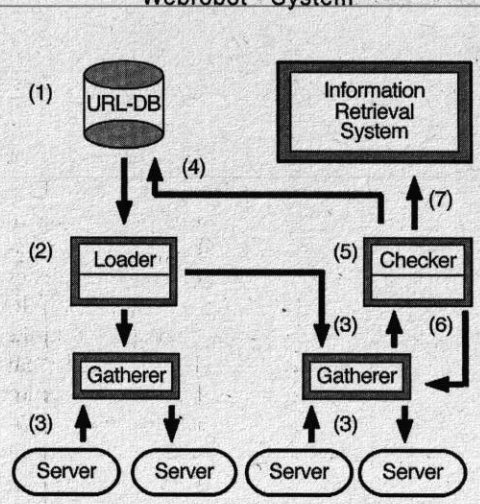

Abb. 3.1. Komponenten eines Webrobot-Systems

Links

Web Gathering Subsystem
- [www.npac.syr.edu/users/gcheng/homepage/thesis/node99.html]

AltaVista Search Engine 3.0, Software Product Description
- [http://solutions.altavista.com/docs/AVSE-3.0-SPD-2.4.pdf]

A System for Collecting and Analyzing Topic-Specific Web Information
- [www9.org/w9cdrom/293/293.html]

Slow Robot
- [http://docs.iplanet.com/docs/manuals/compass/301c/admin/filling.htm]

FAST Web Crawler-FAQs
- [www.fastsearch.com/support/crawler.asp]

CRAWLER
- [http://wwwbrauer.informatik.tu-muenchen.de/seminare/web/WS0001/vortrag02.html]

Harvest-NG System overview
- [http://webharvest.sourceforge.net/ng/develop/overview.shtml]

Digital Libraries
- [http://dbpubs.stanford.edu:8090/pub/2000-29]

Robots Text Files
- [www.global-positioning.com/robots_text_file/index.html]

3.1.2 Erfassung des WWW durch Webrobots

Die wesentliche Aufgabe der Gatherer ist es, den vorhandenen Datenbestand einer Suchmaschine zu aktualisieren und den Bestand mit neuen Dokumenten fortlaufend zu erweitern.

Zielsetzung der Suchmaschinen hinlänglich bereits erfasster Dokumente ist, Veränderungen an den Dokumenten möglichst umgehend zu erkennen und im eigenen Datenbestand entsprechend zu aktualisieren. Aktualisieren bedeutet in diesem Zusammenhang auch zu berücksichtigen, ob ein Dokument noch existiert. Die Aktualität der gespeicherten Daten ist bekanntermaßen ein wesentliches Qualitätskriterium von Suchmaschinen. Wie jeder Anwender aus eigener Erfahrung kennt, ist nichts quälender als Suchergebnisse, die eine Vielzahl von Links enthalten zu Dokumenten deren Inhalt sich entweder schon lange geändert hat oder Verweise zu Ressourcen, die nicht mehr existieren. Suchmaschinen versuchen einmal erfasste Dokumente so häufig wie möglich zu besuchen (auch crawlen genannt), um Veränderungen möglichst sofort zu erkennen. Die Realität zeigt jedoch, dass es den Suchmaschinen aufgrund beschränkter Systemressourcen, der Änderungsgeschwindigkeit und dem anhaltenden Wachstum im Internet nur sehr schwer möglich ist, Veränderungen zeitnah zu erkennen und ihrem Bestand entsprechend zu aktualisieren.

Besondere Beachtung finden URL's bei den Suchmaschinen. Spezielle Filter extrahieren alle URL's, die sich in den Dokumenten befinden. Versteht man das Hypermedia als ein Netzwerk von weltweit gegenseitig mit Hyperlinks verknüpften Ressourcen wird deutlich, dass die Suchmaschinen gezielt möglichst alle URL's erfassen und weiterverfolgen. Über diese Systematik ist es ihnen möglich neue Dokumente zu identifizieren und theoretisch sogar das gesamte World Wide Web vollständig zu erfassen.

Zur Initiierung eines Crawl-Prozesses (Überprüfung gespeicherter URL's) erhält ein Gatherer über den Loader eine von der URL-Datenbank erzeugte Liste aller URL's, die zu besuchen sind. Diese Liste wird von der URL-Datenbank an Hand bestimmter Kriterien, wie z.B. Datum des letzten Besuchs, Änderungshäufigkeit des Dokuments oder Zugehörigkeit zu einem bestimmten Netzwerkbereich sowie Art des zu stellenden HTTP-Requests, erwartetem Dokumententyp oder weiteren individuellen Auswahlkriterien erstellt.

Der Gatherer startet mit den URL's als Adressen HTTP-Requests an die ausgewählten Server und fordert entweder Informationen über die betreffende Ressource (konditionaler Request) oder fragt gleich die vollständige Ressource zur Übermittlung an. Der Server liefert die gewünschte Information bzw. überträgt die angeforderte Datei mit allen dazugehörigen Dateien an den Gatherer. Umgangssprachlich wird dieser Vorgang auch oft als „*... der Server wird vom Robot besucht ...*" bezeichnet.

Die Requests erfolgen als HTTP-Befehle wie z.B. dem GET-Befehl:

GET http://www.noesis.de/emarketing/online_marketing.html HTTP/1.1

Mit dem HTTP-Request überträgt der Gatherer im Request Header seinen individuellen User Agent (Client-Kennung), wodurch es dem Host bzw. einer Protokollisierungssoftware möglich ist, den Gatherer zu identifizieren. So lautet beispielsweise die Client-Kennung von Google *Googlebot*, FAST identifiziert sich mit *FastCrawler* und Fireball übermittelt *KIT-Fireball* als User Agent. Eine sehr gute Übersicht über Crawler bietet weiterführend *The Web Robots Pages*.

Erfolgt ein Verbindungsaufbau und der Request fehlerfrei, antwortet der angefragte WWW-Server entsprechend dem jeweilig gestellten HTTP-Befehl bzw. Methode:

GET-Methode

Es wird das vollständige Dokument inkl. aller Protokoll-Header-Informationen übermittelt.

Konditionalen GET-Methode

Das angefragte Dokument wird nur dann vollständig übermittelt, wenn die Bedingung erfüllt ist. Ansonsten liefert der Server nur Header-Informationen.

Der Übertragung der angeforderten Ressource werden Protokoll-Header-Informationen vorangestellt. Sie liefern für die Indexierung interessante Informationen, die bei der Relevanzbewertung berücksichtigt werden können. Die Header-Informationen können weiter dazu eingesetzt werden, Webressourcen nach bestimmten Kriterien zu analysieren, zu bewerten und zu kategorisieren. Die Header-Informationen werden deshalb zur Bearbeitung an die URL-Datenbank sowie an das Retrievalsystem weitergegeben und konkret bei der Indexierung berücksichtigt.

Weiter liefern die HTTP-Header-Informationen wichtige Angaben über die Möglichkeit des Zugriffs auf Ressourcen sowie die Verfügbarkeit bzw. den Status (sogenannte Status Code Informationen) des betreffenden Server. Diese Informationen führen, in Abhängigkeit ihrer Bedeutung und systemspezifischer Einstellungen, bei den Robotsystemen zu entsprechend differenzierten Maßnahmen wie z.B. der Löschung eines URL oder zu einem späteren Wiederbesuch.

In Hinblick auf ihre Wirkung bei der Aufnahme in den Index einer Suchmaschine sind die wichtigsten Status Codes mit ihrer Bedeutung kurz zusammengefasst:

Status Code 200 OK

Das Dokument befindet sich unter dem angefragten URL und der Request konnte vom Server entsprechend der Request-Methode erfüllt werden. Die übertragenen Daten werden vom Robotsystem verarbeitet.

Status Code 301 Moved Permanently

Das Dokument befindet sich nicht mehr unter dem angefragten URL; der Server hat den Request an den aktuellen URL weitergeleitet. Der alte URL kann vom Robotsystem durch den neuen URL ausgetauscht oder vollständig gelöscht werden.

Status Code 302 Moved Temporarily

Das Dokument befindet sich zur Zeit nicht mehr unter dem angefragten URL. Der Server hat den Request temporär an einen anderen URL weitergeleitet. Der alte URL kann vom Robotsystem durch den neuen URL ersetzt oder vollständig gelöscht werden.

Status Code 304 Not Modified

Bei dem konditionalen GET-Request *if-modified-since* bzw. *if-not-modified-since* der eine Dokumentenübertragung nur dann ausführt, sofern ein Dokument geändert wurde, liefert der Server mit dem Response-Header den 304 Code. Die Suchmaschine kann die darin enthaltenen Information zur Bestimmung der Änderungshäufigkeit von Dokumenten verwenden.

Status Code 401 Unauthorized

Eine Übertragung eines Dokuments ist nur nach vorherigem Autorisierungsverfahren möglich. Da Robots dies nicht ausführen können, kann ein solches Dokument auch nicht in den Datenbestand aufgenommen werden.

Status Code 404 Not Found

Die angefragte Ressource befindet sich unter der verwendeten URL nicht mehr auf dem Server. Die Datei wird aus dem Datenbestand der Suchmaschine gelöscht.

Status Code 414 Request-URL Too Long

Der Server kann den angefragten URL-Request nicht beantworten, da er zu lang ist. Der URL wird aus der URL-Datenbank der Suchmaschine gelöscht.

Status Code 500 Internal Server Error

Der Server ist im Moment des Requests technisch nicht in der Lage die Anfrage (z.B. wegen Überlastung) zu beantworten. Der betreffende URL wird in eine Warteschleife eingereiht und nach Ablauf einer bestimmten Zeit nochmals besucht. Dieser Vorgang kann sich mehrmals wiederholen, bis ein gesetzter Wert erreicht ist und eine Löschungsmeldung des URL an das Robot-System erfolgt.

Eine vollständige Übersicht und Beschreibung zum HTTP-Protokoll und den Status Codes findet man bei der *Network Working Group*.

Links

Network Working Group
- [www.w3.org/Protocols/rfc2068/rfc2068]

The Web Robots Pages
- [www.robotstxt.org/wc/robots.html]

Search engine robots that visit your web site
- [www.jafsoft.com/searchengines/webbots.html]

3.1.3 Loader und URL-Datenbank

Die beiden Komponenten *Loader* und *URL-Datenbank* sind grundsätzlich zwei eigenständige Systemkomponenten, die jedoch in sehr engem, funktionalem Zusammenhang stehen. Es ist deshalb erforderlich, beide Komponenten in ihrem wechselseitigen Zusammenwirken zu sehen.

Die Funktion des Loader erstreckt sich auf das Managen von verteilt arbeitenden Suchrobots. Dies beinhaltet das Übergeben von Request-Aufträgen an die Gatherer, das Überwachen der Ausführung und damit verbunden, die Optimierung von Systemressourcen durch eine Analyse der Auslastung der einzelnen Gatherer.

Damit Suchrobots über Informationen verfügen,welche URL überprüft werden soll, erhalten sie vom Loader URL-Listen, mit dem Auftrag an die hierin aufgeführten URL's einen HTTP-Request zu richten. Mit der Übergabe der URL-Liste erfolgt gleichzeitig eine Definition der Art des auszuführenden HTTP-Requests. Zulässige Requests sind der GET-, der konditionale GET- und der HEAD-Request. Die konkrete Art des HTTP-Request richtet sich danach, ob ein Dokument auf seine Existenz hin überprüft, eine Veränderung an einem Dokument erkannt oder ein nicht indexiertes Dokument vollständig geladen werden soll.

Neben der Art des auszuführenden Request wird in der URL-Liste auch der erlaubte Dokumententyp definiert, der unter einem URL erwartet wird. Das HTTP-Protokoll ermöglicht über die Content-Type-Definition den angeforderten Ressourcentyp zu bestimmen. Nur wenn ein Dokument der Content-Type-Definition entspricht, wird es vom Server übertragen. Mittels dieser Methode unterbinden die Systeme u.a., dass Dokumente übertragen werden die nicht vom System verarbeitet werden können oder sollen.

Die Bestimmung,welcher URL zu welchem Zeitpunkt besucht werden soll, erfolgt jedoch nicht durch den Loader, sondern durch die URL-Datenbank. Die URL-Datenbank ist in der Regel eine relationale Datenbank, die die Daten der vom System bereits erfassten URL's nach Kriterien speichert und somit eine differenzierte Wiederbesuchshäufigkeit (Crawl-Perioden) ermöglicht. Sie kann Teil des Robotsystems oder Teil des Information Retrieval-Systems sein.

Die wesentliche Funktion der URL-Datenbank ist die Speicherung und das Verwalten aller URL's. In der URL-Datenbank werden alle vom System übergebenen URL's nach Kategorien gespeichert. Es kann grundsätzlich nach zwei Hauptkategorien unterschieden werden.

(1) Eine Kategorie bilden URL's deren Dokumente bereits durch das Retrieval-System erfasst und indexiert wurden. Diese sind periodisch auf ihre Existenz oder eine Veränderung hin zu überprüfen.

(2) Die andere Kategorie stellen neue URL's dar, die vom System z.B. durch URL-Extraktion aus Dokumenten erfasst wurden. Sie bedürfen einer differenzierten Bearbeitung, da noch keine Informationen über die jeweiligen Ressourcen vorliegen. Die entsprechenden Dokumente müssen erst aus dem WWW geladen und vom System analysiert werden. Durchlaufen sie alle Filter ohne Probleme werden sie in den Index der Suchmaschine aufgenommen.

Eine wichtige Anforderung an URL-Datenbanken ist die Umsetzung einer differenzierten Crawl-Strategie. Da nicht alle Dokumente einer gleichen Änderungshäufigkeit unterliegen, ist es Verschwendung von Systemressourcen, alle erfassten Dokumente in der gleichen Häufigkeit auf Veränderungen hin zu überprüfen. Die URL's werden hierzu in verschiedene Kategorien unterteilt, deren Wiederbesuchrhythmus auf Basis von einem oder mehreren Kriterien beruht. URL's können dabei gleichzeitig auch mehreren Kategorien angehören. URL-Cluster können z.B. an Hand

- der Änderungshäufigkeit eines Dokuments,
- der Tiefe eines Dokuments im Verzeichnis,
- seiner Netzwerkadresse,
- seiner IP-Adresse,
- seiner Wichtigkeit für das System (z.B. als Linkliste),
- der Fehlerhäufigkeit des Hostrechners,
- der Art der Programmierung (statische oder dynamisches HTML Seite),
- des Dokumententyps (z.B. HTML, PDF, RTF, Word, etc.),

gebildet werden.

Zur Umsetzung von Crawl-Clustern werden in der URL-Datenbank erforderliche Zusatzinformationen mit abgespeichert:

- vollständiger URL,
- Hostname und IP-Adresse des Host-Servers,
- Dokumenten-Mime-Type,
- Dokumentenerstellungs- und änderungsdatum,
- Datum des letzten Besuchs,
- Errechneter Wert der Änderungsfrequenz,
- Informationen aus der Robots.txt-Datei,
- Server Status Informationen (in Prozess / Prozess ausgeführt / Fehlermeldung).

Neben der Optimierung der Systemressourcen eines Webrobot-Systems kann die Zuordnung einer Ressource zu einer bestimmten Kategorie sowohl positive als auch negative Auswirkungen auf die Bewertung eines Dokuments haben. Verfolgt eine Suchmaschine beispielsweise die Strategie, möglichst immer aktuelle bzw. neue Informationen bevorzugt anzubieten, erleiden Dokumente die ein älteres Erstellungs- oder Änderungsdatum aufweisen, eine Verschlechterung in ihrer Bewertung und somit ihrer Rangposition.

Die Organisation von URL's nach IP-Adressen kann unterschiedliche Zielsetzungen verfolgen. Sie wird u.a. eingesetzt um auf diesem Wege alle URL's eines bestimmten Hostrechners aus dem Bestand zu löschen, wenn ein Verstoß gegen die Nutzungsordnung erkannt wird. So weist NorthernLight ausdrücklich darauf hin, dass es nicht nur einzelne URL's aus seinem Verzeichnis löscht, wenn es pornographische Inhalte auf den Seiten entdeckt, sondern unverzüglich alle URL's die die gleiche Server IP-Adresse besitzen.

Eine andere Motivation URL's nach IP-Adressen zu verwalten kann die Absicht sein, WWW-Server nicht URL-weise zu besuchen, sondern immer alle URL's einer betreffenden IP-Adresse zum gleichen Zeitpunkt aufzusuchen. Diese Systematik wird auch mittels Netzwerk-Clustern realisiert, in dem immer alle URL's besucht werden, die einer bestimmten Netzwerkadresse oder IP-Adresse zugeordnet sind.

Links

A Prototype WWW Search System
- [www.npac.syr.edu/users/gcheng/homepage/thesis/node99.html]

CEWES MSRC Web-Linked Database Projects
- [www.wes.hpc.mil/pet/tech_reports/ reports/pdf/tr_9841.pdf]

A Dynamic Warehouse for the XML Data of the Web
- [www-sop.inria.fr/orion/TAIWAN/fichierspresentation/file21cobena.ppt]

Crawling Important Sites on the Web
- [http://bibnum.bnf.fr/ecdl/2002/INRIA/INRIA.pdf]

NothernLight
- [www.northernlight.com]

World Wide Web Robots, Wanderers, and Spiders
- [www.csa.iisc.ernet.in/Documentation/WebDoc/Robots/]

3.1.4 Der Checker

Der *Checker* ist für die Aufnahme eines Dokuments in den Datenbestand einer Suchmaschine aus der Betrachtung eines Content-Anbieters, die kritische Systemkomponente. Durch den Einsatz eines Checker soll vermieden werden, dass

keine unerwünschten Ressourcen an das Information Retrieval-System weitergegeben werden. Der Checker übernimmt die wichtige Funktion der Überprüfung aller vom Gatherer übergebenen Eingangsressourcen hinlänglich der vom System definierten Vorgaben. Aufgrund klarer Vorgaben welche Dateitypen in welcher Form von der Suchmaschine verarbeitet und gespeichert werden können, gibt der Checker nur diejenigen Dokumententypen an das Retrievalsystem weiter, die den Vorgaben entsprechen. Soll eine Ressource folglich in den Datenbestand aufgenommen werden, muss sie vollständig den Spezifikationen der jeweiligen Suchmaschine entsprechen.

Ein gutes Beispiel ist der Dokumententyp. Mit Ausnahme nur weniger Suchmaschinen wie z.B. Google oder Altavista, ist das einzig zulässige Textdateiformat das die Suchmaschinen im WWW im allgemeinen zulassen, das HTML-Format. Und dies, obwohl Information Retrieval-Systeme grundsätzlich nahezu alle gängigen Textdateiformate problemlos verarbeiten können. Versucht man ein Nicht-HTML-Textformat, wie zum Beispiel ein Word-Dokument zu indexieren, gelingt dies nicht. Der Checker erkennt das Dateiformat als nicht System konform und löscht mit dem Dokument auch den betreffenden URL aus der URL-Datenbank.

Neben der Kontrolle des Dateiformats ist es für die Suchmaschinen wichtig, alle in den Dokumenten gefundenen URL's auf ihre syntaktische Richtigkeit, ihre Existenz sowie auf die technische Verfügbarkeit des betreffenden Server hin zu überprüfen. Da die Verarbeitung eines URL verschiedene Prozesse, wie das Speichern in der Datenbank, das Initiieren eines Crawl-Prozesses sowie das Verarbeiten des betreffenden Dokuments nach sich zieht, bedeutet die ungeprüfte Speicherung eines inaktiven oder fehlerhaften URL, Verschwendung von Systemressourcen. Weiter bedeutet die Speicherung von fehlerhaften oder nicht mehr existenten URL's eine Verschlechterung der Suchergebnisse, bezogen auf ihre Qualität und Präzision.

Die Funktionsbreite eines Checker kann sich bei seinen Analysefunktionen rein auf das Überprüfen der Dokumente und URL's mittels einiger weniger Filteranwendungen beschränken. Es können jedoch auch sehr umfangreiche und detaillierte Filterketten implementiert sein, mit der Zielsetzung einer völligen Datennormalisierung, Dokumentenanalyse und -klassifikation. Im Regelfall erfolgt jedoch die Datennormalisation und -analyse durch das Retrievalsystem. Ein Checker besteht grundsätzlich aus einer Kette an Filtern, die sequenziell auf ein Dokument angewendet werden. Jeder einzelne Filter kann dabei ein Dokument bzw. den URL verändern oder löschen.

Nachfolgend werden drei wesentliche Filterprozesse erläutert, die von nahezu allen Suchmaschinen angewendet werden und deren erfolgreicher Durchlauf für ein Dokument im Allgemeinen Voraussetzung für die Aufnahme in den Datenbestand ist.

Dokumentenfilter

Zur Überwachung des Dokumententyps wird ein Dokumentenfilter eingesetzt, der zur Filterung der Eingangsdaten auf erlaubte Ressourcen dient. Die Identifikation

des Dokumententyps erfolgt durch die Mime-Type Übermittlung innerhalb des HTTP-Response-Header als Objektinformation. Entspricht ein Eingangsdokument nicht den Systemvorgaben wird es gelöscht und der dazugehörige URL aus der Datenbank entfernt.

Dublettenerkennung

Nach der Überprüfung auf zulässige Dokumententypen filtert der Checker die einzelnen Ressourcen auch darauf, ob sie bereits unter dem gleichen oder einem anderen Domain-Namen indexiert wurden. Dublettenerkennung ist für Suchmaschinen mit halbwegs aktueller Technologie kein Problem.

Ein HTML-Dokument kann über mehrere Domain-Namen auf dem gleichen oder einem anderen Server aufgerufen werden. Hierzu muss lediglich der WWW-Server für jede Domain einen Eintrag auf das gleiche Verzeichnis führen, bzw. eine Kopie eines Dokuments auf einem anderen Server gehostet werden. Ruft man nachfolgende URL's auf, wird immer das gleiche Dokument angezeigt.

- www.noesis.de/emarketing/content_management.html
- www.noesis-ecommerce.de/emarketing/content_management.html
- www.noesis-online-marketing.de/emarketing/content_management.html

Dubletten sind aus Sicht der Suchmaschinen nicht nur tatsächliche Kopien einer Datei, abgelegt unter anderem Dateinamen, sondern inhaltlich übereinstimmende oder nur geringfügig voneinander abweichende Seiten. Existiert ein Dokument identisch unter anderen Domain-Namen auf dem gleichen oder auf anderen Servern, ist es eine Dublette.

Zur eindeutigen Identifizierung von Dokumenten wird bei der Indexierung für jedes Dokument eine eindeutige Kontrollsumme errechnet und mit abgespeichert. In der Praxis wird für einen Datensatz eine Chiffriersumme nach einem Verfahren errechnet das sicherstellt, dass jede Informationseinheit eine eigene Chiffriersumme erzeugt. Unabhängig von der Länge einer Dateneinheit ist dabei jede Chiffriersumme 16 Bytes lang. Bei identischen Dokumenten ergibt sich eine identische Chiffriersumme. Der Filter kann durch einen Vergleich der Chiffriersummen sehr schnell eine Dublizitätserkennung ausführen und entsprechende Maßnahmen wie z.B. das Löschen der entdeckten Dublette vornehmen.

URL-Filter

URL's bilden die wichtigste Grundlage für die Erfassung des Internets und finden besondere Beachtung bei den Filterprozessen. Es wird folglich von den Webrobot-Systemen eine genaue Analyse der erhaltenen URL's hinlänglich ihrer Existenz, Syntax sowie anderer relevanter Kriterien vorgenommen. Die Filter werden auf alle dem System übergebenen URL's angewendet.

In einem ersten Schritt wird überprüft, ob die durch das HTTP-Protokoll definierte URL-Syntax eines gefundenen URL eingehalten wird. Zur Erkennung von

dynamisch generierten Dokumenten wird der URL-String weiter auf Sonderzeichen wie ?, &, %, =, untersucht. Befindet sich ein solches Zeichen im URL, ist das ein eindeutiger Hinweis auf dynamisch erzeugte HTML-Dokumente. Schließt eine Suchmaschine dynamisch erzeugte Dokumente von der Indexierung aus, wird der betreffende URL nicht gespeichert.

Eine Überprüfung des URL auf seine Existenz sowie die Erreichbarkeit eines Servers kann vor der Aufnahme in die Datenbank, als auch im Zuge des regulären Crawl-Prozess erfolgen. In beiden Fällen wird hierzu ein HTTP-Request gestartet und der Status Code des Response-Header ausgewertet. Wird der Status Code *404 file not found* geliefert, ist der URL nicht mehr existent und es kommt zur Löschung des betreffenden URL. Wird ein Status Code der Klasse 5xx zurückgeliefert, deutet das auf Probleme des Hostrechners hin. In solch einem Fall erfolgen im allgemeinen noch weitere Request-Versuche bevor der URL gelöscht wird.

Ein wichtiger Filterprozess stellt den Abgleich eines URL mit einer Black List dar. Mit Hilfe von Black Lists wird sichergestellt, dass Dokumente die gegen die Nutzungsordnung der Suchmaschinen oder nationale Gesetze verstoßen, nicht in den Datenbestand aufgenommen werden. Ein Eintrag in eine Black List ist permanent, d.h. ein URL bzw. ein Dokument ist dauerhaft gesperrt. Die Filterung des URL kann auch in Kombination mit der IP-Adresse erfolgen. Wird ein URL bei *diesem* Verfahren auf die Black List gesetzt, erfolgt gleichzeitig die Sperrung des gesamten IP-Adressbereich.

Über die URL-Datenbank besteht die Möglichkeit die maximale Anzahl von URL's je Domain oder Host auf eine Obergrenze zu beschränken. Hierzu wird in der Datenbank eine Obergrenze festgelegt und die Anzahl der URL's je Host oder Domain bei der Neuaufnahme überprüft. Ist die maximal zulässige Anzahl überschritten, erfolgt keine Aufnahme. Von dieser Maßnahme sind gelegentlich Content-Anbieter betroffen, die ihre Websites bei Massen-Providern unter einer Subdomain bzw. einer einzigen Domain hosten, die für Hunderte oder Tausende von Subverzeichnissen verwendet wird.

Ein Redirect-Filter prüft, ob sich ein Dokument tatsächlich unter dem angegebenen URL sowie auf dem betreffenden Server befindet oder ob eine Weiterleitung auf ein anderes Dokument erfolgt. Eine automatisierte Weiterleitung, auch Redirect genannt, kann z.B. mittels HTTP-Befehl im HTML-Dokument ausgeführt werden. Wird dann der ursprüngliche URL aufgerufen, erfolgt eine automatische Weiterleitung einer Anfrage nach Ablauf einer Zeitvorgabe auf einen anderen als den ursprünglichen URL. Erfolgt ein Redirect, wird dies jedoch im Response-Header als Status Code Information an die Suchmaschine kommuniziert und ist somit erkennbar.

Von einigen Suchmaschinen wird ein Redirect als Spam-Versuch interpretiert und es erfolgt die Löschung des betreffenden URL. Wird hingegen ein Redirect von einem System zugelassen, erfolgt im allgemeinen eine vollständige Überprüfung der betreffenden Ressource unter dem neuen URL.

Links

CEWES MSRC Web-Linked Database Projects
- [www.wes.hpc.mil/pet/tech_reports/ reports/pdf/tr_9841.pdf]

URL Filter
- [www.oasis-europe.org/docs/en/d0305/node33.html]

Frequently Asked Questions (and Answers) about Harvest
- [www.tnt.uni-hannover.de/print/plain/soft/info/harvest/FAQ.html]

3.2 Datenaufbereitung und Analyse

Eine Datenaufbereitung hat als erste Aufgabe Textdateien in ein einheitliches Datenformat umzuwandeln, das vom System effizient verarbeitet werden kann. Die Information Retrieval Systeme der Suchmaschinen analysieren die vom Webrobot-System übergebenen HTML-Dokumente nicht auf Basis ihrer Originaldateien sondern in konvertierten Form.

Die Dokumentenanalyse hat weiter zur Aufgabe, in einem anschließenden Prozess Textdokumente inhaltlich zu erschließen. Ziel ist es, Begriffe in Textdokumenten *aufzuspüren*, die als Schlüsselworte den Sinn bzw. das Thema das ein Dokument behandelt, wiederzugeben. Doch bevor dies möglich ist, müssen zuerst die Zeichenfolgen von Textdokumenten als *Worte im semantischen* Sinn identifiziert und einer bestimmten natürlichen Sprache zugeordnet werden.

Das physikalische Ergebnis der Dokumentenaufbereitung und Analyse dient zur Entwicklung des Datenbestandes der Suchmaschinen und stellt somit die Basis für die Verfahren der Relevanzbewertung dar.

3.2.1 Information Retrieval Systeme

Zum besseren Verständnis der erforderlichen Verfahren zum Aufbau von Datenstrukturen bei Suchmaschinen, soll vorab kurz dargestellt werden, wie Suchmaschinen ihren Datenbestand organisieren.

Suchmaschinen bestehen auf der Systematik von *Information Retrieval Systemen.* Dies sind spezielle Datenbanksysteme zur Verarbeitung von Textdokumenten. Sie werden seit Beginn des WWW bei der Informationssuche in wenig strukturierten Dateitypen, wie z.B. HTML-Dokumenten eingesetzt. Ziel eines Retrievalsystem ist es, Textdokumente so aufzubereiten, dass ein effizient durchsuchbarer Datenbestand entsteht, der Texte unter Berücksichtigung von Bewertungskriterien erfasst und eine Rangfolge der gefundenen Dokumente hinlänglich einer Suchanfrage ermöglicht.

Der Aufbau eines Datenbestands besteht aus verschiedenen Verfahren. Der sogenannte Indexierungsprozess lässt sich dabei im wesentlichen in drei Teilprozesse,

1. die Datennormalisierung,
2. die Dokumentenanalyse,
3. die Bildung von durchsuchbaren Datenstrukturen (auch Indexierung genannt),

unterteilen.

Der Unterschied zwischen Informationen Retrieval Systemen und Tabellen orientierten Datenbanksystemen, wie beispielsweise SQL-Datenbanken, lässt sich am besten durch einen Vergleich darstellen. Betrachtet werden hierzu die Form wie Daten vom System erfasst werden, in welchen Datenstrukturen sie abgespeichert und wie Suchanfragen beantwortet werden. Um zu verstehen, wie Daten bei Tabellen orientierten Datenbanksystemen erfasst und verarbeitet werden, betrachten wir zunächst deren Datenstrukturen.

Die Form der Datenhaltung erfolgt in Tabellen, die relational miteinander verknüpft sein können. Die Tabellen bestehen aus Kolumnen und Zeilen. Die Felder einer Zeile bilden immer einen Datensatz, auch Tupel genannt. Die einzelnen Kolumnen definieren den jeweiligen Feldtyp. In jeder Kolumne wird ein bestimmter Typ von Inhalt erwartet, weshalb auch der Feldtyp bei der Erstellung der jeweiligen Datenbanktabelle in Hinblick auf Zeichentyp und Zeichenlänge definiert werden kann. Der Tabelleninhalt wird entweder über ein Eingabemodul mittels manueller Eingabe oder durch den Import von Daten eingestellt. Eine einfache Tabelle einer Adressdatei kann folgendermaßen aussehen:

Tabelle 3.1. Darstellung einer Datenbanktabelle

Name	Vorname	Strasse	Nr.	PLZ	Ort
Meier	Peter	Pausenstr.	17	80639	München
Huber	Anton	Romanplatz	12	10225	Berlin
Schmidt	Anja	Zehrstrasse	31	60337	Frankfurt

Eine Tupel bzw. Datensatz besteht in diesem Fall aus mehreren Feldern und sieht im Falle des obigen Beispiels wie folgt aus:

Meier	Peter	Pausenstr.	17	80639	München

Der Aufbau der Datenbank mit Inhalten erfolgt entweder durch Direkteingabe über ein Eingabeinterface, durch Prozeß orientierte Einträge oder durch Datenimport. Wesentlich dabei ist, dass durch die Definition der jeweiligen Kolumne der Inhalt hinlänglich seiner Bedeutung definiert wird. So ist bei obiger Tabelle bestimmt, dass alle Inhalte des Feldes *Name* den Namen einer Person repräsentieren. Das System nimmt keine weitere Analyse mehr vor, ob die Zeichenfolge *Meier* auch tatsächlich ein sinnvolles Wort im semantischen Sinn ergibt oder nicht.

Die Suche von Datensätzen erfolgt mittels einer *Data Manipulation Language* wie z.B. SQL. Möchte man einen bestimmten Datensatz in einer Tabelle suchen, erfolgt ein Vergleich aller Inhalte der betreffenden Kolumne mit dem Suchstring. Wird der Name *Meier* in der Kolumne *Name* gesucht, werden hierzu alle Felder der Kolumne nach der Zeichenfolge *Meier* durchsucht und diejenigen Datensätze zurückgeliefert, die eine Zeichenfolge *Meier* beinhalten. Kommt der Name *Meier* mehrmals vor, werden alle Datensätze angezeigt, die die Zeichenfolge *Meier* beinhalten. Die Reihenfolge der Auflistung kann sich nach verschiedenen Kriterien wie z.B. dem Datum der Erstellung des Datensatzes richten. Eine Unterscheidung nach dem Aspekt der Relevanz erfolgt nicht; alle Suchergebnisse sind bezogen auf die Suchanfrage gleich relevant.

Bei Information Retrieval Systemen werden keine strukturierten Daten (z.B. in Tabellenform) zur Verarbeitung übergeben, sondern unstrukturierte Textdokumente aus dem Internet. Aus der Sicht eines Computersystems bestehen Textdokumente zunächst nur aus eine Abfolge von Zeichen. Es kann zunächst nicht erkennen, welche Zeichenfolge ein natürlichsprachiges Wort abbildet, wo es anfängt und aufhört. Während bei Tabellen orientierten Datenbanksystemen durch die Bestimmung der Kolumne definiert wird, dass alle sich hierin befindenden Zeichenfolgen z.B. einen Namen repräsentieren, muss von Information Retrieval Systemen erst durch einen speziellen Sprachfilter erkannt werden, ob es sich bei einer Zeichenfolge um ein Wort im semantischen Sinn handelt.

Die Erkennung von Worten ist für Information Retrieval Systeme Voraussetzung um einen durchsuchbaren Datenbestand aufbauen zu können. Anders als bei Tabellen orientierten Datenbanksystemen müssen sie bei der Analyse von Textdokumenten automatisiert bestimmen, welche Zeichenfolge innerhalb eines Dokuments ein Wort im semantischen Sinn darstellt. Erst nach der Identifikation einer Zeichenfolge als Wort kann ein Begriff so in den Datenbestand übernommen werden, dass bei einer Suche sowohl das betreffende Wort, als auch das Dokument in dem das Wort vorkommt, auffindbar ist.

Die von Information Retrieval Systemen überwiegend eingesetzte Datenstruktur ist ein *invertiertes Dateisystem,* das im nächsten Kapitel noch genauer beschrieben wird. Im wesentlichen beruht es auf einem *Index,* in dem alle Worte geführt werden, die in den erfassten Textdateien vorkommen. Ein Index kann man sich sehr einfach als alphabetisch sortierte Liste aller erkannten Worte vorstellen. Jedes Wort im Index verfügt über einen Verweis zu einer *invertierten Datei.* In dieser invertierten Datei befinden sich wiederum Verweise zu all denjenigen Dokumenten, die das betreffende Wort beinhalten. Für jedes im Index geführte Wort existiert jeweils eine invertierte Datei. Da ein Textdokument im allgemeinen sehr viele Worte beinhaltet, wird es in allen entsprechenden invertierten Dateien mit seiner Dokumenten-Identnummer (DocID) als Verweis geführt. Nachfolgende Grafik verdeutlicht die Struktur eines invertierten Dateisystems an Hand des im Index vorkommenden

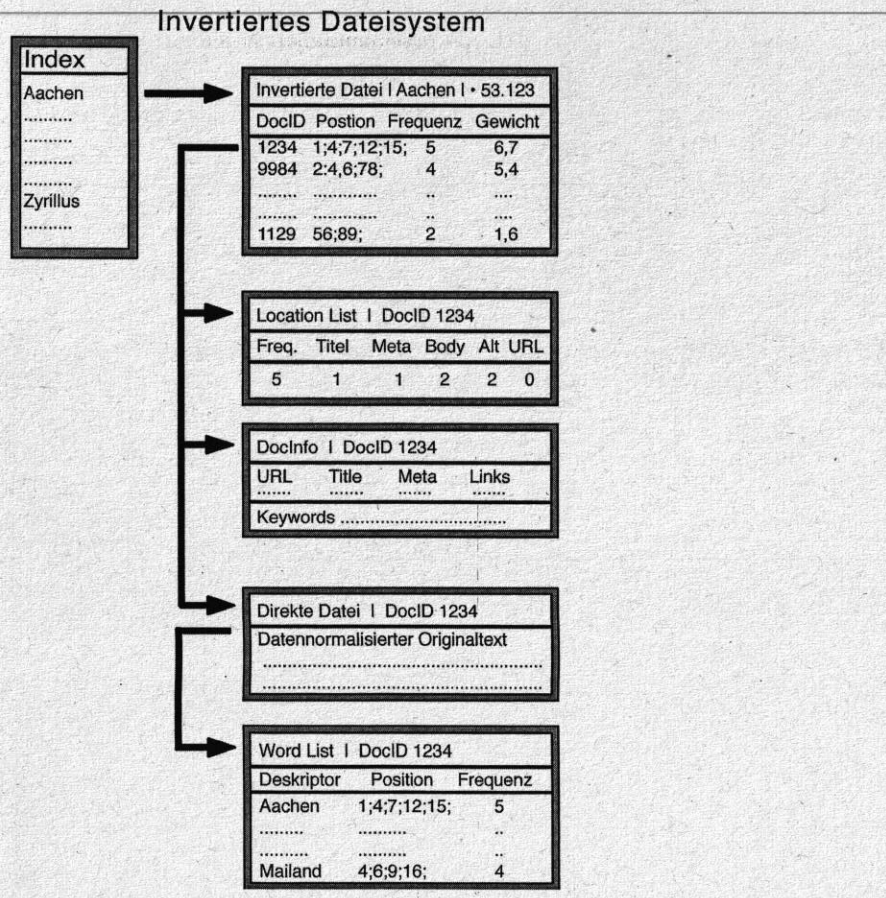

Abb. 3.2. Struktur eines invertierten Dateisystems – schematische Darstellung

Wortes „Aachen". Gezeigt wird die dazugehörige invertierte Datei sowie die Textdokumente, in denen das Wort „Aachen" im Dokument erscheint.

Bei einer Suche wird eine Suchanfrage an den Index gerichtet. Beinhaltet der Index das gesuchte Wort, werden über die betreffende invertierte Datei all diejenigen Dokumente angezeigt, die das gesuchte Wort beinhalten. Das Ergebnis der Suche stellt eine Liste aller gefunden Dokumente mit entsprechenden Verweisen zu den Dokumenten dar, die das Suchwort führen.

Wichtiges Unterscheidungskriterium zu den Suchergebnissen von Tabellen orientierten Datenbanksystemen ist, dass die Information Retrieval Systeme der Suchmaschinen *gewichtete Verfahren* zur Relevanzbestimmung einsetzen. Während SQL-

Datenbanken eine Suchergebnisliste beispielsweise nach Erstellungsdatum der Datensätze oder alphabetisch bzw. numerisch sortieren, setzen die Suchmaschinen einen Algorithmus ein, der sich an der *Relevanz* eines Dokuments zur Suchanfrage orientiert. Ein Dokument ist im Sinne einer Suchanfrage relevanter als ein anderes Dokument, wenn es *inhaltlich* der Suchanfrage eher entspricht, als ein anderes Dokument. Zur Ermittlung der Relevanz setzen die Information Retrieval Systeme verschiedene Verfahren ein, um Dokumente bezüglich ihrer Relevanz differenzieren zu können. Diese Verfahren beruhen, wie in diesem Buch noch ausgiebig dargestellt wird, auf dem Einsatz von verschiedenen Parametern zur Bestimmung von Dokumentengewichten sowie auf Retrieval-Funktionen zur Berechnung der Relevanz eines Dokuments hinlänglich einer Suchanfrage.

Die Aufgabe der Optimierung von Websites bedeutet also im Grunde nichts anderes, als all diejenigen Parameter genau zu kennen und richtig einzusetzen, die Suchmaschinen verwenden, um die Relevanz eines HTML-Dokuments bezüglich einer Suchanfrage zu berechnen. Möchte man eine möglichst gute Rangposition erzielen, muss durch den richtigen Einsatz der Gewichtungsparameter mathematisch ein möglichst hoher Relevanzgrad erzielt werden.

Links

Glossary for Information Retrieval
- [www.cs.jhu.edu/~weiss/glossary.html]

How Search Engines Rank Web Pages
- [www.searchenginewatch.com/webmasters/rank.html]

XIRQL: A Query Language for Information Retrieval in XML Documents
- [www.is.informatik.uni-duisburg.de/bib/fulltext/ir/Fuhr_Grossjohann:01.pdf]

Models in Information Retrieval
- [www.is.informatik.uni-duisburg.de/bib/fulltext/ir/Fuhr:00a.pdf]

Information Retrieval Invited Papers, Tutorials – Baeza-Yates
- [www.dcc.uchile.cl/~rbaeza/cv/invited.html]

Improving an Algorithm for Approximate String Matching
- [www.dcc.uchile.cl/~rbaeza/ftp/engin.ps.gz]

Literature about search services
- [www.lub.lu.se/desire/radar/lit-about-search-services.html]

Information Retrieval und das Web: Grundlagen & Problematik
- [www.inf.uni-konstanz.de/dbis/teaching/ss01/data-on-the-web/local/www_ir.pdf]

Information Retrieval Dokumentverarbeitung
- [www-ai.cs.uni-magdeburg.de/lehre/ws-00-01/DokVer/ab9.pdf]

Methoden und Modelle des Information Retrieval
- [http://page.inf.fu-berlin.de/~kuehn/diplom.ps]

Statistische Verarbeitung natürlicher Sprache
- [www.cl.uni-heidelberg.de/kurs/ss00/statling/]

Skriptum Information Retrieval
- [http://ls6-www.cs.uni-dortmund.de/ir/teaching/courses/ir]

Survey of the State of the Art in Human Language Technology
- [http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html]

Statistical Natural Language Processing
- [http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html]

3.2.2 Verfahren der Datenaufbereitung und Analyse im Überblick

Suchmaschinen sind so konzipiert, dass es ihnen technisch möglich ist, aus den Milliarden von Dokumenten des World Wide Webs einen Datenbestand zu erzeugen, der die Gesamtheit der vorhanden Dokumente so weit wie möglich abdeckt und innerhalb eines Systems zu einem durchsuchbaren Datenbestand agregiert. Wie schon im Kapitel 3.1.1 dargestellt, erfolgt das Auffinden und Erfassen von HTML-Dokumenten durch ein Webrobot-System, das an Hand von URL-Listen Dokumente im Internet aufsucht. Neue Dokumente werden durch das Weiterverfolgen von bisher nicht bekannten URL's identifiziert. Bisher unbekannte URL's können mittels Analyse der HTML-Dokumente gefunden oder durch aktive Anmeldung einer URL an das System übergeben werden.

Soll ein Datenbestand erzeugt werden der es ermöglicht Suchanfragen über alle aus dem WWW erhaltenen und gespeicherten Dateien zu stellen, setzt dies eine besondere Form der Datenaufbereitung und Datenanalyse sowie den Entwurf geeigneter Datenstrukturen voraus. Betrachtet man ein Textdokument so erkennt man unschwer, dass es aus der Sicht eines Computersystems zunächst nur aus einer Vielzahl an unterschiedlichen Zeichen besteht, die keinen Aufschluss über den Inhalt eines Textdokuments zulassen. Wesentliche Aufgabe der Suchmaschinen ist jedoch zu erkennen, welches Thema ein Dokument behandelt und in welcher natürlichen Sprache es verfasst ist. Denn nur durch eine inhaltliche Interpretation der Dokumente kann eine Suchanfrage im Sinne eines präzisen Ergebnisses beantwortet werden.

Diese Überlegungen führen dazu ein Verfahren einzusetzen, das die Zeichenketten zu Worten verbindet und eine Erkennung der natürlichen Sprache ermöglicht. Desweiteren ist eine Methodik erforderlich, die aufgrund einer Analyse aller Worte genau diejenigen bestimmt, die den Inhalt eines Dokuments repräsentieren. Oder anders ausgedrückt, es müssen die Worte ausgeschlossen werden, die

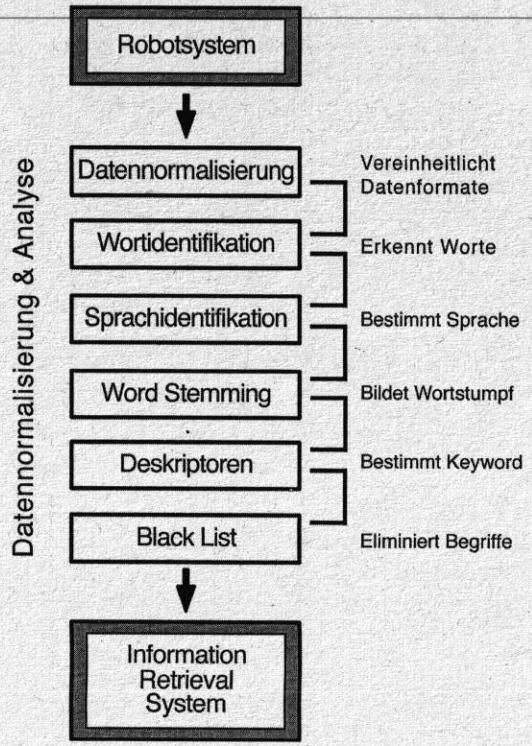

Abb. 3.3. Prozesse der Datennormalisierung und Dokumentenanalyse

keine inhaltliche Repräsentanz ermöglichen, aber gleichzeitig die Worte herausgefiltert werden, die den Sinn eines Textdokuments wiedergeben.

Im Gegensatz zu kontrollierten Systemen in denen Dokumententypen vordefiniert, Dokumente einheitlich strukturiert und die Qualität und Programmiersprache genormt sind, besteht das Web aus einer Unsumme an heterogenen Dokumenten, die zum Teil in schlechter Programmierqualität erstellt sind und darüber hinaus oftmals unterschiedliche Programmiererweiterungen oder Multimediaelemente beinhalten können. Damit ein System Daten effizient verarbeiten kann, müssen jedoch alle Daten ein einheitliches Datenformat besitzen. Aufgabe der Datenaufbereitung und Dokumentenanalyse ist es, aus dem Internet erfasste Textdokumente so aufzubereiten, dass daraus ein effizient durchsuchbarer Datenbestand gebildet werden kann.

Die vom Robotsystem übergebenen Ressourcen durchlaufen hierfür verschiedene Filter mit der Zielsetzung, die Dokumente in ein einheitliches Datenformat umzuwandeln, Worte im semantischen Sinn zu identifizieren und Keywords

(Schlüsselwörter) zu bestimmen, die einen Text inhaltlich wiedergeben. Die von den einzelnen Suchmaschinen eingesetzten Filterprozesse sind in Umfang und Ausprägung grundsätzlich systemspezifisch und können in Teilen sowohl vom Robotsystem als auch vom Retrievalsystem der Suchmaschine ausgeführt werden. Die gebräuchlichsten Verfahren sind dabei

- die Datennormalisierung,
- die Wortidentifikation,
- die Sprachidentifikation
- die automatisierte Keyword-Gewinnung.

Oben dargestellte Grafik bildet die allgemein üblichen Filterprozesse ab.

Links

Automatic Text Analysis
- [www.dcs.gla.ac.uk/~iain/keith/data/pages/14.htm]

Document Processor
- [www.infotoday.com/searcher/may01/liddy.htm]

Automatic Text Analysis
- [www.dcs.gla.ac.uk/Keith/pdf/Chapter2.pdf]

Solving The Word Mismatch Problem Through Automatic Text Analysis
- [http://citeseer.nj.nec.com/xu97solving.html]

Information Retrieval-Textanalyse
- [wwwiti.cs.uni-magdeburg.de/~sattler/lectures/agenten.ps]

Introduction in Information Retrieval
- [www.dcs.gla.ac.uk/Keith/Chapter.1/Ch.1.html]

3.2.3 Datennormalisierung

Der erste Prozess der auf die vom Webrobot-System erhaltenen Dokumente angewendet wird, ist die Transformation bzw. Konvertierung aller Dateien in ein einheitliches Datenformat, das vom System effizient verarbeitet und gespeichert werden kann. Kern des Normalisierungsprozess stellt die Umwandlung unterschiedlicher Datenformate und Datenqualitäten in ein System spezifisches Standardformat dar und beinhaltet das Entfernen von Programmiercode, sowie eine Restrukturierung der Datei, die sich an den technischen Erfordernissen des Systems orientiert. Erweiternd kann die Datennormalisierung auch eine Datenkompression der konvertierten Dateien zum Zwecke der Einsparung von Plattenspeicher beinhalten.

Angewendet auf HTML-Textdateien bedeutet eine *Datennormalisierung*, dass der gesamte HTML-Programmiercode entfernt wird. Bei nicht sauber erstellten HTML-Dokumenten kann es hier insofern zu Problemen kommen, dass der Filter ganz oder in Teilbereichen nicht erkennt, wo der Programmiercode endet und wo Text beginnt. Im Zuge der Separation des natürlichsprachigen Textes vom Programmiercode werden bei modernen Systemen auch Programmiererweiterungen wie z.B. JavaScript entfernt. Der Einsatz von Programmiererweiterungen führt jedoch nach wie vor bei verschiedenen Systemen zu dem Problem, dass er nicht eindeutig erkannt wird. Die Folge für das Dokument kann eine mangelhafte Berücksichtigung durch die Suchmaschine nach sich ziehen.

Wichtig für die Weiterverarbeitung ist die Erkennung der Dokumentenstruktur. HTML-Dokumente gelten allgemein hin als schwach strukturiert. Über die beiden Tags HEAD und BODY wird jedoch zumindest eine grobe Grundstruktur festgelegt, die es ermöglicht Worte im Dokumentenkopf bzw. im Dokumentenkörper differenziert zu berücksichtigen. Innerhalb des Dokumentenkopfs kann noch eine Unterscheidung vorgenommen werden, ob ein Wort im Dokumententitel oder innerhalb eines der Meta-Tags vorkommt. Eine mögliche Strukturierung des Dokumentenkörpers erfolgt in HTML u.a. durch die <H1> bis <H6>-Tags. Diese Tags dienen zur Bestimmung von Überschriften und leiten Absätze ein.

Die durch das HTML vorgegebenen Möglichkeiten der Strukturierung eines Dokuments werden bei der Datennormalisierung durch Erkennung der betreffenden Tags erfasst und in der konvertierten Datei entsprechend berücksichtigt. Gleiches gilt für die Darstellung von Sonderzeichen. Sprachspezifische Sonderzeichen wie zum Beispiel die deutschen Umlaute, werden von der für HTML üblichen Abbildungsmethode in eine systemspezifische Darstellungsform überführt.

Den umgewandelten Dokumenten wird im Zuge der Datennormalisierung eine eindeutige Dokumentennummer (die sogenannte DocID) zugeordnet, unter der das Dokument sowie sein URL fortan identifizierbar sind.

Links

Document Processor
- [www.infotoday.com/searcher/may01/liddy.htm]

Chancen und Grenzen der maschinellen Indexierung
- [www.agi-imc.de/.../$FILE/Chancen%20und%20Grenzen%20der%20 maschinellen%20Indexierung.pdf]

3.2.4 Wortidentifikation

Zum Aufbau einer durchsuchbaren Datenstruktur durch Retrieval-Systeme sind Worte im semantischen Sinn erforderlich, die ein Textdokument inhaltlich repräsentieren. Nur wenn Worte im semantischen Sinn identifizierbar sind, können diese indexiert und danach gesucht werden. Die Wortidentifikation ist ein Konvertierungsprozess, der eine Vielzahl von Zeichen innerhalb eines Dokuments in eine Menge an lexikalisch sinnvollen Worten umwandelt. Ein Problem in diesem Zusammenhang stellt wie bereits erwähnt, das Erkennen von Worten im Sinne einer natürlichen Sprache dar. D.h. ein Filter zur Wortidentifikation muss in der Lage sein, Bitfolgen als Worte im semantischen Sinne identifizieren zu können sowie Zeichen und Zahlen hiervon zu unterscheiden. Eine Entfernung des Programmiercodes bei HTML-Dokumenten erfolgt bereits im Zuge der Datennormalisierung.

Auf den ersten Blick mag die Wortidentifikation lediglich auf dem Erkennen von Leerzeichen als Wortseparatoren hinaus laufen. Realisiert man jedoch zur Worterkennung ausschließlich nur diese Methodik, würden grundsätzlich alle Zeichen und Buchstaben zwischen zwei Leerzeichen ein Wort im semantischen Sinn definieren. Betrachtet man hingegen einen Text genauer, stellt man fest, dass ein solches Verfahren nur ungenaue Ergebnisse liefern kann. Erschwerend wirken bei einer exakten Wortidentifikation die Berücksichtigung von Zahlen, Bindestrichen, Satzzeichen sowie die Groß- und Kleinschreibung von Worten, die ein verfeinertes Verfahren der Wortidentifikation erfordern.

Ein differenzierter Prozess der Wortfindung bei Hypermedia-Textdokumenten basiert auf der Erkennung von wiederkehrenden binären Assoziationen von Inhaltsworten. Somit wird zur Identifikation von lexikalisch sinnvollen Begriffen ein mehrstufiger Worterkennungsfilter auf das betreffende Dokument angewendet. Der Filter erkennt Worte durch die Klassifizierung von Symbolen in die drei Klassen:

- gültige Symbole zur Bildung eines Wortes,
- Symbole zur Trennung von Worten,
- besondere Prozesssymbole.

Erstere werden durch das jeweilige Alphabet dargestellt und zur weiteren Verarbeitung des hierdurch gebildeten Wortes an den nachfolgenden Filter weitergegeben. Durch die Zugrundelegung eines Wörterbuchs kann eine Bedeutungsidentifikation und somit Wortidentifikation im lexikalischen Sinn erfolgen. Es ist jedoch eine Methode zur Erkennung von Worten erforderlich, die nur in Großbuchstaben geschrieben sind. Eine gesonderte Definition bedingt auch die Erkennung von reinen Zahlen sowie von Worten, die eine Buchstaben-Zahlen Kombination wie z.B. 1994AD (1994 anno domini) beinhalten. Sowohl reine Zahlen als auch Buchstaben-Zahlenkombinationen müssen dabei als semantisch sinnvolle Zeichenfolgen identifizierbar sein.

Symbole zur Trennung von Worten dienen dem System zu erkennen, wann ein Wort beginnt und wann es endet. Diese Symbole sind z.B. Leerzeichen, Bindestriche, Kommas, etc. und sind im ASCII-Zeichensatz definiert. Die exakte Bedeutung ob ein solches Zeichen ein Symbol zu Worttrennung darstellt oder ob es Teil eines Wortes ist, hängt von der betreffenden Sprache ab und muss sprachspezifisch entschieden werden.

Ein einfaches Beispiel verdeutlicht dies. Während im Englischen ein Leerzeichen zwischen zwei Worten nicht immer zwei eigenständige Worte im semantischen Sinn definiert, sondern oftmals auch ein lexikalisch zusammen gehörender Begriff durch Leerzeichen getrennt sein kann (z.B. information retrieval system), trennen in der deutschen Sprache Leerzeichen allgemein lexikalisch eigenständige Worte.

Als Ergebnis des Wortidentifikationsprozesses stellt der Filter eine Liste an gefundenen Begriffen im lexikalischen Sinn bereit, die zur Generierung von geeigneten Schlüsselwörtern an den Keyword-Relevanzfilter bzw. vorab, an das Word Stemming Modul übergeben wird. Sofern dem Keyword-Relevanzfilter noch ein Stoppwortfilter vorgeschaltet ist, durchläuft die hier erzeugte Wortliste vorab einen Prozess der dazu führt, alle Worte aus der Liste zu eliminieren, die ohne inhaltlich repräsentative Bedeutung für ein Dokument sind.

3.2.5 Sprachidentifikation

Über die Robotsysteme erfassen Suchmaschinen mittels Link-Verfolgung alle Textdokumente die vom System zugelassen sind. Über das Internet sind bei genauer Betrachtung Dokumente in allen natürlichen Sprachen erreichbar. Da die Webrobots nicht in einem abgeschlossenen System Hyperlinks weiterverfolgen, sondern alle Verweise auf Ressourcen im gesamten WWW aufsuchen, ist es sehr wahrscheinlich, dass Verweise zu Dokumenten in unterschiedlichen Sprachen führen.

Würden die Suchmaschinen alle Textdokumente sprachlich undifferenziert erfassen und verarbeiten, führt dies zu einem unstrukturierten Datenbestand und hätte eine erhebliche Verschlechterung der Suchergebnisse zur Folge. Die Information Retrieval Systeme der Suchmaschinen verarbeiten folglich Eingangsressourcen mit der Maßgabe, zwischen den jeweiligen Sprachen zu unterscheiden und ihren Index sprachorientiert zu verwalten. Die Folge ist, dass Dokumente die nicht den definierten Sprachen entsprechen auch nicht von den Suchmaschinen in ihren Datenbestand aufgenommen werden. Möchte folglich ein Content-Anbieter ein Dokument im deutschsprachigen Datenbestand bei Google oder Altavista indexieren lassen, muss die natürliche Sprache in der das Dokument verfasst ist, überwiegend Deutsch sein.

Die Separation der Datenbestände nach Sprachen führt zu einer erheblichen Verbesserung der Suchergebnisse. Durch die Trennung der Indexe in unterschiedliche Sprachen werden sowohl Mehrdeutigkeiten von Worten unterschiedlicher Sprachen ausgeschlossen, als auch die Möglichkeit eröffnet, Suchergebnisse

nur in einer bestimmten Sprache zu erhalten. Für eine spezielle sprachspezifische Suche bieten sowohl Google, Alltheweb und Lycos Expertensuchen an. Über die Definition der gewünschten Sprache wird dabei eine Suchanfrage ausschließlich an den betreffenden Index gerichtet.

Zur getrennten Erfassung von Dokumenten und dem Aufbau sprachspezifischer Datenbestände, werden bei der Verarbeitung der Textdokumente spezielle Sprachfilter eingesetzt. Die Aufgabe von Sprachfiltern ist es zu erkennen, ob ein Textdokument in einer bestimmten definierten Sprache, bzw. in welcher natürlichen Sprache es verfasst ist. Entspricht ein Dokument einer definierten natürlichen Sprache, wird es an das hierfür vorgesehene sprachspezifische Indexierungsmodul weitergeben. Wird die Sprache nicht eindeutig erkannt oder ist ein Dokument in einer nicht zugelassenen natürlichen Sprache erstellt, wird das Dokument und der dazugehörige URL vom System gelöscht.

Sprachfilter können über die technische Kompetenz der Erkennung von unterschiedlichen natürlichen Sprachen verfügen oder lediglich die Funktion besitzen zu erkennen, ob ein Dokument in der einzigen vom System erlaubten natürlichen Sprache verfasst ist. Die eingesetzte Software und deren Methoden zur Sprachidentifikation können sehr unterschiedlich sein. Eine reine Sprachbestimmung über die Auswertung des Meta-Tag LANGUAGE ist jedoch in keinem Fall ausreichend und wird als Angabe bei der Indexierung nicht beachtet.

Um die jeweiligen Sprachen möglichst exakt zu bestimmen, kann ein kombiniertes Verfahren verwendet werden, bei dem sowohl *statistische Methoden* als auch ergänzend ein *Wörterbuch* zum Einsatz kommt.

Werden *statistische Verfahren* der Spracherkennung angewendet, basieren diese oftmals auf der Theorie der *Hidden-Markov-Modelle*. Die Hidden-Markov-Modelle sind eine Klasse von statistischen Modellen, die auf der Theorie der Markovketten basieren und vor allem in der Sprachverarbeitung Anwendung finden. Im Aufgabenbereich der Worterkennung dienen Markov-Modelle speziell der Repräsentation zeitlicher Abfolgen artikulatorischer Gesten und eignen sich zur Wortmodellierung oder Worterkennung.

Bei diesem Verfahren wird die Zeichenfolge der zu klassifizierenden Texte erfasst und mit der für die jeweilige Sprache typischen Zeichenfolge verglichen. Ausgehend vom beobachteten Ähnlichkeitsgrad wird der Text dann z.B. als deutschsprachig oder fremdsprachig klassifiziert. Im Zusammenhang mit der Erkennung einer natürlichen Sprache ist die Darstellung der sprachspezifischen Sonderzeichen einer Sprache im HTML-Code relevant.

Ergänzend wird zur möglichst genauen Bestimmung einer Sprache, bei nicht eindeutiger Identifikation mittels statistischer Verfahren, ein Wörterbuch zur Verifikation zu Grunde gelegt. Eine Erweiterung um ein Wörterbuch kann insbesondere dann erforderlich werden, wenn im Text verstärkt Eigennamen, Lehnwörter oder fachspezifische Terminologien verwendet werden, die eine Bestimmung der Dokumentensprache beeinflussen. Zur Bestimmung der jeweiligen natürlichen Sprache

erfolgt in solch einem Fall ein Abgleich der im Textdokument auftretenden Worte mit einem Wörterbuch. Erst durch die Kombination des statischen Verfahrens, unter Zugrundelegung von Wörterbüchern, kann die eindeutige Erkennung einer Sprache auch im Zweifelsfall zuverlässig erfolgen.

Links

Statistische Sprachmodelle
- [www.coli.uni-sb.de/~thorsten/gk-workshop/node11.html]

Other Web Pages Related to Cross-Language Text Retrieval
- [www.ee.umd.edu/medlab/mlir/resources.html]

Cross-Language LSI
- [www.cs.duke.edu/~mlittman/courses/Archive/INLS379/xlang/xlang.html]

Multilingual and Monolingual Term Identification and Applications
- [www.cs.columbia.edu/~min/presentations/candidacy/]

Automatic Text Analysis
- [www.dcs.gla.ac.uk/Keith/pdf/Chapter2.pdf]

3.2.6 Word Stemming

Das *Word Stemming* ist eng mit der automatisierten Bestimmung von Keywords und dem Aufbau des Index verbunden. Word Stemming bedeutet *Bilden eines Wortstamms*, wobei mit dem Wort *stem* nicht nur die Grundform eines Wortes gemeint sein muss, sondern auch ein um den Suffix (Nachsilbe) und / oder Prefix (Vorsilbe) eines Wortes gekürzter Wortstumpf definiert wird. Ziel dieses Verfahrens ist es, die Anzahl von Dokumenten die bei einer Suchanfrage berücksichtigt werden zu erhöhen sowie die Anzahl an Worten im Index ohne wesentlichen Bedeutungsverlust zu reduzieren.

Die diesem Verfahren zu Grunde gelegte Überlegung ist, dass der Wortstamm eines Wortes grundsätzlich die Bedeutung eines Wortes repräsentiert und dass semantische Abwandlungen von Worten, wie z.B. durch Pluralbildung oder Deklinationen, keine Abweichungen vom eigentlichen Sinn eines Wortes bedeuten. Entsprechend diesem Prinzip hat das Wort *house* im Englischen für ein Retrievalsystem das Word Stemming einsetzt, die gleiche Bedeutung wie *houses*. Dokumente mit den Deskriptoren *house* bzw. *houses* werden bei Anwendung von Word Stemming beide im Index unter dem erzeugten Wortstamm *house* geführt.

Der Prozess des Word Stemming beinhaltet das Transformieren von Worten auf ihre semantische Grundform oder einen durch das System definierten Wortstumpf. Bei der am meisten eingesetzten Form der Wortstammbildung, der Suffix-Entfernung, werden Worte die als Substantive in der Pluralform im Dokument

vorkommen, durch Entfernung der Plural bildenden Nachsilbe in die Singularform transformiert. In der im Englischen leicht zu bildenden Pluralform wird dazu das die Pluralform bildende Suffix „s“ bzw. „es“ mittels einem speziellen Filter erkannt und entfernt, wodurch der Wortstamm erhalten bleibt. Bei der Prefix-Entfernung wird hingegen einem Wort nach bestimmten Regeln seine Vorsilbe entfernt und nur der um die Vorsilbe verkürzte Wortstamm indexiert.

Entsprechend der jeweiligen Word Stemming-Strategie der Indexierung verarbeitet auch der Query Prozessor Suchanfragen in der entsprechend transformierten Form. D.h. auf Suchworte wird der gleiche Stemming-Prozess angewendet wie bei der Indexierung. Es findet somit auch bei der Suchanfrage eine Übersetzung eines Suchwortes in einen Wortstumpf statt.

Es ist wichtig hervorzuheben, dass Stemming-Algorithmen immer individuell auf eine bestimmte Sprache ausgerichtet sind und sich nicht alle Sprachen eignen, ein Stemming anzuwenden. Während das Word Stemming in der englischen Sprache aufgrund seiner relativ einfachen Semantik und Grammatik durch das Entfernen von Suffixen und Präfixen relativ effiziente und präzise Suchergebnisse liefert, führen die bekannten Algorithmen in grammatikalisch anspruchsvolleren Sprachen zu eher mangelhaften Ergebnissen.

Die Mehrzahl aller Suchmaschinen im Internet wenden kein Word Stemming an, sondern indexieren im Volltextmodus jeden Begriff, der im Dokument als Wort im semantischen Sinn erkannt wird. Ob eine Suchmaschine Stemming-Algorithmen einsetzt, lässt sich einfach dadurch nachvollziehen, in dem eine Suchabfrage mit einem Begriff in der Singular- und in der Pluralform ausgeführt wird. Ist die Anzahl der Dokumente des Suchergebnisses in beiden Fällen gleich, wird Word Stemming eingesetzt. Die Kenntnis über den Einsatz von Word Stemming ist insbesondere in Hinblick auf die Bestimmung von repräsentativen Deskriptoren im HTML-Dokument sehr relevant. Unterscheidet eine Suchmaschine beispielsweise zwischen der Singularform und der Pluralform, muss ein Deskriptor in beiden Formen im Dokument vorkommen, damit ein Dokument Teil des Suchergebnisses beider Formen des Suchwortes wird.

Links

Conflation and Stemming
- [www.comp.lancs.ac.uk/computing/research/stemming/paice/article.htm]

Indexing in our model of IR
- [www.cs.tcd.ie/courses/baict/baim/jf-im/13Indexing.pdf]

Stemming Algorithmus
- [www.cis.uni-muenchen.de/people/Schulz/SeminarSoSe2001IR/FilzmayerMargetic/ referat.html]

Fortgeschrittene Algorithmen für Stemming
- [http://stinfwww.informatik.uni-leipzig.de/~kherrman/stemming.ps.gz]

Textverarbeitung
- [www.cis.uni-muenchen.de/people/Schulz/SeminarSoSe2001IR/Nagy/node3.html]

3.2.7 Deskriptorengewinnung

Von allen Problemen die beim Aufbau und Pflege des Datenbestandes eines Retrievalsystems zu lösen sind, ist die Bestimmung von Deskriptoren die schwierigste Aufgabe. Primäres Ziel der Analyse von Textdokumenten ist es, diejenigen Begriffe in einem Dokument zu identifizieren, die dazu geeignet sind ein Thema *inhaltlich* zu repräsentieren. D.h. aus der Gesamtmenge an Worten eines Textdokuments müssen genau die Worte vom System erkannt werden, über die es möglich ist, den thematischen Inhalt eines Dokuments darzustellen. Diese Teilmenge an Begriffen werden allgemein hin als *Schlüsselwörter* oder *Keywords*, aber auch als *Deskriptoren* bezeichnet.

Wie im Kapitel Information Retrieval Systeme (s. Kap. 3.2.1) schon erläutert, werden nicht die einzelnen Textdokumente bei einer Suchanfrage auf das Vorkommen eines Suchbegriffs untersucht. Sondern es werden die einzelnen Dokumente durch Worte repräsentiert die im Text vorkommen und die den Inhalt am genauesten wiedergeben. Für jedes Wort das im Index vorkommt existiert eine invertierte Datei, die unter dem betreffenden Begriff Verweise zu allen Dokumenten führt, die das Wort beinhalten. Je nach Systemvorgaben des IR-Systems muss ein Begriff in einer bestimmten Häufigkeit im Dokumententext vorkommen, um als Deskriptor geeignet zu sein.

Zum Aufbau einer Indexdatei ist es somit erforderlich Deskriptoren aus natürlichsprachigen Textdokumenten zu generieren, die ein Dokument *inhaltlich* repräsentieren. Diese Systematik ist immer dann einzusetzen, wenn bei einer Suchanfrage nicht jedes einzelne Dokument vollständig durchsucht werden soll, sondern Dokumente aufgrund von repräsentativen Begriffen gefunden und als relevant oder nicht relevant betrachtet werden sollen.

Ein sehr wesentlicher Teil des Indexierungsprozesses ist folglich die Anwendung eines *Keyword-Relevanzfilters* auf ein Textdokument mit dem Ziel, eine Liste an repräsentativen Deskriptoren zu generieren. Wie umfangreich die einzelnen Dokumente ausgewertet werden ist von System und Indexierungseinstellung unterschiedlich. Es besteht die Möglichkeit ein Dokument vollständig zu analysieren, d.h. alle Worte eines Dokuments zu berücksichtigen oder nur bestimmte Teilbereiche eines Dokuments zu indexieren. Bei HTML-Dokumenten kann sich eine partiale Kontextanalyse auch nur auf den Inhalt zwischen den HEAD-Tags

oder auf eine bestimmte Anzahl von Begriffen, wie z.B. die ersten 100 oder 200 Worte eines Dokuments, beschränken.

Das Finden und Vergeben von inhaltsbezogenen Deskriptoren dient drei wichtigen, miteinander in Beziehung stehenden Zielen:

1. Der Suche nach Dokumenten die für eine Anfrage relevant sind.
2. Die Verknüpfung von Dokumenten, die thematisch zusammengehören.
3. Der Relevanzbestimmung der einzelnen Dokumente, auf Basis von repräsentativen Begriffen, bezogen auf eine Suchanfrage.

Die automatisierte Deskriptorengewinnung mittels Keyword-Relevanzfilter soll in diesem Sinne so *erschöpfend* und *spezifisch* wie möglich erfolgen. Hierbei stellen relevante Bestimmungsmerkmale bei der Anzahl und Genauigkeit von repräsentativen Begriffen die Erfordernis an den *Recall* (Menge aller berücksichtigter Dokumente) und die *Precision* (Genauigkeit der Suchantwort) dar. So bedeutet *erschöpfend*, dass alle Themen eines Dokumentes mittels Deskriptoren repräsentiert sind, was positive Auswirkungen auf den Recall hat. *Spezifisch* bedeutet hingegen, dass aus der Gesamtheit möglicher Themen eines Dokumentes genau diejenigen gefiltert werden, die das betreffende Dokument am zutreffendsten repräsentieren, was zu einer Verbesserung der Precision führt.

Die Schwierigkeit eines Keyword-Relevanzfilters besteht bekanntermaßen darin, mittels automatisierter Verfahren zu entscheiden, *welche* Begriffe ein Dokument *grundsätzlich* repräsentieren und aus dieser Menge heraus diejenigen zu filtern, die ein Dokument *inhaltlich* am *genauesten* abbilden. Denn Ziel ist es, unter Berücksichtigung des jeweiligen Anspruchs an Recall und Precision, einer Suchabfrage möglichst genau die Dokumente in einem bestimmten Umfang als Ergebnis zu liefern, die der Anfrage auch am exaktesten entsprechen.

Zur Bestimmung welche Worttypen grundsätzlich geeignet sind um als Deskriptoren zu fungieren, ist eine Betrachtung der einzelnen Wortarten sowie deren Funktion im Satz hilfreich. Ein Satz in einer natürlichen Sprache besteht aus unterschiedlichen Wortarten, wie z.B. Substantiven, Verben, Adjektiven, Bindewörtern, die grammatikalisch erforderlich sind, um einen Gedanken auszudrücken. Jeder Text besteht wiederum aus einer Vielzahl an Wörtern, die in ihrer Gesamtheit dazu dienen, inhaltlich ein Thema zu beschreiben.

Die überwiegende Anzahl der Worte eines Textes ist jedoch nicht dazu geeignet, ein Dokument in der Form abzubilden, dass die einzelnen Worte das Thema auch tatsächlich *inhaltlich* in Bezug auf eine Suchabfrage repräsentieren. Einen Großteil von Worten die über alle Sachgebiete hinweg keinen Erkenntnisgewinn über den Inhalt eines Dokuments ermöglichen, sind die so genannten Füllwörter, wie z.B. bestimmte und unbestimmte Artikel (der, die, das, ein, eins, einer), Bindewörter (und, oder), Präpositionen, Wortkonjunktionen, Pronomen, Fragewörter oder auch Modalverben. Untersucht man die verbleibenden Worttypen genauer, welche einzeln betrachtet ein Thema inhaltlich am ehesten wiedergeben, zeigt sich, dass Inhalte am ehesten durch Substantive abgebildet werden können.

In diesem Zusammenhang wurde in empirischen Untersuchungen festgestellt, dass sich natürlichsprachige Texte unterschiedlicher Sachgebiete in ihrem Wortschatz erheblich unterscheiden. Auf einzelne Begriffe bezogen bedeutet das, dass die Häufigkeit eines Wortes das einem bestimmten Sachgebiet angehört, mit der Bedeutsamkeit dieses Begriffs für das jeweilige Sachgebiet korreliert.

Als Entscheidungsgrundlage, ob ein Wort als Schlüsselwort *wichtig* ist oder nicht, kann die *Häufigkeit seines Vorkommens* im Text Ausschlag gebend sein. So bauen die meisten Ansätze der automatisierten Indexierung auf der Beobachtung auf, dass die Häufigkeit einzelner Begriffe in einem natürlichsprachigen Text, mit der Bedeutsamkeit dieser Wörter für die inhaltliche Repräsentation korreliert. Es erkannte bereits H.P. Luhn, ein Pionier der automatischen Indexierung, dass ein unmittelbarer Zusammenhang zwischen Worthäufigkeit und Wortbedeutung für einen Text besteht.

Das *Prinzip des geringsten Aufwands* das auch als *Zipf'sche Gesetz* bekannt ist, besagt, dass es für den Verfasser eines Textes einfacher ist, bestimmte Worte zu wiederholen die ein Thema beschreiben, als ständig nach neuen Begriffen zu suchen. Dabei kommen diese den Inhalt repräsentierenden Schlüsselwörter zwar im Text verstärkt vor, aber im Verhältnis zur Gesamtwortmenge nur in *mittlerer* Worthäufigkeit. Auf Anhieb liegt die Vermutung nahe, dass diejenigen Worte die besonders häufig im Text auftreten, sich bevorzugt als Keywords eignen. Es hat sich aber empirisch gezeigt, dass sich sehr häufig auftretende Wörter nicht als Deskriptoren eignen. Weiter sind sehr selten auftretende Wörter auch keine geeigneten Deskriptoren.

Die Erkenntnis, dass Substantive Themen eines Dokuments repräsentieren können, führt zu der Indexierungsmethode von Retrievalssystemen, den Inhalt bzw. ein Thema von Textdokumenten über die Erfassung der vorkommenden Substantive abzubilden. Wie erwähnt, dienen jedoch nicht alle Substantive eines Texts dazu den Sinn wiederzugeben, sondern nur diejenigen, die mit *mittlerer Häufigkeit* vorkommen. Der generelle Zusammenhang von Worthäufigkeit und Relevanz lässt sich demzufolge wie nachfolgend abgebildet darstellen.

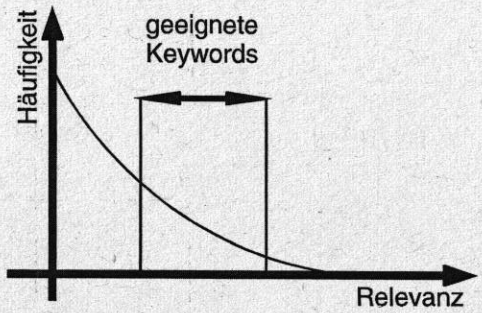

Abb. 3.4. Zusammenhang von Worthäufigkeit und Wortrelevanz

Links

Relevanzfilter und Ähnlichkeitsmaß von Dokumenten mittels Keywords
- [www2.iicm.edu/cguetl/education/student/hkonrad/seminar.html]

Information Retrieval and Search
- [www.cs.sfu.ca/~cameron/Teaching/D-Lib/IR.html]

Information Retrieval Indexierung: Grundidee
- [www.home.fh-karlsruhe.de/~rire0001/DVII/IR.pdf]

Automatische Indexierung strukturierter Dokumente
- [www.fh-darmstadt.de/all/tmp/tdf/knorz.doc]

Automatic Text Analysis
- [www.dcs.gla.ac.uk/Keith/Chapter.2/Ch.2.html]

3.2.8 Stoppwortliste und Black List

Stoppwortlisten sind Listen von Worten, die bei der Erfassung von Schlüsselwörtern zur Indexierung nicht berücksichtigt werden sollen. Ziel ist es mittels Stoppwortlisten irrelevante oder unzulässige Begriffe effizient zu identifizieren und zu eliminieren. Hierzu werden alle in einem Dokument erkannten Begriffe mit den Worten in der Stoppwortliste abgeglichen. Kommt ein Wort sowohl als Deskriptor als auch als Stoppwort vor, wird es aus der Liste der Keywords eines Dokuments gestrichen. Durch Stoppwortlisten ist es möglich, Begriffe die nicht im Index erscheinen sollen von der Indexierung auszuschließen. Häufig werden Worte auf die Stoppwortliste gesetzt, die gegen die Nutzungsordnung der Suchmaschine, gesetzliche Bestimmungen oder allgemeingültige Wertvorstellungen verstoßen.

Der Einsatz von Stoppwortlisten eignet sich auch sehr gut um grammatikalische Füllwörter wie bestimmte und unbestimmte Artikel, Bindewörter, Präpositionen, Wortkonjunktionen, Pronomen, Fragewörter sowie Modalverben von vorne herein auszuscheiden. Die Einträge in Stopplisten sind grundsätzlich sprachorientiert und können dynamisch angepasst werden. Die Anwendung von Stoppwortlisten kann im Zuge der Deskriptorengewinnung erfolgen oder als eigenständiger Teilprozess dem Keyword-Relevanzfilter vor- oder nachgeschaltet sein.

Neben den Stoppwortlisten kommen bei den Suchmaschinen auch *Black Lists* zum Einsatz. Gleich den Stoppwortlisten handelt es sich bei den Black List um Wortlisten von unzulässigen Worten. Der Unterschied zur Stoppwortliste ist die Art der Konsequenz, die sich an das Auftreten eines Wortes anschließt. Während die Stoppwortliste das betreffende Wort lediglich aus der Liste der zu indexierenden Worte löscht, führt das Vorkommen eines Wortes das sich auf der Black List befindet, zur Elimination des gesamten Dokuments.

Links

Automatische Dokumentenindexierung
- [www.iuk.hdm-stuttgart.de/nohr/KM/KmAP/Indexing.pdf]

Google Stopwords
- [www.ranks.nl/tools/stopwords.html]

3.3 Datenstrukturen der Information Retrieval Systeme

Die Notwendigkeit der Suchmaschinen Textdokumente inhaltlich so zu unterscheiden, dass Suchergebnisse entsprechend ihrer Relevanz sortiert werden können, erfordert spezielle Datenstrukturen. Diese Datenstrukturen müssen so angelegt sein, dass alle Dokumente im Datenbestand gefunden werden, die zu einer Suchanfrage relevant sind. Darüber hinaus müssen die Datenstrukturen dem Query Processor alle Informationen liefern, die es ermöglichen eine Differenzierung der Dokumente, bezogen auf ihre Relevanz zu einer Suchanfrage, vornehmen zu können. Die von der Mehrzahl der Suchmaschinen eingesetzte Datenstruktur ist das invertierte Dateisystem, das effiziente Suchen ermöglicht und in dem alle Informationen zur Unterscheidung der Relevanz gespeichert werden können.

3.3.1 Besonderheit der Datenstrukturen von IR-Systemen

In jedem Informationssystem existieren gewöhnlich zwei wesentliche Datenstrukturen. In der einen Datenstruktur werden die an das System übergebenen Dateien gesichert und verwaltet. Das Dateiformat in dem Daten gespeichert werden ist durch das System definiert und orientiert sich an den Erfordernissen der Verwaltung, Datenhaltung und dem Zugriff auf die Daten.

Eine zweite Datenstruktur macht den Zugang zu den gespeicherten Dokumenten über Suchanfragen möglich. Bei Suchmaschinen ist es das invertierte Dateisystem, das über entsprechende Verweise auf die gespeicherten Dokumente verfügt, um diese bei Suchanfragen zu finden. Ziel von Retrieval-Systemen ist es, eine durchsuchbare Datenstruktur anzulegen mittels derer es möglich ist, Textdokumente schnell und effizient zu finden. Für Information Retrieval Systeme bieten sich hierzu verschiedene Datenstrukturen wie z.B. die N-Gram-Datenstruktur, die PAT-Datenstruktur oder auch die Signature File-Datenstruktur an. Die jedoch am häufigsten eingesetzte Datenstruktur, sowohl in herkömmlichen bibliographischen Information Retrieval Systemen, als auch bei den Suchmaschinen, ist das „invertierte Dateisystem“ mit einer zentralen Indexdatei.

Die Erfordernis für eine besondere Datenstruktur liegt in dem Umstand, schwach strukturierte Textdokumente so verarbeiten müssen, dass es IR-Systemen

möglich ist, Suchanfragen nicht innerhalb der einzelnen Dokumente selbst auszuführen, sondern über eine gesonderte Datenstruktur. D.h. die einzelnen Dokumente werden so aufbereitet, dass sie durch eine bestimmte Anzahl von Schlüsselwörtern inhaltlich abgebildet werden. Dieser Systematik liegt die Erkenntnis zu Grunde, dass Textdokumente grundsätzlich nach bestimmten Themen, die sie inhaltlich abdecken, gesucht werden. Anders ausgedrückt, jemand der bei einer Suchmaschine eine Suchanfrage stellt erwartet Textdokumente, die sich *inhaltlich* möglichst exakt mit dem Thema beschäftigen, das durch die Suchwörter definiert wird.

In diesem Zusammenhang ist eine Unterscheidung zwischen den einzelnen Dokumenten erforderlich, was deren Relevanz zu einer Suchanfrage anbelangt. Während Tabellen orientierte Datenbanken alle Datensätze ohne inhaltliche Differenzierung liefern, ist ein wesentliches Merkmal der Information Retrieval Systeme Suchergebnisse in Hinblick auf ihre Relevanz zu unterscheiden. Diese Anforderung ist besonders bei Textdokumenten erforderlich, da die einzelnen Dokumente aufgrund ihrer inhaltlichen Ausarbeitung, bezogen auf eine Suchanfrage, sehr unterschiedlich relevant sein können.

Die Information Retrieval Systeme müssen also zunächst Textdokumente mittels geeigneter Datenstrukturen so erfassen, dass sie hinlänglich aller Themen die sie beinhalten erfasst und effizient aufgefunden werden können. Weiter muss es durch besondere Verfahren möglich sein, diejenigen Dokumente, die Teil eines Ergebnisses sind, entsprechend ihrer Relevanz zur Suchanfrage sowie zueinander diskriminieren zu können.

Die technische Realisation erfolgt auf Basis der Systematik eines gewichteten invertierten Dateisystems. Textdokumente sind in einem invertierten Dateisystem durch ihre Schlüsselwörter im Index und den ihnen zugeordneten invertierten Dateien so organisiert, dass eine Suchanfrage alle Dokumente liefert, die den bestimmten Suchbegriff beinhalten. Durch verschiedene Gewichtungsverfahren erhalten die einzelnen Dokumente im Zuge der Indexierung eine Bewertung, wonach sich berechnen lässt inwieweit sie einem bestimmten Thema entsprechen. Die Informationen zur Berechnung der Relevanz werden im invertierten Dateisystem mit abgespeichert.

Ein invertiertes Dateisystem bei Suchmaschinen basiert im allgemeinen auf drei verschiedenen Dateistrukturen,

- den direkten Dateien,
- den invertierten Dateien,
- und dem Index.

Die Gesamtheit aller Maßnahmen zur Entwicklung dieser Datenstrukturen wird Indexierung genannt. Nachfolgende Grafik zeigt eine allgemeine Struktur eines invertierten Dateisystems, welche jedoch bei den einzelnen Retrieval-Systemen individuell abweichen kann. Die gewählte Darstellungsform der Tabellen dient einer verbesserten visuellen Darstellung und entspricht nicht der realen Form der einzelnen Tabellen. Auf die Funktionsweise des invertierten Dateisystems wird nachfolgend detailliert eingegangen.

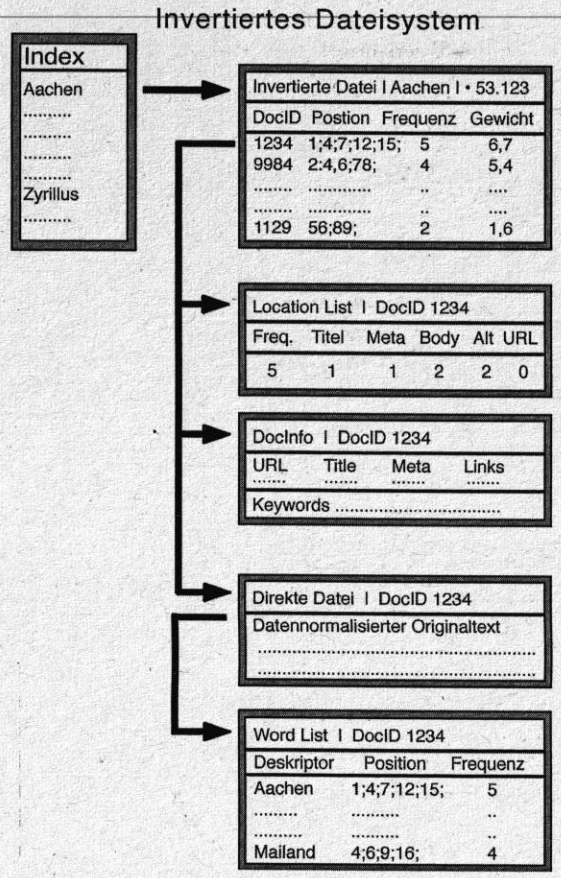

Abb. 3.5. Invertiertes Dateisystem – schematische Darstellung

Links

Index Compression vs. Retrieval Time of Inverted Files for XML Documents
- [www.is.informatik.uni-duisburg.de/bib/fulltext/ir/Fuhr_Goevert:02a.pdf]

Information Retrieval
- [www.uni-duisburg.de/FB3/CL/ses/docs/ir.html]

Texte und Quellen zum Thema Information Retrieval
- [www.hbi-stuttgart.de/nohr/Ir/ir.htm]

Computer-unterstütztes Indexieren
- [wwwai.wu-wien.ac.at/Publikationen/Kaiser/diss.html]

Research Resources on Cross-Language Text Retrieval
- [www.ee.umd.edu/medlab/mlir/papers.html]

Using Intelligent Agents to enhance search engines performance
- [www.firstmonday.dk/issues/issue2_3/jansen/index.html]

Adaptive Systems & Interaction Group, Microsoft Research
- [http://research.microsoft.com/~sdumais/]

Information Retrieval and Filtering
- [www.nersc.gov/~cding/papers/index.html#web]

Combining Content and Collaboration in Text Filtering
- [www.csee.umbc.edu/~ian/pubs/mlif.ps.gz]

File Structures
- [www.dcs.gla.ac.uk/~iain/keith/data/pages/66.htm]

File Structures
- [www.dcs.gla.ac.uk/Keith/Chapter.4/Ch.4.html]

3.3.2 Direkte Dateien

Die *direkten Dateien* stellen datennormalisierte Textdateien der Originaldokumente sowie mögliche ergänzende Dateien dar. Sie sind von ihrem Ursprungsformat (z.B. HTML) in ein System spezifisches Dateiformat konvertiert worden, so dass sie vom System intern möglichst effizient verarbeitet werden können. Die direkten Dateien werden entweder vollständig vom System gespeichert oder es erfolgt nur eine Speicherung ausgesuchter Bereiche des Dokuments. Eine partiale Erfassung und Speicherung kann sich beispielsweise auf bestimmte Head-Informationen wie den Dokumententitel oder verschiedene Meta-Tag-Angaben beschränken, aber auch eine bestimmte Anzahl von Worten innerhalb des Dokumentenkörpers beinhalten.

Verschiedene kommerzielle Suchmaschinen im Internet wie z.B. Google, speichern jedoch die konvertierten Originaldokumente vollständig in komprimierter Form ab, was ihnen ein Textstreaming bei der Volltextsuche ermöglicht. Die vollständige Speicherung aller indexierten Dokumente hat bei Google weiter den Vorteil, dass Webressourcen auch aus dem *Archiv* der Suchmaschine aufgerufen werden können und somit nicht nur als Original vom Server der betreffenden Datei verfügbar sind.

Neben dem konvertierten Originaltext enthält jede direkte Datei eine *Word List* in der alle Deskriptoren aufgeführt sind, die ihr bei der automatisierten Deskriptorengewinnung zugewiesen werden. Die Word List kann Teil der direkten Datei sein oder wie es gelegentlich der Fall ist, in Form einer eigenständigen Datei angelegt werden. Mit der Erfassung eines Dokuments wird jeder direkten Datei eine DocID (einzigartiger numerischer Identifikator) zugewiesen, über die sie vom System effizient verwaltet werden kann.

Links

Grundelemente Dateistrukturen
- [www8.informatik.uni-erlangen.de/IMMD8/Lectures/DOKUMENTENMANAGEMENT/v05.4.ps.gz]

3.3.3 Invertierte Dateien

Die *invertierten·Dateien* stellen eine Umkehrung der direkten Dateien dar. Eine invertierte Datei verweist auf sämtliche direkte Dateien, die einen bestimmten Begriff als Keyword in ihrer Word List führen. D.h. eine invertierte Datei für ein bestimmtes Schlüsselwort setzt sich aus Verweisen zu all denjenigen direkten Dateien zusammen, die durch einen bestimmten Deskriptor repräsentiert werden. Der Verweis auf die direkten Dateien erfolgt durch einen numerischen Identifikator, der in der invertierten Datei gespeichert ist. Bei Systemen die Gewichtungs- oder Klassifikationsverfahren unterstützen, wie das bei den Suchmaschinen allgemein hin der Fall ist, können in den invertierten Dateien neben dem numerischen Identifikator auch umfangreiche Informationen zur Gewichtung, wie z.B. der Positionen eines Keywords im Dokument oder auch die Häufigkeit der Vorkommnis im Dokument, gespeichert werden. Diese Informationen werden in der invertierten Datei entweder direkt dem Identifikator zugeordnet oder in einer weiteren Datei, der *Location Lists*, gespeichert, die über den jeweiligen Identifikator referenziert ist.

Eine wichtige Erweiterung der invertierten Datei kann, zur effizienteren Verarbeitung der Angaben zur Relevanzberechnung, eine separate Location List sein. Die Einträge der Location List dienen zur exakten Lokalisierung einzelner Worte im Dokument. Für jedes Keyword im Dokument gibt die Location List die genaue Anzahl der Vorkommnis, die Stelle eines Wortes innerhalb des Originaltextes sowie die Stelle innerhalb des URL an. Kommt ein Begriff in einem Dokument mehrmals vor, wird jede Vorkommnis exakt festgehalten und zusätzlich aufsummiert.

Die Location List differenziert die verschiedenen Angaben weitergehend und ermöglicht somit verfeinerte Gewichtungsverfahren. Von besonderer Bedeutung für die Bewertung von Dokumenten sind die Begriffe die sich im Dokumententitel, im Meta-Tag DESCRIPTION und im Meta-Tag KEYWORDS befinden. Weiter wird oftmals eine Unterscheidung vorgenommen, ob ein Begriff als Überschrift im Dokumentenkörper vorkommt. Überschriften in HTML-Dokumenten werden von den Suchmaschinen an Hand des HATML-<h1> bis <h6>-Tags identifiziert.

Innerhalb des im Browser sichtbaren Textbereichs, also innerhalb der Body-Tags eines HTML-Dokuments, kann weitergehend eine Unterscheidung nach der Lage im Dokument erfolgen. Die genaue Bestimmung der Position eines Wortes im Dokument oder im URL ermöglicht eine weiterführende, differenzierte Gewichtung. So können Deskriptoren, die weiter am Anfang eines Textes stehen, als wichtiger für die inhaltliche Repräsentanz eines Dokuments gewertet werden, als diejenigen, die später im Text vorkommen. Wird eine Wortposition bestimmt, erfolgt zudem die Zuordnung eines *absoluten* numerischen Werts, der es ermöglicht die relative Lage eines Schlüsselwortes in Abhängigkeit aller Worte im Dokument zu bestimmen. Durch dieses Verfahren ist es nicht nur möglich die Position zu bestimmen, sondern auch die *Distanzen* zwischen verschiedenen Begriffen zu berechnen. Durch eine Berechnung der Entfernung verschiedener Worte zueinander kann eine differenzierte Gewichtung erfolgen, die Wortfolgen mit kurzen Entfernungen als relevanter erkennt, als Wortfolgen mit größeren Entfernungen zu einander. Diese Methode wird *Proximity-Verfahren* genannt und gewichtet Dokumente höher, die Suchworte bei kombinierten Suchen möglichst nahe beieinander beinhalten.

Neben der Angabe der genauen Lage eines Deskriptor, erfolgt in der invertierten Datei auch ein Eintrag über dessen Häufigkeit im Dokument. Weiter wird die Worthäufigkeit aller Dokumente zu einem Wert aufsummiert. Dieser Wert zeigt an wie oft ein bestimmtes Wort insgesamt in allen Dokumenten und somit im gesamten Datenbestand vorkommt. Da laufend neue Dokumente im Datenbestand aufgenommen werden, muss dieser Wert periodisch aktualisiert werden. Die Werte über die Häufigkeit von Schlüsselwörtern in einem bestimmten Dokument als auch im gesamten Datenbestand stellen, wie sich nachfolgend noch zeigen wird, relevante Ausgangswerte für die statistischen Gewichtungsverfahren dar.

In der *Docinfo-Datei* werden weiterführende Informationen eingetragen, die eine Suchmaschine zur Darstellung eines Dokuments in der Suchergebnisliste benötigt. Die jeweiligen Informationen die hier gespeichert werden, sind von System zu System unterschiedlich. So extrahieren Altavista, Fireball und Lycos aus den HTML-Dokumenten zur Darstellung der Ergebnisliste den vollständigen Inhalt des Dokumententitel sowie das Meta-Tag DESCRIPTION als Kurzbeschreibung für ein Dokument, ergänzt um den URL. Google verzichtet hingegen auf die Angaben des Meta-Tag DESCRIPTION als Kurzbeschreibung und setzt stattdessen ein Textstreaming-Verfahren ein, das in Abhängigkeit des Suchbegriffs genau den Textausschnitt im archivierten Dokument darstellt, in dem sich der Suchbegriff befindet.

Links

Inverted Files
- [www.dcs.gla.ac.uk/~iain/keith/data/pages/72.htm]

Invertierte Datei
- [www.linguistik.uni-erlangen.de/tree/html/corsica/zierl97/node66.html]

Invertierte Datei
- [www.ib.hu-berlin.de/~wumsta/infopub/textbook/definitions/d98.html]

Indexierung mittels invertierter Dateien
- [www.cis.uni-muenchen.de/people/Schulz/SeminarSoSe2001IR/Nagy/node4.html]

File Structures
- [www.dcs.gla.ac.uk/Keith/Chapter.4/Ch.4.html]

3.3.4 Indexdatei

Der *Index* innerhalb des invertierten Dateisystems ist grundsätzlich eine sortierte Liste aller vorkommenden Begriffe, mit einem Verweis zu der jeweiligen invertierten Datei. Jedem Indexbegriff wird eine eindeutige invertierte Datei zugeordnet, die alle Verweise zu denjenigen Dokumenten beinhaltet, die den betreffenden Begriff als Deskriptor führen.

Bei einer kontrollierten Indexierung liegt dem Index ein genaues Wörterbuch der zulässigen Begriffe zu Grunde. Dadurch werden nur Deskriptoren über das System indexiert, die Element des Wörterbuchs sind. Das bedeutet u.a. auch, dass orthographisch falsch geschriebene Worte nicht indexiert werden. Der Einsatz eines Wörterbuchs ermöglicht weiter eine Kontrolle über die Zulässigkeit von Worten. So können Begriffe oder auch Dokumente in denen die Begriffe vorkommen ausgeschlossen werden, sofern sie nicht im Wörterbuch vorkommen. Dem Vorteil der Kontrolle über die indexierbaren Begriffe steht der nicht unerhebliche Nachteil gegenüber, dass neue Worte die in einer von Wissenschaften geprägten Gesellschaft sehr schnell entstehen, nur zeitlich sehr verzögert aufgenommen werden. Das Ergebnis ist, dass Datenbestände von Suchmaschinen die kontrollierte Indexierung betreiben, niemals wirklich aktuell sind.

Bei unkontrollierter Indexierung werden hingegen alle Begriffe die als Deskriptoren vom Keyword-Filter generiert werden, unabhängig ihrer genauen Schreibweise indexiert. Eine unkontrollierte Indexierung schließt jedoch nicht die Anwendung von Black Lists aus. Über die Methodik einer unkontrollierten Indexierung können zwar sehr einfach neue Worte und Schreibweisen in den Index aufgenommen werden, gleichzeitig wird jedoch auch jedes falsch geschriebene Wort indexiert.

Eine Strategie der unkontrollierten Indexierung verfolgt u.a. Google, bei der auch falsch geschriebene Worte in den Index aufgenommen werden, obwohl Google über ein sehr gutes und umfangreiches Wörterbuch verfügt. Das Wörterbuch bei Google wird nicht nur im Zuge der Indexierung eingesetzt, sondern auch bei Suchanfragen. Wird eine Suchanfrage orthographisch falsch gestellt, wird neben den gefundenen falsch geschriebenen Suchergebnissen auch eine Suchanfrage mit der korrigierten Schreibweise angeboten. Altavista, Lycos und Fireball liefern hingegen ohne Hinweis auf die richtige Schreibweise alle Dokumente als Suchergebnis, die das betreffende Keyword in der falsch geschriebenen Form beinhalten. Eine Fehlerkorrektur erfolgt nicht.

Eine wichtige Festlegung der Indexierungsparameter im Zusammenhang mit der Behandlung der Schreibweise von Worten, stellt die Handhabung unterschiedlicher Schreibweisen in Hinblick auf die Groß- und Kleinschreibung dar. Ein Wort kann entweder ausschließlich mit Großbuchstaben (1), Kleinbuchstaben (2) oder gemischt in Groß- und Kleinbuchstaben (3), (4) geschrieben werden:

(1) HAUS
(2) haus
(3) Haus
(4) hAuS

Eine Berücksichtigung der exakten Schreibweise führt in obigem Beispiel bei Zeichen genauer Indexierung zu vier unterschiedlichen Deskriptoren. Im Allgemeinen unterscheiden jedoch Information Retrieval-Systeme bei kontrollierter Indexierung nur zwischen Schreibweise (1) und (2). Erscheint ein Wort in der unter (3) oder (4) dargestellten Form wird es in die Schreibweise (2) konvertiert. Relevant im Hinblick auf die Bestimmung von Keywords ist bei der Indexierung noch die Unterscheidung zwischen Fall (1) und Fall (2). So existieren Suchmaschinen die bei einer vollständigen Großschreibung eines Wortes (Fall 1) dieses als eigenes Indexwort erfassen. Diese Strategie verfolgt u.a. Fireball, während Google, Altavista und Lycos alle Worte, unabhängig ihrer Schreibweise, nur in der Form (2) sowohl bei der Indexierung als auch bei Suchanfragen berücksichtigen. Wird ein Wort in der Form (1) sowohl im Dokument als auch in einer Suchanfrage verwendet, erfolgt immer eine System interne Umwandlung in die Form (2).

Um einen Index für alle Dokumente im System aufzubauen wird wie dargestellt, die Dokumentenorganisation von direkten Dateien invertiert; d.h. umgekehrt und über invertierte Dateien für jeden einzelnen Term abgebildet. Dadurch werden die Dokumente bei einer Suchanfrage über den jeweiligen Begriffe im Index erreicht. Üblicherweise ist ein Index sequentiell nach Schlüsselwörtern geordnet. Mit dem Index kann somit jeder inhaltsbeschreibende Begriff als Zugriffsschlüssel zu den Dokumenten benutzt werden.

Um beispielsweise alle Dokumente zu finden die den Begriff „Computer“ beinhalten, wird der Index auf diesen Begriff hin sequentiell durchsucht und die betreffende invertierte Datei identifiziert. Das invertierte Dateisystem liefert als Ergebnis

eine Liste all derjenigen Dokumente, die im Originaldokument den gesuchten Begriff beinhalten. Zur Identifikation der betreffenden direkten Dateien wird der Identifikator eingesetzt. Zusätzlich werden weitere Informationen, die für eine Gewichtung erforderlich sind, in der internen Ergebnisliste mitgeliefert. Da die gesamten Informationen von Dokumenten über den Index und die invertierten Dateien verfügbar sind, muss nicht mehr jedes einzelne Dokument auf einen Suchbegriff hin durchsucht werden. Dies zählt als einer der Hauptvorteile der invertierten Dateien und führt zu einer erheblichen Leistungssteigerung bei Suchanfragen.

Bei sehr großen Datenbeständen und einer sehr umfangreichen Anzahl an Begriffen im Index kann ein einfaches lineares Listing der Indexworte, die linear sequentiell durchsucht werden, nicht hinreichend effizient sein. Ein Verfahren zur Leistungssteigerung in invertierten Dateisystemen ist beispielsweise der Einsatz von B-Trees, die anstelle eines linearen Indexes auf die invertierten Listen verweisen. Systemtechnisch kann diese Dateiorganisation mittels lokaler Dateiorganisation auf einem Server oder mittels globaler Dateistruktur auf mehrere Server verteilt werden. Bei global organisierten invertierten Dateien verfügt jeder Server über eine logisch organisierte Teilmenge (z.B. nach Buchstabenfolgen a-c, d- f oder Deskriptoren: haur, haus, haut,...) aller invertierten Dateien. Über eine relativ flache Baumstruktur werden entsprechend den Buchstabenfolgen von Deskriptoren diese schnell gefunden. Jede Hierarchieebene repräsentiert dabei eine Position des betreffenden Buchstabens im Begriff. Ein Beispiel für den Begriff „Haus“ macht dies deutlich.

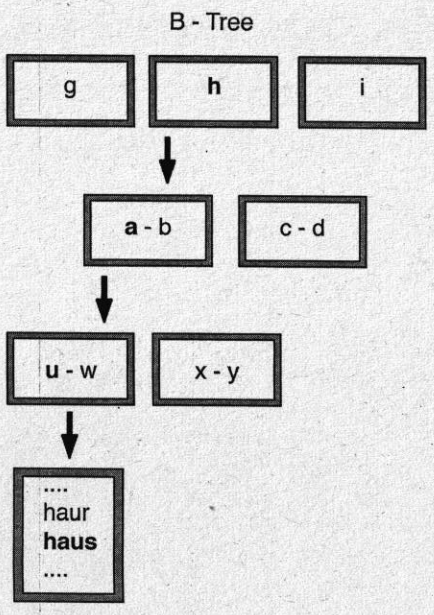

Abb. 3.6. Beispiel eines b-tree organisierten invertierten Dateisystems

Eine der wichtigsten Entscheidung bei der Erstellung eines Indexes ist die Definition der Indexierungsstrategie und damit verbunden die Bestimmung geeigneter Gewichtungsmodelle (s. Kap. 4). D.h. ob und wie Dokumente sowie deren Deskriptoren gewichtet und mittels welcher Retrieval-Funktion geeignete Dokumente identifiziert und bewertet werden können.

Indexe stellen den Zugang zur Datenbasis für die Anfragen via Querry Processor, der Suchkomponente einer Suchmaschine, dar. Entsprechend der definierten Suchstrategien (z.B. einfache Keywordsuche, Suche mittels Boolescher Operatoren, Volltextsuche, Expertensuche, etc.) und deren Retrieval-Funktionen müssen die Indexe so konzipiert sein, dass sie hierzu über alle erforderlichen Daten und Gewichtungsinformationen verfügen und diese effizient organisieren.

Im nachfolgenden Kapitel beschäftigen wir uns ausgiebig mit den verschiedenen Gewichtungsmodellen, auch Indexierungsmodelle genannt, die bei den Suchmaschinen zum Einsatz kommen.

Links

Index-sequential files
- [www.dcs.gla.ac.uk/~iain/keith/data/pages/72.htm]

B-Trees
- [www.dcs.gla.ac.uk/~iain/keith/data/pages/83.htm]

Sortierung der Indexdatei
- [www.linguistik.uni-erlangen.de/tree/html/corsica/zierl97/node67.html]

File Structures
- [www.dcs.gla.ac.uk/Keith/Chapter.4/Ch.4.html]

4 Relevanz und Gewichtungsmodelle

Das wesentlichste Unterscheidungsmerkmal von Information Retrieval Systemen im Vergleich zu klassischen Tabellen orientierten Datenbanksystemen ist deren Funktionalität, Suchergebnisse entsprechend ihrer *Relevanz* zu einer Suche differenzieren zu können. Relevanz kann auch im Sinne von *Ähnlichkeit* gedeutet werden. Information Retrieval Systeme sind so konzipiert, dass es möglich ist, aus der Gesamtheit aller im Datenbestand vorhandenen Dokumente, genau diejenigen Textdokumente zu finden, die zu einer Suchanfrage ein Mindestmaß an Ähnlichkeit besitzen. Dies erfordert von den Suchmaschinen relevante Dokumente von irrelevanten Dokumenten unterscheiden und relevante Dokumente hinlänglich ihres Ähnlichkeitsgrads zur Suchanfrage sortieren zu können.

Um eine Unterscheidung vornehmen zu können, welche Dokumente eines Datenbestandes inhaltlich über einen Bezug zu einer Suchanfrage verfügen und zu welchem Grad eine Ähnlichkeit besteht, müssen Gewichtungsmodelle eingesetzt werden, die Dokumente hinlänglich ihrer Relevanz zu einer Suchanfrage unterscheiden können. Nachfolgend werden die wichtigsten Gewichtungsmodelle dargestellt, die von Suchmaschinen im Internet verwendet werden. Sie lassen sich grob in *Vektorraum basierte Gewichtungsmodelle* und *Hypermedia basierte Gewichtungsmodelle* unterteilen. Da die Bestimmung von Relevanz auf mathematischen Modellen zur Berechnung der Ähnlichkeit basiert, muss bei der Beschreibung der einzelnen Modelle hierauf Bezug genommen werden. Die Erklärung der einzelnen mathematischen Modelle ist jedoch allgemein gut verständlich.

Was unter dem Begriff *Relevanz* zu verstehen ist und warum er ein wichtiges Merkmal bei der Verarbeitung und Suche von Dokumenten in einem Datenbestand darstellt, wird nachfolgend ausführlich erklärt. Ein grundlegendes Verständnis über die Bedeutung der Relevanz und damit verbunden, eine Kenntnis über die einzelnen Gewichtungsmodelle ist Voraussetzung, um eine Website für Suchmaschinen optimieren zu können.

4.1 Zusammenhang von Relevanz und Rangbildung

Betrachten wir bekannte Datenbanksysteme wie z.B. SQL basierte Datenbanken, so basieren sie auf einem binären Entscheidungskriterium, das nur dann Datensätze als Ergebnis berücksichtigt, wenn sie einer Suchanfrage zu 100 Prozent entsprechen. Faktisch erfolgt dies durch einen Vergleich des Suchstrings mit dem