Maik Metzen, Erich Kachel

Ressourcen optimieren: das Budget-Problem bei Suchmaschinen

Suchmaschinen müssen wie alle Unternehmen effizient mit ihren Ressourcen umgehen. Daher crawlen sie für eine Domain nicht immer alle Seiten, die dort vorhanden sind. Webmaster können und sollten daher über ein vernünftiges Indexierungsmanagement steuern, dass genügend Raum für das Crawlen der wichtigen Seiten übrig bleibt. Der Beitrag von Maik Metzen und Erich Kachel gibt Hilfestellung, wie insbesondere größere Sites die Anzahl zu crawlender Seiten optimieren können.

Auf der Suche nach neuen Informationen auf bekannten URLs bzw. Webadressen und Verlinkungen zu neuen, bisher noch unbekannten Webseiten sind die Crawler von Google & Co. darauf angewiesen, diese möglichst einfach auffinden zu können. Aus Gründen der Wirtschaftlichkeit teilen sie einer Domain je nach ihrer eingeschätzten Wichtigkeit ein bestimmtes Budget an zu crawlenden Seiten zu. Die Unterseiten finden sie in der Regel über die interne Verlinkung und über ggf. eingereichte Sitemaps.

Durch eine ungünstige Verlinkung kann es gerade auf umfangreicheren Sites passieren, dass die Crawler immer wieder im Kreis herumgeschickt oder auf immer tiefere Linkpfade geleitet werden und das Crawlingbudget daher falsch bzw. nicht richtig ausgenutzt wird. Aus Sicht des Seitenbetreibers sollte der Crawler jedoch bevorzugt schnell neue Inhalte und Topthemen erfassen, wenig signifikanten Inhalten nur wenig Ressourcen zuteilen und auch nur die passenden URLs in den Suchindex der Suchmaschine aufnehmen.

Doch wie entstehen solche Crawler-Fallen? Typischerweise tauchte diese Symptomatik erst mit dem Aufkommen komplexer Content-Management- und Shop-Systeme auf. Mithilfe solcher Systeme lassen sich fast beliebige Strukturen derselben Inhalte generieren und miteinander verlinken. Genau hier kann es passieren, dass anstatt einer eindeutigen Linkstruktur eine fast willkürlich wirkende, sich zudem noch dynamisch ständig ändernde Linklandschaft entsteht. Um diesen Umstand zu verstehen, ist festzuhalten, dass für den Crawler jede kleinste Variation einer URL als eigenständig, also auch als eigene Seite bewertet wird – auch wenn der Inhalt unverändert ist. Wenn also z. B. innerhalb einer Produktkategorie eines Online-Shops nur zwei Produkte zu sehen sind, so kann ein Nutzer mit einem Blick das günstigere Produkt erfassen – der Crawler findet aber für jeweils aufsteigend und absteigend nach Preis sortierte Ergebnisse zwei Seiten, die einzeln besucht werden müssen. Hier werden also Ressourcen für eine Seite verwendet, die einer anderen – außer in der Sortierung – inhaltlich gleicht. Wäre dies auf einer Domain nur ein einmaliger Einzelfall, wäre das natürlich kein Problem. Stellt man sich aber vor, dass bei jeder Seite mit Sortiermöglichkeit eine zweite Kopie erzeugt wird, summiert sich dies durch die systemisch erzeugte Vervielfältigung ganz schnell auf.

Optimierung des Crawl-Vorgangs

Suchmaschinencrawler versuchen normalerweise, Links im Dokument korrekt zu erkennen und ihnen zu folgen. Ein HTML-Dokument kann aber neben den HTML-typischen Linkstrukturen wie A-Tags auch Verlinkungen über JavaScript-Events oder Pop-up-Fenster enthalten. Simple JavaScript-Verlinkungen kann Google schon seit einiger Zeit erkennen, wie die Abbildung 1 unten zeigt. Steigt jedoch die Komplexität der Umsetzung, werden also JavaScript-Funktionen zusätzlich aufgerufen oder die URLs aus Stücken erst durch JavaScript zusammengesetzt, erkennt sie der Crawler nicht. Diese können dann dazu verwendet werden, um den Crawl-Vorgang an dieser Stelle zu optimieren, wobei der reguläre

DER AUTOR

Maik Metzen ist Geschäftsführer der AKM3 GmbH und seit über zehn Jahren im Online-Marketing aktiv. Er ist Co-Founder des Zigarrenshops Noblego und Co-Autor des SEO-Buchs „SEO – Strategie, Taktik und Technik“

DER AUTOR

Erich Kachel ist Wirtschaftsinformatiker und CTO der AKM3 GmbH. Er verantwortet die Technik und die Produktentwicklung der internen Tools und ist Ansprechpartner für technisches SEO und Websicherheit.

	OPTIMIERTE VERLINKUNGEN UNTERSCHIEDLICHEN GRADES	GOOGLE SPIDER
1	<li data-href="/produkte/hosen/">	Nein
2	<li onclick="location.href='/produkte/hosen/'">	Nein
3	<li onclick="link(,/produkte/hosen/')">	Nein
4		Nein
5	<a data-extend="L3Byb2R1a3RlL2hvc2VuLw==>	Nein
6	<li data-href="http://www.shop.de/produkte/hosen/">	Ja
7		Ja
9		Ja
10		Ja
11		Ja
12		Ja
13		Ja

Abb. 1: Übersicht optimierter Verlinkungen unterschiedlichen Grades; in den ersten fünf Beispielen gelangt der Nutzer zu einer neuen Seite, ohne dass eine zusätzliche URL gecrawlt wird

Nutzer sie weiterhin wie gewöhnlich verwenden kann.

Die Technik dahinter setzt darauf, die Links nicht in üblicher HTML-Syntax auszugeben. Stattdessen werden entweder Design-Elemente oder aber auch A-Tags mit einer Click-Eigenschaft erweitert und etwas umgeschrieben. Immer jedoch wird versucht, keine ganzen URLs im Quelltext sichtbar zu hinterlegen.

Beispiel Nummer 5 ist von besonderer Natur, denn hier wird der Link komplett mit base64 codiert. Eine fast übertriebene Maßnahme in Anbetracht der in der Tabelle gezeigten „Blindheit" des Crawlers gegenüber leicht umgestalteten Links, jedoch vielleicht angebracht, wenn sichergestellt werden soll, dass einzig ein JavaScript-Interpreter in der Lage ist, diese Links auch tatsächlich zu entziffern und sie damit auf lange Zeit automatischen Crawlern zu entziehen.

Der richtige Umgang mit Produktfiltern

Ein weiteres bekanntes Problem vieler Online-Shops ist die Anzahl an Links, die benötigt wird, wenn die Produktauswahl eingegrenzt werden soll. Beispielhaft wird von einer einfachen Auswahl bestehend aus „Farbe", „Länge" und „Umfang" ausgegangen. Eine noch heute gängige Umsetzung ist dabei das Setzen von GET-Parametern mit der Auswahl der jeweils gewählten Produkteigenschaften. Bereits bei dieser beispielhaften Auswahl aus drei Filtern mit je drei Optionen ergeben sich 36 mögliche URL-Kombinationen.

Beispiel vorhandene Filterauswahl:

- **Farben:** blau, grün, rot
- **Länge:** 32, 33, 34
- **Umfang:** 32, 33, 34

Eine einfache Kombination aus diesen Filtermöglichkeiten ergibt eine Reihe von URLs (siehe Abbildung 2).

Diese URLs werden bei einer normalen Verlinkung von Crawlern erfasst und es werden aus Sicht der Suchmaschine neue URLs, also vermeintlich neue Seiten erzeugt. Dabei zeigen sie tatsächlich aus Sicht des Besuchers jeweils nur eine Unterauswahl bereits bekannter Produkte. Diese unnötigen Seiten zahlen aber auf das Crawlingbudget ein und verringern dieses Stück für Stück.

Es gibt grundsätzlich zwei übliche Techniken, um dieses Problem zu umgehen. Die erste besteht im Ersetzen der URLs durch solche, die von den Crawlern nicht erkannt werden können. Der Abschnitt zu „Optimierung des Crawl-Vorgangs" beschreibt mögliche Umsetzungen.

FILTEREINSTELLUNG	ENTSTANDENE URL
Farbe: Blau	/suche/?farbe=blau
Farbe: Blau, Länge: 32	/suche/?farbe=blau&laenge=32
Farbe: Blau, Länge: 33	/suche/?farbe=blau&laenge=33
Farbe: Blau, Länge: 34	/suche/?farbe=blau&laenge=34
Farbe: Blau, Länge: 32, Umfang: 32	/suche/?farbe=blau&laenge=32&umfang=32
Farbe: Blau, Länge: 33, Umfang: 32	/suche/?farbe=blau&laenge=33&umfang=32
Farbe: Blau, Länge: 34, Umfang: 32	/suche/?farbe=blau&laenge=34&umfang=32
Farbe: Blau, Länge: 32, Umfang: 33	/suche/?farbe=blau&laenge=32&umfang=33
Farbe: Blau, Länge: 33, Umfang: 33	/suche/?farbe=blau&laenge=33&umfang=33
Farbe: Blau, Länge: 34, Umfang: 33	/suche/?farbe=blau&laenge=34&umfang=33
[…]	[…]
Farbe: Grün	/suche/?farbe=gruen
Farbe: Grün, Länge: 32	/suche/?farbe=gruen&laenge=32
Farbe: Grün, Länge: 33	/suche/?farbe=gruen&laenge=33
Farbe: Grün, Länge: 34	/suche/?farbe=gruen&laenge=34
Farbe: Grün, Länge: 32, Umfang: 32	/suche/?farbe=gruen&laenge=32&umfang=32

Abb. 2: Exemplarische URLs, die typischerweise beim Filtern von Produkten entstehen

Die zweite Technik verwendet HTML-Formulare, um die Filtereinstellungen zu setzen. Grundsätzlich wird dabei der Umstand genutzt, dass Crawler keine Formulare absenden und somit auch keine Seiten generiert werden können. Hier gibt es weitere Verbesserungen, die zusätzlich umgesetzt werden können. Beispielsweise können JavaScript-Ereignisse verwendet werden, um die Filter im Augenblick der Auswahl zu setzen, ohne Bestätigung eines Absende-Buttons. Die Varianten können sich hier unterscheiden. Wichtig dabei ist stets, dass im Quellcode keine für Crawler verwertbaren Links zu den Filtereinstellungen zu finden sind. Exemplarisches Setzen der Filter über JavaScript-Events (siehe Abbildung 3)

```
<select id="products-sort" onchange="location.href='http://www.
shop.de/produkte/hosen/' + this.value;">
<option value="?sort=desc">Preis absteigend</option>
<option value="?sort=asc">Preis aufsteigend</option>
</select>
```

Abb. 3: Exemplarisches Setzen der Filter über JavaScript-Events

Abb. 4: Typische interne Verlinkung einer Paginierung

```
[...]
/produkt/hosen/?seite=202
/produkt/hosen/?seite=203
/produkt/hosen/?seite=204
/produkt/hosen/?seite=205
/produkt/hosen/?seite=206
[...]
```

Abb. 5: Exemplarische URLs von Paginationsseiten

Diese Vorgehensweise macht vor allem dann Sinn, wenn man viele Filtermöglichkeiten hat. Auch sollte man überlegen, erst ab einer gewissen Filtertiefe auf diese Methode zurückzugreifen, da Filterungen auf 1. und oft auch auf 2. Ebene relevant sein können für das Auffinden im Google-Index. Diese sollten daher auch mit dem Meta-Tag „robots“ und der Direktive „index, follow“ ausgezeichnet werden.

Der richtige Umgang mit Paginierung

Ein weiteres und gewichtiges Problem in Online-Shops sind die Paginierungsseiten bei langen Produktlisten. Sie lassen sich nicht vermeiden, möchte man doch dem Nutzer die Möglichkeit lassen, durch die Produkte zu stöbern. Ist die Anzahl an Produkten jedoch groß, können diese Hunderte einzelne Seiten zur Darstellung benötigen.

Bei einer Paginierung wie in Abbildung 4 ergeben sich z. B. URLs für die einzelnen Seiten wie sie in Abbildung 5 zu sehen sind.

Die daraus resultierenden einzelnen URLs führen dazu, dass die Crawler sehr viel Zeit mit dem Folgen der erzeugten Seiten verbringt – allein in diesem Beispiel wären es 208. Aus einer solchen Anzahl an ähnlichen Seiten ergeben sich in der Regel keine wirklichen SEO-Vorteile, aber einige Nachteile: Die Seiten sind unvorteilhaft verlinkt und erzeugen eine Linkkette. Sie stellen nur sicher, dass alle Produkte gecrawlt werden können.

Diesem Umstand kann man unterschiedlich begegnen. Ein Ansatz besteht aus einer Kappung der Paginierung auf z. B. 50 Seiten. Dies hat zur Folge, dass mehrere Produkte auf einer Paginationsseite platziert werden müssen. Durch die Kappung findet der Crawler ein definiertes Ende für die Paginierung und kann gesparte Ressourcen für eine weitere Produktkategorie verwenden.

Ein anderer, radikalerer Ansatz besteht im Setzen von JavaScript-Links, wie im Abschnitt zu „Optimierung des Crawl-Vorgangs“ beschrieben. Kombinieren kann man diese Technik mit der Kappung auf 50 Seiten – ab Seite 51 sind die Links dann für den Crawler nicht mehr auffindbar, der Nutzer kann aber weiterhin stöbern. Entscheidet man sich für eine solche Herangehensweise, muss sichergestellt werden, dass die wichtigsten Produkte über die Paginierung gecrawlt werden können und dass eine ordentliche Sitemap angelegt wird (im besten Fall eine oder mehrere Sitemaps für Produkte). Generell empfiehlt es sich, auffindbare Paginationsseiten mit dem Meta-Tag „robots“ und der Direktive „noindex, follow“ auszuzeichnen sowie die durch den Crawler erreichbaren Seiten untereinander mit dem Link-Tag und den Direktiven „prev“ und „next“ logisch zu verbinden.

Generell ist eine Crawl-Optimierung auch durch entsprechende Ausschlüsse über die robots.txt sowie die Webmaster-Tools empfehlenswert, jedoch dienen die oben beschriebenen Ansätze dazu, insbesondere bei größeren Seiten die Anzahl entstehender URLs zu minimieren. Die Ansätze haben zudem den Vorteil, dass alle Nutzer weiterhin auf der Seite navigieren können und redundante URLs vom Crawling ausgeschlossen werden.¶