Thema: **Web-Page Crawler auf der Basis von Konzepten von Cho, Molina, Page**
Veranstaltung: Google tanzt. - Zur Anatomie einer Suchmaschine
Dozent: Professor Stock
Referent: Daniel Ritter
Semester: Sommersemester 2004

Was ist ein Crawler ?

Ein Crawler ist ein **Softwareprogramm** Ein Crawler ruft **automatisiert** Webseiten auf und findet auf ihnen **Hypertext-Links zu weiteren Webseiten**, welche danach ebenfalls in einer **festgelegten Reihenfolge aufgerufen** werden. Dies findet nach vorher festgeschriebenen **Regeln** statt. Ein Crawler ist an einen **Index** gekoppelt, in welchen er die gefundenen Seiten einspeist. Ein Crawler kann auch zur **Index-Pflege** benutzt werden, indem er nicht mehr existierende Seiten aus dem Index **entfernt** oder geänderte und umgezogene Seiten im Index **aktualisiert**.	Schritt 1 Übergabe einer Start-URL an den Crawler Schritt 2 Einlesen des Inhaltes der Start-URL. Schreiben der gefundenen Links in einen Speicher. Weitere individuelle Schritte für den spezifischen Crawler Schritt 3 **Einlesen des Inhaltes einer der URL's aus dem URL-Speicher.** Schreiben der gefundenen Links in den Speicher. Weitere individuelle Schritte für den spezifischen Crawler Schritt 4 Überprüfen, ob der Speicher leer ist oder eine andere Abbruchbedingung erfüllt wurde. Falls weiter gecrawlt werden soll, Rücksprung zu Schritt 3. Ansonsten Sprung zu Schritt 5. Schritt 5 Ende des Crawlvorgangs

Web-Page Crawler auf der Basis von Konzepten von Cho, Molina, Page

„Wenn man nicht zufällig IRGENDEINE URL als nächstes besucht, sondern versucht die relevantesten Seiten zuerst zu besuchen, wird der Index der Suchmaschine qualitativ besser werden. Ausserdem können Ressourcen gespart werden“

Kriterien nach denen die nächste zu crawlende URL ausgesucht werden kann: **1. Ähnlichkeit zu einem Suchwort für den Crawler (Similarity to driving query)** Seiten werden zuerst gecrawlt, auf die Links zeigen, die dem Suchwort ähnlich sind. Beim aktualisieren von Seiten werden solche zuerst gecrawlt, die das Suchwort (oft) enthalten. **2. Backlink Anzahl** Die Anzahl der Links die auf eine URL zeigen generieren ihren Backlinkwert **3. Pagerank** Einsatz des Pageranks, besonders interessant für bereits erfasste Seiten die aktualisiert werden sollen, bei nicht bekannten Seiten muss der PR geraten werden. **4. Location Metric** Bestimmte Eigenschaften der URL des Dokuments, z.B. wenige Unterverzeichnisse, Endung .com, bestimmtes Suchwort in URL enthalten	Verschiedene Abbruchbedingungen für Crawler: **1. Crawl & Stop** In diesem Konzept besucht der Crawler eine festgelegte Anzahl von K Seiten mit einer beliebigen Start-URL. Ein ideal funktionierender Crawler hätte nach dem Besuch von K Seiten aus allen verfügbaren Seiten nur die K relevantesten extrahiert. Wären also N Seiten durch Links (K < N) verfügbar gewesen, hätte der Crawler aus diesen N verfügbaren Seiten die K relevantsten Seiten extrahiert und den Vorgang beendet. Einfach ausgedrückt könnte man sagen, dass der Crawler eine gewissen Menge von Seiten besuchen darf, die Menge der verfügbaren Seiten jedoch grösser ist und er nur die relevantesten Seiten aus der Gesamtmenge besuchen sollte. **2. Limited Buffer Crawl** Auch bei diesem Vorgang besucht der Crawler eine festgelegte Anzahl von K Seiten. Sein Speicherplatz in den er erfasste Seiten ablegen kann ist jedoch nur S gross (S < K). Der Crawler hat also nicht genug Platz um alle gecrawlten Dokumente unterzubringen. Hier entfernt der Crawler während des Crawlingvorgangs Dokumente aus seinem Speicher um Platz für neue zu schaffen. Bei einem ideal funktionierenden Crawler wären nach dem Erfassen von S Dokumenten nur noch die relevantesten aus der Gesamtmenge K im Speicher. Da dem Crawler die Relevanz von neue erfassten Dokumenten nicht bekannt ist muss er versuchen sie auf der Basis der bereits vorhandenen Dokumente zu „erraten“. **3. Crawl & Stop with Threshold (Threshold = Schwelle)** Hier wird vorgegangen wie bei der ersten Methode. Jedoch wird ein Ziel-Qualitätskriterium für das Crawlen vergeben. Nach dem Crawvorgang von K Dokumenten muss ein perfekter Crawler alle Dokumente mit dem höchsten Qualitätswert erfasst haben.

Eigenschaften des Googlebots

Googlebot ist auf mehrere Rechner **verteilt**. Es gibt 2 verschiedene **Googlebot-Varianten** 64.*.*.* (Freshbot läuft ständig) 209.*.*.* (Deepbot läuft einmal im Monat) Seiten mit **höherem Pagerank** werden **häufiger besucht** Googlebot **crawlt langsamer als er könnte** um Server nicht voll auszulasten. Für lokale Google-Sites wie z.B. google.de werden **lokalsprachige Dokumente anders gewichtet.** Googlebot hält sich an die **robots.txt**	Seit Googlebot **berücksichtig Backlinks nur bei Seiten mit einem PR > 3** Freshbot **besucht eine Seite sehr schnell**, nachdem man sich für**Ad-Sense** angemeldet hat. Googlebot kann folgende **Dateiformate** besuchen und speichern: html, pdf, asp, jsp, hdml, shtml, xml, cfm, doc, xls, ppt, rtf, wks, lwp, wri. Das Schalten von A**d-Words** für eine Seite **beeinträchtigt NICHT ihr Ranking** und die Besuche vom Bot (offizielles Google Statement) Das **Ändern der URL** einer Seite wird das Ranking der Seite und damit auch die **Besuchsfrequenz des Bots verändern**, auch wenn die Inhalte gleich bleiben.

Literatur:

Chu Molina, Page - Efficient crawling through URL ordering
http://www7.scu.edu.au/programme/fullpapers/1919/com1919.htm

Definitions for crawler
Google Suche: define:crawler

Google's FAQ for the Googlebot
http://www.google.com/bot.html

The GNU wget crawler
http://www.gnu.org/software/wget/wget.html

The robots exclusion standard
http://www.robotstxt.org/

Google Watch - A look at how Google's monopoly, algorithms, and privacy policies are undermining the Web
http://www.google-watch.com

Download:

Download des Thesenpapiers und der Präsentation unter
http://www.daniel-ritter.de/arbeiten/