Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Studiengang Geoinformatik

Aufbau und Vergleich webbasierter Suchmaschinen

Bachelorarbeit

vorgelegt von: *Tino Schuldt*

Zum Erlangen des akademischen Grades

„Bachelor of Engineering“ (B.Eng.)

Erstprüfer:	Prof. Dr.-Ing. Andreas Wehrenpfennig
Zweitprüfer:	Dipl.-Inform. Jörg Schäfer
Eingereicht am:	31.08.2015
URN:	urn:nbn:de:gbv:519-thesis2015-0633-6

Danksagung

Für die Unterstützung dieser Bachelorarbeit möchte ich mich zunächst bei einigen Personen bedanken.

Ich bedanke mich zuerst einmal bei Prof. Dr.-Ing. Andreas Wehrenpfennig und Dipl.-Inform. Jörg Schäfer für die Betreuung meiner Bachelorarbeit.
Weiterhin bedanke ich mich bei allen Personen, die mich mit Korrekturlesen unterstützt haben.
Ein besonderer Dank gilt meiner Familie, die mich während der Bearbeitungszeit unterstützt haben.

Kurzfassung

Diese Arbeit beschäftigt sich mit dem Aufbau und Vergleich der Suchmaschinen. Dadurch soll ein Einblick auf die Funktionsweise unterschiedlicher Suchmaschinen geschaffen und durch einen Vergleich der Suchergebnisse Rückschlüsse auf die Qualität der Suchmaschine gezogen werden. Durch Marktanalysen werden außerdem die beliebtesten Suchmaschinen der Nutzer dargestellt und daraus mögliche Trends für die Entwicklung des Suchmaschinenmarktes gezogen. Ein weiterer Teil der Arbeit beschäftigt sich mit dem Aufbau einer eigenen Suchmaschine, um nach eigenen Webseiten suchen zu können.

Abstract

The following thesis deals with the structure and comparison of the described search engines, thereby giving an insight into the functioning of different search engines and drawing conclusions about the quality of the services by comparing their respective search results. Through market analysis, the search engines are additionally ranked by its user popularity which allows a further outlook on development trends of the search engine market. Another chapter of the thesis addresses the setup of an own search engine to search for personal websites.

Inhaltsverzeichnis

1 Einführung

Heutzutage gibt es viele Wege zur Informationsbeschaffung. Das Internet ist eines der wichtigsten davon. Dieses Netz von Informationen ist dezentral organisiert. Durch die große weltweite Vernetzung der Informationen kann jeder Teilnehmer auf diese Informationen zugreifen (vgl. [ITW151]).

1.1 Problemstellung

Das Problem bei der Informationsbeschaffung liegt beim Suchen. Der Nutzer muss die Informationen erst einmal finden können. Als das Internet aufkam, gab es dafür sogenannte Webverzeichnisse. Diese beinhalteten Verweise zu den Informationen und wurden kategorisch eingeordnet und sortiert. Dies geschah manuell durch einen Betreiber. Als die Informationen im Internet exponentiell zunahmen, kamen die Betreiber der Webverzeichnisse nicht mehr hinterher. Zu viele neue Webpräsenzen mit Informationen entstanden. Dieses Problem löste nur die automatisierte Suchmaschine (vgl. [Dir15], S.23-24).

Praktisch jeder Mensch, der im Internet nach Informationen sucht, nutzt dafür heutzutage eine Suchmaschine. Der Markt ist geprägt von vielen verschiedenen webbasierten Suchmaschinen mit unterschiedlichen Ansätzen. Jede dieser Suchmaschinen liefert unterschiedlich qualitative Ergebnisse. Dennoch setzen viele Nutzer größtenteils auf eine einzige Suchmaschine.

1.2 Ziel der Bachelorarbeit

Ziel dieser Arbeit ist es einen Einblick in die unterschiedlichen Suchmaschinen zu bekommen und diese miteinander zu vergleichen. Dabei wird primär auf den Aufbau und die Funktionsweise einer webbasierten Suchmaschine mit HTML-Dokumenten eingegangen. Webbasiert bedeutet, dass diese Suchmaschinen auf dem „World Wide Web“ beruhen und diese als Webanwendung nutzbar sind. Außerdem werden die Suchmaschinen dargestellt, klassifiziert und verglichen. Zudem werden Algorithmen vorgestellt, die bei Suchmaschinen zum Einsatz kommen und verantwortlich sind für die unterschiedlichen Ergebnisse. Anschließend werden Methoden zur gezielten Suche gezeigt. Die beliebtesten Suchmaschinen auf dem Markt werden durch Marktanteile analysiert und dargestellt. Daraus werden Rückschritte auf mögliche Trends des Suchmaschinenmarktes für die Zukunft gezogen. Mit frei verfügbarer Open-Source-Software werden dann zwei Suchmaschinen aufgebaut, um nach Webseiten suchen zu können.

2 Theorie einer Suchmaschine

Dieses Kapitel beschäftigt sich zunächst mit den Grundlagen einer Suchmaschine. Hierzu werden die Aufgaben und Anforderungen an eine Suchmaschine, und wie diese prinzipiell aufgebaut ist, untersucht. Außerdem wird die Funktionsweise mithilfe eines „UML-Aktivitätsdiagramms“ dargestellt. Weiterhin werden noch Richtlinien für die Suchmaschinen beschrieben. Zum Schluss wird noch auf die Probleme eingegangen, die Suchmaschinen beim Erfassen von Webseiten haben.

2.1 Prinzip

Vorgänger von Suchmaschinen waren Webverzeichnisse, deren Einträge manuell geprüft wurden. Diese Art der Katalogisierung war sehr kostenintensiv. Die Betreiber der Webverzeichnisse kamen mit der Pflege, aufgrund der Vielzahl von neuen Webseiten, nicht hinterher. Aus diesem Grund wurde nach einer alternativen Möglichkeit gesucht, damit Webseiten automatisch in den Index aufgenommen werden. Durch diese Entwicklung entstand die Suchmaschine. Eine Suchmaschine besteht prinzipiell aus einer Webanwendung, die im Hintergrund selbstständig nach Webseiten im Web sucht und diese Inhalte automatisch katalogisiert. Je nach Eingabe des Suchbegriffes werden die Inhalte nach Relevanz sortiert und in einer Ergebnisliste ausgegeben (vgl. [Enr15]).

2.2 Anforderungen

Eine Suchmaschine muss bei einer Suchanfrage sehr schnell antworten und die dazugehörigen Webseiten, nach einer gewissen Relevanz sortiert, wiedergeben. Diese Ergebnisse sollten thematisch zu der Eingabe passen und keine Links zu einer nicht mehr erreichbaren Webseite enthalten. Weiterhin gehören zu den Anforderungen eine einfache Bedienung, Benutzerfreundlichkeit und eine geringe Netzwerkbelastung. Von Suchmaschinen wird erwartet, dass diese das gesamte Internet durchsuchen. Praktisch ist dies jedoch unmöglich zu realisieren. Das liegt einerseits an der enormen Größe von Informationen und Daten im Netz und andererseits an dem großen Arbeitsaufwand der Informationsdurchsuchung, der betrieben werden muss.

Mit dem technologischen Fortschritt nehmen auch die Erwartungen an einer Suchmaschine zu. Eine weitere Anforderung an Suchmaschinen ist, dass diese sich in einer kontinuierlichen Weiterentwicklung durch neue Aufgaben und Herausforderungen befinden sollen (vgl. [Onl15]).

2.3 Aufgaben

Die Aufgaben einer Suchmaschine kann in vier Segmente gegliedert werden:

- Suchen und Indexieren von Webseiten
- Katalogisieren und Bewerten der Inhalte
- Speichern der Informationen
- Generierung von Ergebnislisten anhand der gesammelten und aufbereiteten Informationen (vgl. [Enr15])

2.4 Aufbau

Für jede der in Kapitel 2.3 dargestellten Aufgaben gibt es eine zuständige Komponente. Das Komponentendiagramm in der Abb. 2.1 zeigt die vier wesentlichen Komponenten (vgl. [Enr15]):

- Der Crawler zur Erfassung der Webseiten und Verfolgung der Hyperlinks
- Der Indexer für die Aufbereitung und Bewertung der Informationen
- Die Datenbank für die Speicherung der Informationen
- Der Searcher, der die Suchergebnisse passend zur Anfrage liefert

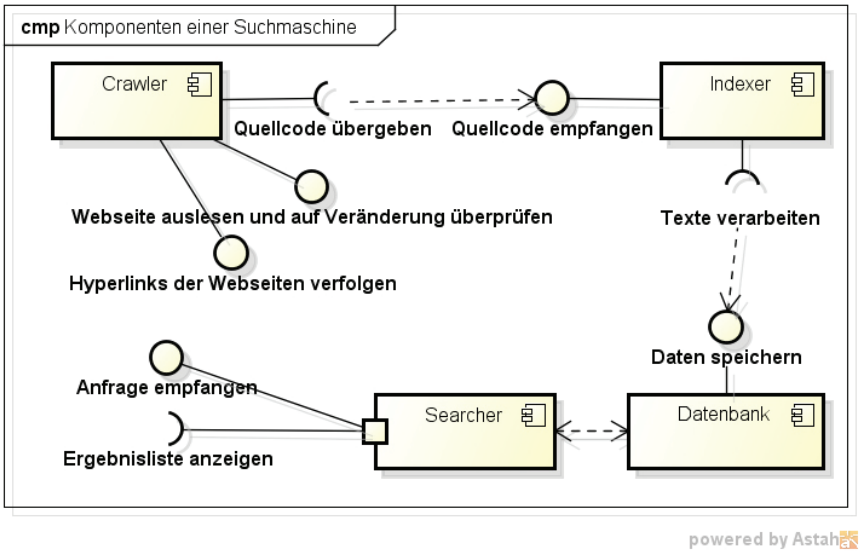


Abb. 2.1: Komponentendiagramm einer Suchmaschine

2.4.1 Der Crawler

Der Crawler (auch Spider genannt) ist die suchende Komponente einer Suchmaschine und hat die Aufgabe Dokumente zu suchen. Um Dokumente auffinden zu können, benötigt der Crawler zunächst einmal eine Ausgangmenge bekannter Webseiten. Die sogenannten „seed pages“ oder „seed set“. Im nächsten Schritt werden diese bekannten Webseiten besucht, die Dokumente indexiert und alle Hyperlinks im Dokument wiederum verfolgt. Mit dieser Methode sollen dann alle im Web vorhandenen Dokumente gefunden werden. Allerdings können sich die Dokumente der Webseiten im Laufe der Zeit verändern. Deshalb wiederholt sich der Crawling-Vorgang ständig und muss alle Dokumente, die gelöscht oder verändert wurden,

überprüfen. Erfolgt diese Überprüfung nicht, können Treffer auf bereits gelöschte Dokumente führen. Verändert eine Webseite ihren Inhalt regelmäßig, so wird der Crawler diese Webseite öfters besuchen als eine Webseite, bei der sich kaum etwas verändert.
Die Webseiten im Internet sind sehr verstreut und nicht jede Webseite führt unmittelbar zu einer anderen Webseite. Je mehr unterschiedliche Webseiten in der Ausgangsmenge vorhanden sind, desto mehr können auch gefunden werden. Durch diese Methode können allerdings nicht alle Webseiten im Internet entdeckt werden. So existieren weiterhin noch Webpräsenzen, die nicht verlinkt worden sind. Suchmaschinen, die über einen längeren Zeitraum existieren, haben den Vorteil, dass diese auch Dokumente finden, die nicht mehr mit dem Crawling-Verfahren gefunden werden können (vgl. [Dir15], S.37-40).

2.4.2 Der Indexer

Der Indexer ist die aufbereitende Komponente einer Suchmaschine und hat die Aufgabe die vom Crawler gelieferten Dokumente so zu zerlegen und aufzubereiten, damit diese für die Suche effizient genutzt werden können. Die gefundenen Dokumente werden in indexierbare Einheiten zerlegt, wie zum Beispiel in einzelne Wörter, Wortstämme oder N-Gramme. N-Gramme stellen Zeichenfolgen dar, die sich im Text wiederholen. Das N steht für die Anzahl der Buchstaben. Ein 1-Gramm teilt zum Beispiel den Text in einzelne Buchstaben ein (vgl. [Brä15]).
Weiterhin wird deren Vorkommen innerhalb des Dokuments verzeichnet. Dieser Prozess erstellt eine Repräsentation der Dokumente (Index) und hilft der Suchmaschine dabei möglichst schnell die Dokumente durchsuchen zu können. Der Index, der dadurch entsteht, nennt sich invertierter Index. Mit dem Algorithmus des invertierten Index können die Daten in einer geeigneten Datenstruktur abgelegt werden. Dieser Algorithmus wird näher in Kapitel 4.1 beschrieben. Da jede Suchmaschine auf unterschiedliche Dokumente im Crawling-Prozess zugreift, entsteht für jede Suchmaschine ein eigener repräsentativer Suchindex. Das hat zusätzlich wiederum Auswirkungen auf die Suche. Suchbegriffe, die bei der Repräsentation nicht erfasst wurden, werden somit auch bei der Suche nicht angezeigt. Wenn so zum Beispiel der Autor eines Dokumentes nicht erfasst wird, dann kann später die Suche auch nicht auf den Autor eingeschränkt werden (vgl. [Dir15], S.48-53).

2.4.3 Die Datenbank

Die Datenbank ist die Komponente, in der die erfassten und aufbereiteten Informationen stehen. Bei der Erfassung der Informationen entsteht eine große Menge an Daten. Um diese geeignet abzulegen zu können, werden für diesen Zweck NoSQL-Datenbanken eingesetzt. Der Begriff „NoSQL“ steht für die Bezeichnung „Not only Structured Query Language“. Für verschiedene Einsatzzwecke gibt es eine spezielle NoSQL-Datenbank, die dafür optimiert wurde, zum Beispiel Graphendatenbanken, dokumentenorientierte Datenbanken und Key-

Value-Datenbanken. Viele NoSQL-Datenbanken stehen als Open-Source-Software zur Verfügung, dies betrifft die Beispiele HBase, MongoDB, Cassandra und CouchDB (vgl. [Die15]), (vgl. [Joo15]).

Relationale Datenbanken eignen sich für die Speicherung der Daten nicht, da diese Leistungsprobleme bei großen Datenmengen bekommen können. NoSQL-Datenbanken bieten für den speziellen Anwendungszweck deutlich mehr Vorteile. Die Vor- und Nachteile der zwei Datenbanksysteme sind in der Tabelle 2.1 dargestellt.

Datenbank	Vorteile	Nachteile
Relationale Datenbanken	+ Unterstützung von Transaktionen + Verwaltung der Daten in Tabellen + Verknüpfung der Tabellen durch Fremdschlüssel + Einfache Umsetzung	– Leistungsprobleme bei großen Datenmengen – Festes Schema von Anfang an – Nur vertikale Skalierbarkeit durch neuen Speicher
NoSQL-Datenbanken	+ Schneller Zugriff + Schemalos + Horizontale Skalierbarkeit durch neue Server + Verteilte Systeme + Hohe Ausfallsicherheit + Verarbeitung riesiger Datenmengen	– Daten können für kurze Zeit inkonsistent sein – keine bis schwach ausgeprägte Transaktionskonzepte

Tabelle 2.1: Vergleich von rationalen Datenbanken und NoSQL-Datenbanken

2.4.4 Der Searcher

Der Searcher bildet die Verbindung zwischen den Benutzereingaben und der Suchmaschine. Der Nutzer gibt im Suchfeld seine Stichwörter für die Suche ein und enthält kurz darauf eine Trefferliste mit allen relevanten Dokumenten. Der Prozess, der dabei ausgeführt wird, ist ein Abgleich der Suchbegriffe mit dem aufbereiteten Datenbestand der Suchmaschine. Die Treffer werden dann nach der Relevanz sortiert und ausgegeben. Der Algorithmus dahinter ist von Suchmaschine zu Suchmaschine unterschiedlich. Meistens wird der Algorithmus von den Betreibenden der Suchmaschinen streng geheim gehalten, um das Ausnutzen der Algorithmen zu verhindern (vgl. [Dir15], S.58-61).

2.5 Funktionsweise

Das UML-Diagramm in der Abb. 2.2 soll den gesamten Vorgang einer Suchmaschine verdeutlichen. Der Startpunkt der Suchmaschine beginnt mit einer Ausgangsmenge bekannter Webseiten. Diese Webseiten durchsucht der Crawler und liest den Quellcode der Webseiten ein. Anschließend verzweigt sich der Vorgang. Der Quellcode wird dann dem Indexer übergeben und dieser wird danach in einzelne Bestandteile zerlegt. Der Indexer wird zunächst die Daten aufbereiten und diese in einer Datenbank ablegen. Aber auch werden die erkannten Hyperlinks auf der Webseite in eine Queue (Warteschlange) des Crawlers übergeben. Dieser Vorgang wiederholt sich kontinuierlich, um weitere Webseiten für den Crawler-Vorgang zu finden. Der Benutzer gibt nun seine Suchbegriffe wie zum Beispiel „Hochschule Neubrandenburg“ in das Suchfeld der Suchmaschine ein. Der Searcher kommuniziert mit der Datenbank und gibt dem Benutzer eine Ergebnisliste aller relevanten Dokumente aus. Auf eines der Ergebnisse kann der Benutzer klicken, um auf eine andere Webseite zu gelangen. Zum Schluss wird dem Benutzer die angeklickte Webseite angezeigt (vgl. [Dir15], S.37 ff.).

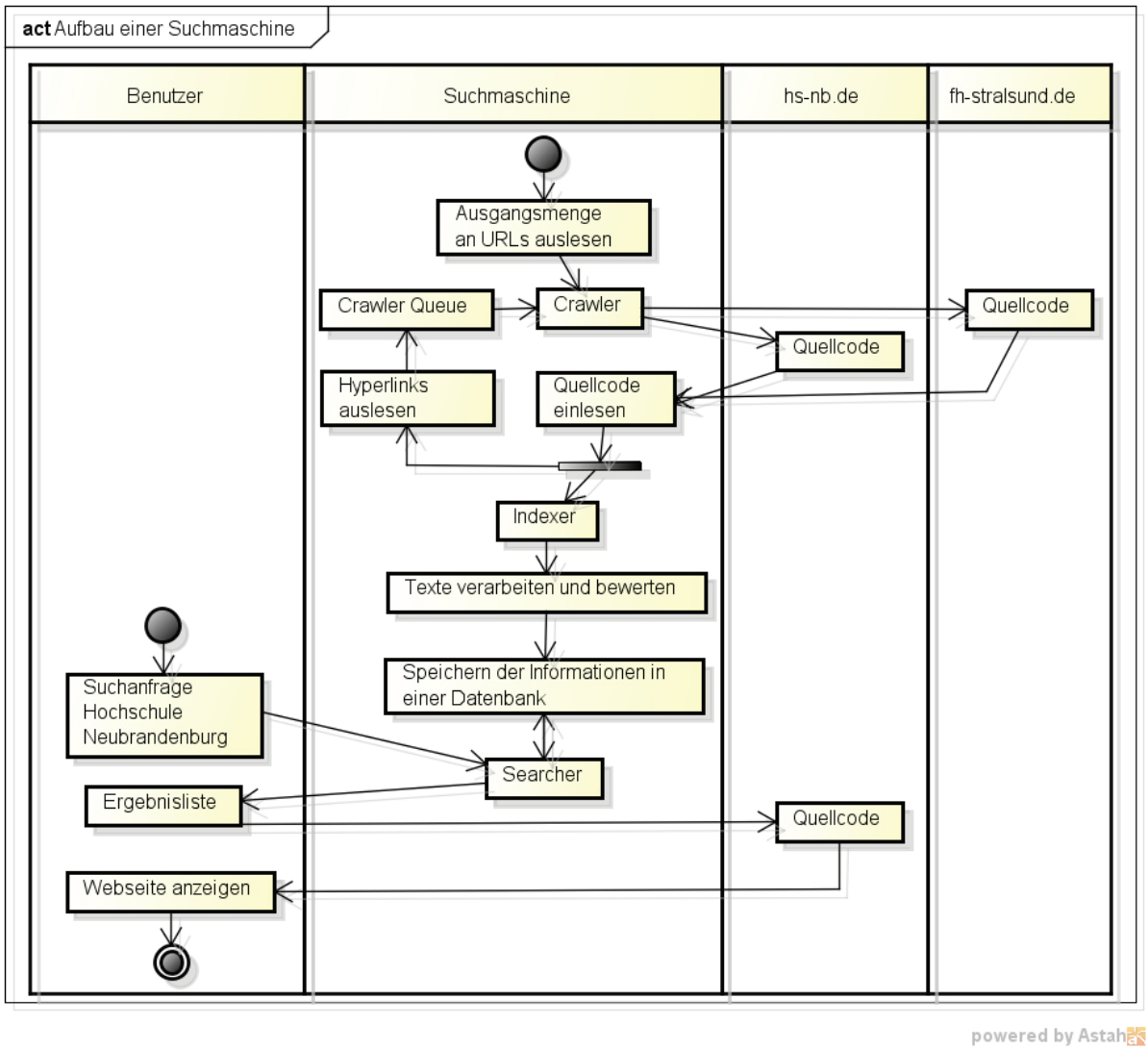


Abb. 2.2: Aktivitätsdiagramm einer Suchmaschine

2.6 Ranking

Suchmaschinen finden bei einer Suchanfrage sehr viele relevante Dokumente. Jedoch ist es schwer allen Dokumenten zu folgen. Meistens werden nur die ersten paar Seiten einer Suchanfrage genutzt. Besonders die Treffer auf den letzten Seiten werden kaum in Betracht gezogen. Um dies zu unterbinden, ist es sinnvoll diese Treffer nach einer geeigneten Relevanz zu sortieren. Dadurch können die relevanten Suchergebnisse direkt auf den ersten Seiten angezeigt werden. Mehrere unterschiedliche Algorithmen und Faktoren nehmen Einfluss auf das Ranking eines Dokumentes. Dabei gilt, je höher der Rank einer Seite, desto höher wird das entsprechende Dokument in der Ergebnisliste angezeigt. Jede Suchmaschine kann diesen Algorithmus mit unterschiedlichen Faktoren individuell gestalten. Der Algorithmus wird bei den meisten Suchmaschinen sehr diskret behandelt und ist der Kern einer heutigen Suchmaschine, der diese einzigartig macht (vgl. [Dir15], S.89-91).

2.7 Suchmaschinen steuern und ausschließen

Der in Kapitel 2.4.1 beschriebene Crawler findet auch Webseiten, die möglicherweise von Webseitenbetreibenden nicht gefunden werden möchten. Um dieses Problem zu lösen, gibt es die Konvention „Robots Exclusion Standard“, an der sich die Suchmaschinen halten können. Dieser „Robots Exclusion Standard“ ermöglicht die Steuerung der Crawler und die Indexierung durch akzeptierte Befehle. Es gibt zwei Möglichkeiten die Suchmaschinen dadurch zu steuern. Einerseits durch „Metadaten“ in jedem Dokument und andererseits mit einer Datei namens „robots.txt“ (vgl. [Dir15], S.41-44).

2.7.1 Metadaten

Metadaten sind Informationen in einem Dokument, die bei einem Aufruf nicht direkt sichtbar sein müssen. Zum Beispiel kann in das Meta-Tag „description“ eine kurze Beschreibung des Dokumentinhalts erfolgen. Die Suchmaschine nutzt dieses Meta-Tag um Beschreibungen zu erstellen. Die Metadaten gelten jedoch nur für das jeweilige Dokument. Und müssen deshalb für jedes Dokument angepasst werden (vgl. [Dir15], S.41-44).

2.7.2 Robots.txt

Eine weitere Methode zur Steuerung ist die „robots.txt“-Datei, die sich im obersten Verzeichnis (Wurzelverzeichnis) einer Webpräsenz befinden muss. Diese Datei enthält Anweisungen, mit denen Webseiten von dem Crawler-Prozess ausgeschlossen werden können. Wie in Abb. 2.3 dargestellt, können durch die Angabe von „Disallow“ einzelne Webseiten ausgeschlossen oder mit „Allow“ für den Crawler-Prozess erlaubt werden. Auch Sitemaps werden für eine gezielte Steuerung der Crawler unterstützt. Diese Sitemaps enthalten eine vollständige Auflistung aller Seiten einer Webpräsenz und können dadurch eine vollständige Abdeckung einer großen Webpräsenz gewährleisten. Diese Anweisungen können auch eingeschränkt für einzelne Suchmaschinen oder für alle Suchmaschinen gelten. Wie in der Abb.

2.4 dargestellt, erfolgt die Angabe über einen bestimmten Crawler mit dem Befehl „User-agent“, gefolgt von dem Namen des Crawlers. Statt den Namen des Crawlers kann auch der Stern-Platzhalter verwendet werden. Dieser sagt aus, dass die nachfolgenden Anweisungen für alle Suchmaschinen gelten (vgl. [Dir15], S.41-44).

```
User-agent: *
Disallow: /search
Disallow: /sdch
Disallow: /groups
Disallow: /images
Disallow: /catalogs
Allow: /catalogs/about
Allow: /catalogs/p?
Disallow: /catalogues
Sitemap: http://www.gstatic.com/culturalinstitute/sitemaps/www_google_com_culturalinstitute/sitemap-index.xml
Sitemap: https://www.google.com/edu/sitemap.xml
Sitemap: https://www.google.com/work/sitemap.xml
Sitemap: http://www.google.com/hostednews/sitemap_index.xml
Sitemap: http://www.google.com/maps/views/sitemap.xml
Sitemap: http://www.google.com/sitemaps_webmasters.xml
Sitemap: http://www.google.com/ventures/sitemap_ventures.xml
```

Abb. 2.3: Eine robots.txt-Datei mit Sitemaps (übernommen aus http://google.de/robots.txt)

```
User-agent: msnbot-media
Disallow: /
Allow: /shopping/$
Allow: /shopping$
Allow: /th?

User-agent: Twitterbot
Disallow:

User-agent: *
Disallow: /account/
Disallow: /bfp/search
Disallow: /blogs/search/
Disallow: /entities/search
```

Abb. 2.4: Eine robots.txt-Datei mit dem Ausschluss von bestimmten Crawlern (übernommen aus http://bing.com/robots.txt)

2.7.3 Robots-Richtlinien

Suchmaschinen im Internet benötigen spezielle Richtlinien. Die Initiative der drei größten Suchmaschinen von Google, Microsoft und Yahoo bringen mehr Klarheit. Diese legen gemeinsame Regeln für die Datei „robots.txt“ und den „Robots-Meta-Tags“ fest. Die drei Suchmaschinen beziehen sich auf das existierende Robots Exclusion Protocol (REP) aus dem Jahr 1994. Suchmaschinen haben eigene Tags entwickelt, mit denen Crawler gesteuert werden. Diese Gemeinsamkeiten haben Google, Microsoft und Yahoo bekannt gegeben. In der folgenden Tabelle 2.2 sind die Richtlinien für den Einsatz in der Datei „robots.txt“ zusammengetragen. Und in der darauffolgenden Tabelle 2.3 sind die Richtlinien für den Einsatz der HTML Meta-Richtlinien aufgeführt. Diese gelten für die 3 größten Suchmaschinen von Google, Microsoft und Yahoo (vgl. [Kla15]).

Richtlinie	Wirkung	Einsatzmöglichkeiten	Beispiele
Disallow	Der Crawler soll die angegebene Seite nicht indexieren.	Diese Anweisung schützt bestimmte Webseiten oder Pfade davor gecrawlt zu werden.	`Disallow: /` Gesamte Webpräsenz soll nicht gecrawlt werden.
Allow	Der Crawler soll die angegebene Seite indexieren.	Nützlich ist Allow im Zusammenhang mit Disallow, sodass große Teile gesperrt werden können, außer kleine Teile darin.	`Disallow: /verzeichnis/` `Allow: /verzeichnis/datei.htm`
$ Wildcard Unterstützung	Bezieht sich auf das Ende einer URL.	Dateien mit einem bestimmten Muster oder einem bestimmten Datentyps.	`Disallow: /*.pdf$` Dateien mit der Endung .pdf sollen nicht gecrawlt werden.
*** Asterisk-Wildcard Unterstützung**	Gibt dem Crawler an, dass dieser nach einer Sequenz von Zeichen suchen soll.	Webseiten mit bestimmten Mustern, z.B. überflüssigen Parametern oder Session-IDs in der URL.	`Disallow: /geheim*/` Verzeichnisse, die mit geheim beginnen, werden nicht indexiert.
Sitemaps Location	Gibt dem Crawler an, wo Sitemaps zu finden sind.	Verweisen auf Orte, die Feeds beinhalten, um Crawlern zu helfen diese zu finden.	`Sitemap: http://www.meine-web-site.de/sidemap.xml`

Tabelle 2.2: Suchmaschinen Richtlinie für den Einsatz der Datei robots.txt (vgl. [Kla15])

Richtlinie	Wirkung	Einsatzmöglichkeiten	Beispiele
NOINDEX META Tag	Der Crawler soll eine bestimmte Seite nicht indexieren.	Die gecrawlte Seite soll nicht in den Index aufgenommen werden.	`<meta name="robots" content="noindex">`
NOFOLLOW META Tag	Der Crawler soll dem angegebenen Link zu einer bestimmten Seite nicht folgen.	Schützt öffentliche Bereiche vor dem Crawler, sodass Links auf der Seite nicht verfolgt werden.	`<meta name="robots" content="nofollow">`
NOSNIPPET META Tag	Der Crawler soll Snippets[1] für bestimmte Seiten nicht in den Suchergebnissen anzeigen.	Kein Snippet soll für eine Seite in den Suchergebnissen angezeigt werden.	`<meta name="robots" content="nosnippet">`
NOARCHIVE META Tag	Gibt an, dass die Webseite in der Suchmaschine kein „Im Cache" Link anzeigen soll.	Im Cache der Suchmaschine soll keine Kopie der Webseite zur Verfügung gestellt werden.	`<meta name="robots" content="noarchive">`
NOODP META Tag	Der Crawler soll für eine bestimmte Seite nicht den Titel und das Snippet des Open Directory Projekts[2] verwenden.	Für eine bestimmte Seite nicht den Titel und das Snippet aus dem Open Directory Projekts verwenden.	`<meta name="robots" content="noodp">`

Tabelle 2.3: Einsatz der Suchmaschinen HTML Meta-Richtlinie (vgl. [Kla15])

[1] Snippet ist ein kurzer Textauszug aus einer Webseite zum Anzeigen in der Ergebnisliste einer Suchmaschine
[2] Open Directory Projekt (ODP) ist das größte von Menschen gepflegte Webverzeichnis des Internets.

2.8 Probleme

Suchmaschinen können bei der Erfassung der Webseiten mit Hindernissen konfrontiert werden. Der Crawler einer Suchmaschine stößt dabei gleich auf mehrere Probleme. Das erste Problem ist, dass es im Internet viele gleiche Inhalte unter verschiedenen Adressen gibt, die sogenannten Dubletten. Diese können auch innerhalb und außerhalb der gleichen Webseite vorkommen. Innerhalb zum Beispiel durch eine Benutzer- und einer Druckansicht und außerhalb durch Kopien gleicher Inhalte. Diese Dubletten müssen aus Kosten von Zeit, Rechenleistung und Geld möglichst vermieden werden. Deswegen ist es wichtig bereits im Crawling-Prozess das Original zu erkennen. Ein zweites Problem sind Fallen (sog. Spider Traps). Diese entstehen durch automatisch generierte Inhalte, bei denen es durch Blättern in unendliche Tiefen gehen kann. Ein Beispiel dafür sind automatisch erstellte Kalender im Web. Bei diesen Kalendern kann unendlich weit vor- und zurückgeblättert werden. Der Crawler würde nun jeden Link verfolgen und somit auf dieser Webseite hängen bleiben. Ein weiteres Problem sind Webseiten ohne relevanten Inhalt, die nur dafür da sind Nutzer auf eine Webseite zu locken und Werbung zu schalten. Diese Spam-Inhalte sind zu einem großen Teil im Internet vorhanden und sollen vom Crawling-Prozess ausgeschlossen werden. Der Grund dafür liegt auch hier bei den beschränkten Ressourcen einer Suchmaschine. Denn der Index hat eine Grenze und jedes Spam-Dokument nimmt den Platz für ein relevantes Dokument weg.

Ein weiteres Problem betrifft die Webseitenbetreibenden. Diese können Methoden zum Steuern von Web-Crawlern einsetzen. Aber diese bieten keine Garantie dafür, dass dennoch ein Crawler die Webseiten indexiert. Daher sollten Webseitenbetreiber private Inhalte schützen oder erst gar nicht im Internet offen legen (vgl. [Dir15], S.29 ff.).

3 Arten von Suchmaschinen

Dieses Kapitel wendet sich den verschiedenen Arten der Suchmaschinen zu. Suchmaschinen lassen sich in verschiedene Arten einteilen. Das sind Volltextsuchmaschinen, Spezialsuchmaschinen, Metasuchmaschinen, Hybridsuchmaschinen und dezentrale Suchmaschinen. Weiterhin wird in diesem Kapitel noch auf die Vorgänger der Suchmaschinen eingegangen. Diese sind nicht direkt eine Suchmaschine und weichen vom Prinzip einer Suchmaschine stark ab, werden aber zum Verständnis und aufgrund der Alternative zur klassischen Suchmaschine hier erwähnt.

3.1 Volltextsuchmaschinen

Volltextsuchmaschinen sind vollautomatisch und nutzen Crawler zum Indexieren der Webseiten. Diese „hangeln“ sich von Webseite zu Webseite anhand der Hyperlinks und versuchen das Web in der Breite zu durchsuchen. Anschließend werden die Webseiten im eigenen Datenbestand aufgenommen und für die spätere Suchanfrage aufbereitet. Diese Art der Suchmaschine ist eigenständig und besitzt ihren eigenen Index. Der Aufbau und die Funktionsweise solcher Volltextsuchmaschinen wurden in Kapitel 2 bereits näher abgehandelt.

3.2 Spezialsuchmaschinen

Spezialsuchmaschinen sind thematisch beschränkte Suchmaschinen und dienen dazu bestimmte zielgenaue Recherchen zu ermöglichen. Im Grunde funktionieren diese wie die Volltextsuchmaschinen, allerdings verfolgen diese das Ziel möglichst alle Dokumente einer Webpräsenz zu erfassen. Dadurch ist diese Art der Suchmaschinen größtenteils sehr viel umfangreicher. Außerdem kann dadurch noch gezielter mit Begriffen gesucht werden. Weiterhin bieten diese Spezialsuchmaschinen viel mehr Suchmöglichkeiten an. Das Ranking der Suchmaschine wird dementsprechend angepasst, dass nur die relevanten und thematisch passenden Dokumente angezeigt werden. Einige Spezialsuchmaschinen beschränken sich auf bestimmte Bereiche, bei denen bestimmte Quellen angegeben werden. Solche Quellen werden größtenteils von Hand ausgewählt, z.B. bei Nachrichten und wissenschaftlichen Webseiten. Eine Spezialsuchmaschine eignet sich gut bei der Suche eines bestimmten Bereiches (vgl. [Dir15], S.19 ff.), (vgl. [Uni15]).

3.2.1 Archivsuchmaschinen

Archivsuchmaschinen sind ein Sonderfall der Spezialsuchmaschinen. Diese erfassen regelmäßig Inhalte aus dem Internet und machen diese Versionen verfügbar. Alte Versionen werden dabei nicht überschrieben, sondern mit einem Datum versehen. Diese Suchmaschinen erlauben es einen Blick in die Vergangenheit zu werfen und können dabei helfen Dokumente, die im Original nicht mehr verfügbar sind, trotzdem noch aufzurufen. Auch hier tritt das-

selbe Problem wie bei den Volltextsuchmaschinen auf. Archivsuchmaschinen können nicht das ganze Internet abdecken.

Ein Beispiel so einer Archivsuchmaschine ist die in der Abb. 3.1 dargestellte „Wayback Machine“[3]. Diese umfasst bereits über 450 Mrd. Versionen von Dokumenten. Um nach Archiven suchen zu können, wird statt den Suchbegriffen die URL einer Webseite angegeben. Danach erscheint eine grafische Kalenderdarstellung mit den Jahren, Monaten und Tagen zu allen verfügbaren Versionen der Webseite. Nach dem Klick auf eine verfügbare Version wird die Version der Webseite von dem Tag der Indexierung angezeigt (vgl. [Dir15], S.242-243).

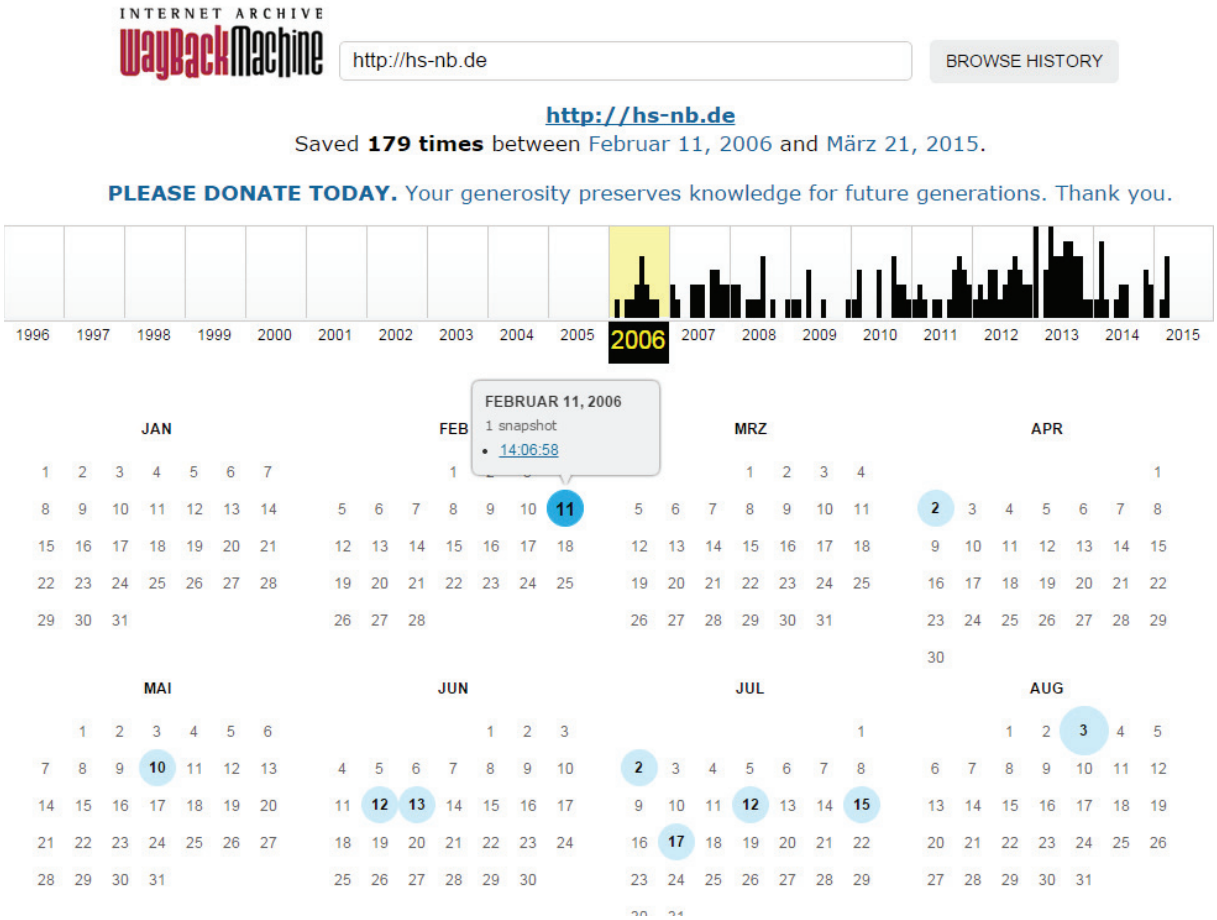


Abb. 3.1: Archivierte Versionen der Hochschule Neubrandenburg auf der Seite „Wayback Machine“

3.3 Metasuchmaschinen

Eine weitere Art von Suchmaschinen sind die Metasuchmaschinen. Diese Metasuchmaschinen sehen auf dem ersten Blick aus wie jede andere Suchmaschine auch, besitzen aber keinen eigenen Datenbestand.

Früher konnten Suchmaschinen das ganze Web nicht vollständig abdecken. Die Idee einer Metasuchmaschine bestand darin, dass die Suchmaschinen, wie in der Abb. 3.2 dargestellt miteinander verknüpft werden. Dazu wird eine Anfrage zu den verschiedenen Suchdiensten gestellt und diese suchen in ihren Datenbanken nach den Suchbegriffen. Die Metasuchma-

[3] Wayback Machine ist unter der Internetpräsenz https://archive.org/web/ erreichbar.

schine sammelt dabei meist nur wenige Treffer der anderen Suchmaschinen. Diese Treffer enthalten auch nur die Beschreibungen und URLs der Suchergebnisse. Die besten Ergebnisse der Suchmaschinen werden dann zusammengefasst und Überschneidungen zwischen gleichen Treffern der verschiedenen Suchmaschinen (Dubletten) werden reduziert. Zum Schluss werden die Treffer nach einer eigenen Relevanz sortiert und in einer eigenen Ergebnisliste ausgegeben. Da viele Suchmaschinen mit einbezogen werden, muss die Metasuchmaschine auf alle dieser Suchmaschinen warten. Die Metasuchmaschine ist deshalb nur so schnell, wie die langsamste Suchmaschine auf die zugegriffen wird. Antwortet eine Suchmaschine nach einer eingestellten Zeit nicht schnell genug, wird diese ausgelassen. Bei Metasuchmaschinen sind standardmäßig viele verschiedene Arten von Suchmaschinen voreingestellt und können bei Bedarf auch manuell ausgewählt werden (vgl. [Dir15], S.21 ff.), (vgl. [Uni15]).

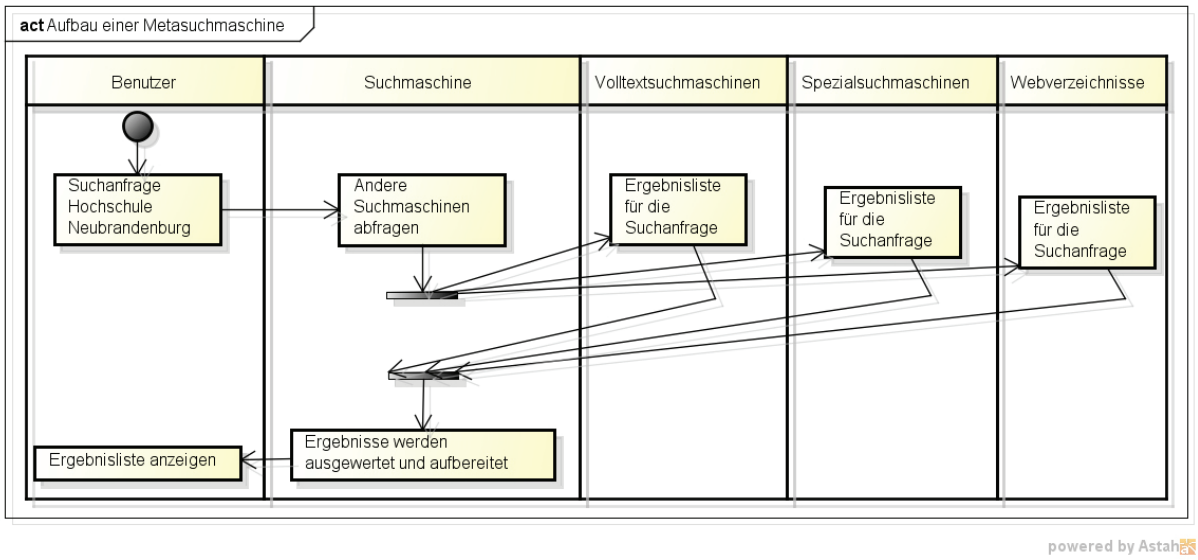


Abb. 3.2: Aktivitätsdiagramm zum Aufbau einer Metasuchmaschine

3.4 Hybridsuchmaschinen

Hybridsuchmaschinen verbinden ausgewählte Inhalte vom Internet mit Inhalten in den eigenen Datenbanken. In den Datenbanken befinden sich meist Inhalte aus dem sogenannten „Deep Web“[4], die von Volltextsuchmaschinen nicht gefunden werden, beispielsweise weil diese Seiten eine Zugangsberechtigung benötigen oder in Datenbanken liegen. Hybridsuchmaschinen können, ähnlich wie Metasuchmaschinen, aus einer Kombination von mehreren Suchdiensten bestehen. Zum Beispiel Volltext-, Spezial-, Metasuchmaschinen und Webverzeichnissen (vgl. [Dir15], S.21), (vgl. [Sch15]).

[4] Deep Web bezeichnet das Versteckte Web, das für Suchmaschinen nicht zugänglich ist.

3.5 Dezentrale Suchmaschinen

Die Projekte Faroo und YaCy[5] nutzen vergleichsweise einen ganz anderen Ansatz und fungieren als dezentrale Suchmaschine. Jeder Nutzer kann seine eigene Suchmaschine installieren und konfigurieren. Dabei wird kein zentraler Server genutzt, sondern mit einer Peer-to-Peer Verbindung werden die Suchergebnisse der einzelnen Knotenpunkte anderer Nutzer ausgetauscht. Jeder Nutzer ist somit Teil eines großen Suchnetzwerks. Die Nutzer entscheiden dabei, welche Inhalte aufgelistet werden und in welcher Reihenfolge diese Ergebnisse erscheinen. Der integrierte Web-Crawler erfasst die Daten, parst und speichert diese zunächst lokal im Suchindex. Bei einer Websuche kann nun auf den lokalen und auf den Suchindex von anderen Peers im Netzwerk gesucht werden. Um den Datenschutz zu gewährleisten, werden die versendeten Suchbegriffe vor dem Absenden verschlüsselt (vgl. [YaC15]), (vgl. [Gol15]), (vgl. [Taz15]).

3.6 Webverzeichnisse

Die Vorgänger der Suchmaschinen sind die Webverzeichnisse (auch Webkataloge genannt). Diese Webverzeichnisse sind keine Suchmaschinen, werden aber aufgrund der Vollständigkeit und des späteren Vergleichs hier erwähnt. Außerdem greifen Meta- und Hybridsuchmaschinen noch auf Webverzeichnisse zurück.

Webverzeichnisse bilden ein hierarchisch geordnetes System von eingetragenen Webseiten. Alle Webseiten werden von den Menschen manuell in das System eingepflegt. Das heißt, sie werden beschrieben und danach in das jeweilige Klassifikationssystem eingeteilt. Webverzeichnisse bieten ein Eingabefeld, mit dem Stichwörter eingegeben werden können. Die Stärken der Webverzeichnisse liegen vor allem bei der hierarchischen Anordnung. Diese ermöglicht das Navigieren vom Allgemeinen zum Speziellen.

Webverzeichnisse spielen heutzutage fast keine Rolle mehr, denn der Ansatz der Suchmaschinen hat diese größtenteils verdrängt. Dennoch gibt es viele Webverzeichnisse im Internet. Das größte von Menschen gepflegte Webverzeichnis im Internet ist das „Open Directory Project“ (vgl. [Dir15], S.23-24).

3.7 Vor- und Nachteile

Zum Schluss des Kapitels werden in der folgenden Tabelle 3.1 die Vor- und Nachteile der beschriebenen Suchmaschinenarten und der Webverzeichnisse dargestellt.

[5] Die YaCy Suchmaschine kann auch direkt im Webbrowser unter folgender URL http://search.yacy.net/ austesten werden.

Art	Vorteile	Nachteile
Volltext-suchma-schinen	+ Riesiger eigener Datenbestand + Aktuelle Inhalte + Volltextsuche + Niedrige Suchzeit + Suchbegriffe werden automatisch vervollständigt + Fehlerhafte Schreibweise wird korrigiert	– Wissenschaftliches geht unter – Suche nach Datum oft fehlerhaft – Aufwendige Pflege der Suchmaschine – Benötigt viel Rechen- und Speicherkapazität – Persönliche Daten der Nutzer können gespeichert werden – Ranking der Suchergebnisse kann manipuliert werden
Spezial-suchma-schinen	+ Nur thematische Bereiche werden durchsucht + Randgebiete besser abgedeckt + Hohe Relevanz der gefundenen Quellen	– Geeignete Suchmaschine schwer aufzufinden – Nur für ein bestimmtes Thema oder Gebiet geeignet – Qualität und Umfang unterschiedlich
Archiv-suchma-schinen	+ Webseiten können datumsbezogen angezeigt werden + Veränderte oder nicht mehr erreichbare Webseiten werden angezeigt + Vergleich von älterer Webseite und heutiger Webseite	– URL einer Webseite wird benötigt – Unregelmäßige und unvollständige Indexierung – Kontroverse Inhalte die nachträglich entfernt wurden können noch angezeigt werden
Mata-suchma-schinen	+ Größere Datenmenge + Einheitliche Präsentation des Suchergebnisses + Identische Treffer werden nur einmal angezeigt	– Liefert nur Trefferauswahl des Suchdienstes zurück – Höhere Zugriffszeit, da viele Suchmaschinen abgefragt werden – Wenige Suchmöglichkeiten – Qualität der Treffer niedriger
Hybrid-suchma-schinen	+ Kombination aus mehreren Arten von Suchmaschinen + Gefundene Seiten sind aktuell + Eigener kleiner Datenbestand	– Schlechtere Qualität, da kein großer Datenindex vorhanden ist
Dezentrale Suchma-schinen	+ Robust gegen Ausfall + Speichert an zentraler Stelle kein Nutzerverhalten + Privatsphäre und Datenschutz ist gewährleistet + Nutzer können Teil des Suchnetzwerkes werden + Konfiguration einer eigenen Suchmaschine	– Kein eindeutiges Ranking – Keine Möglichkeit zur zentralen Zensur – Indexierung und Suchergebnisse hängen stark von der Anzahl der Nutzer ab – Spam könnte sich verbreiten – Zusätzliche Installation einer Software auf dem Computer
Webver-zeichnisse	+ Ergebnisse sind kategorisch und hierarchisch angeordnet + Schnelle Navigation vom Allgemeinen zum Speziellen + Hohe Qualität der Einträge	– Einträge erfolgen nur manuell – Wenig Umfang, kaum aktuelle Inhalte – Nur Stichwortsuche – Sortierung meist nur alphabetisch – Verweisen auf keine einzelne Dokumente, sondern meist nur auf die Startseite einer Webpräsenz – Beschränkt auf bestimmte Themen und Sachgebiete

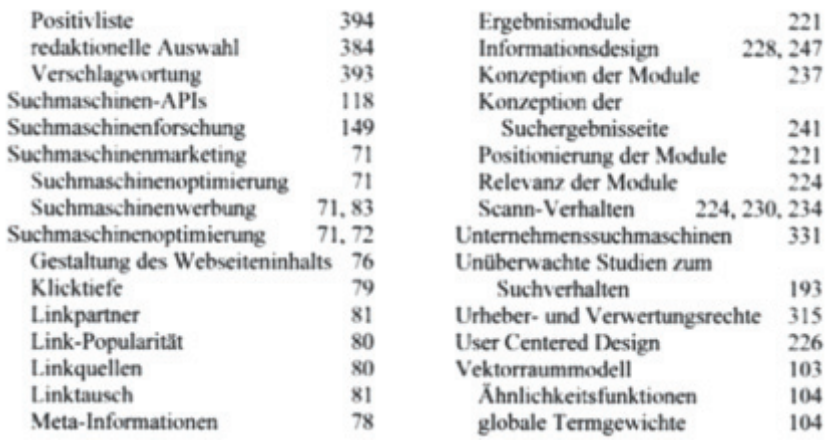
Tabelle 3.1: Vergleich der Suchmaschinen Arten (vgl. [Uni15]), (vgl. [Dir15]), (vgl. [OnP15])

4 Algorithmen

Dieses Kapitel handelt von Algorithmen, die bei einer Suchmaschine angewendet werden. Zunächst werden mehrere Algorithmen beschrieben, die bei einer Suchmaschine angewendet werden. Dazu zählen Algorithmen, die sowohl für das Speichern, als auch für das Verarbeiten der Daten zuständig sind. Zum Schluss werden Algorithmen beschrieben, die das Ranking wesentlich beeinflussen.

4.1 Invertierter Index

Bei dem invertierten Index verweisen einzelne Wörter (Indexeinträge) zu den gesuchten Dokumenten. Dieser Index einer Suchmaschine besteht aus vielen bekannten Einträgen, die zu den Dokumenten verweisen, in dem das Wort vorkommt. Der Algorithmus des invertierten Index ist ähnlich dem eines analogen Buchregisters und lässt sich mit diesem gut vergleichen. Um herauszufinden, an welcher Stelle ein Wort im Buch steht, muss nicht das komplette Werk gelesen werden. Es reicht der Blick in das Buchregister. Wie in der Abb. 4.1 enthält dieses Buchregister die Wörter alphabetisch sortiert und verweist auf die jeweilige Seite im Buch. Der Unterschied zwischen einem invertierten Index und dem Buchregister ist, dass im Buchregister nicht alle Wörter, die im Buch stehen, auch vorkommen, sondern nur die für den Text sinngemäßen.



Positivliste	394	Ergebnismodule	221
redaktionelle Auswahl	384	Informationsdesign	228, 247
Verschlagwortung	393	Konzeption der Module	237
Suchmaschinen-APIs	118	Konzeption der	
Suchmaschinenforschung	149	Suchergebnisseite	241
Suchmaschinenmarketing	71	Positionierung der Module	221
Suchmaschinenoptimierung	71	Relevanz der Module	224
Suchmaschinenwerbung	71, 83	Scann-Verhalten	224, 230, 234
Suchmaschinenoptimierung	71, 72	Unternehmenssuchmaschinen	331
Gestaltung des Webseiteninhalts	76	Unüberwachte Studien zum	
Klicktiefe	79	Suchverhalten	193
Linkpartner	81	Urheber- und Verwertungsrechte	315
Link-Popularität	80	User Centered Design	226
Linkquellen	80	Vektorraummodell	103
Linktausch	81	Ähnlichkeitsfunktionen	104
Meta-Informationen	78	globale Termgewichte	104

Abb. 4.1: Auszug eines Buchregisters (übernommen von [Dir15], S.49)

Da die Masse der Dokumente sehr hoch ist, gibt es beim Indexieren technische Probleme. Um den Zugriff zu beschleunigen und um die kontinuierliche Aktualisierung zu ermöglichen arbeiten Suchmaschinen nicht nur mit einem zentral abgelegten Index, sondern mit mehreren verteilten Indices. Das Crawling und die Indexierung könnten ohne mehrere Indices nicht kontinuierlich aktualisiert werden und damit würden Einträge auf nicht mehr existierende oder geänderte Dokumente verweisen.

Das folgende vereinfachte Beispiel erklärt, wie der Prozess der Indexerstellung erfolgt. Zunächst werden die Dokumente mit dem dazu gehörigen Text aus der Tabelle 4.1 indexiert. Anschließend werden die in den Dokumenten enthaltenden Wörter alphabetisch sortiert und

mit einem Verweis auf das jeweilige Dokument versehen. Die Tabelle 4.2 soll dieses Beispiel veranschaulichen. Wird nun nach dem Wort „Trefferliste“ gesucht, so werden alle Dokumente angegeben, die dieses Wort enthalten. In diesem Beispiel sind das D2 und D5. Auch Kombinationen von Suchbegriffen sind möglich. Dasselbe Beispiel wird nun durch die beiden Suchbegriffe „Suchmaschinen Trefferliste“ ersetzt. Der Suchbegriff „Trefferliste“ kam in D2 und D5 vor und der Suchbegriff „Suchmaschinen“ in D2 und D3. Als Ergebnis werden die Dokumentnummern verglichen und alle Dokumente dargestellt, die gemeinsam enthalten sind. In diesem Beispiel wäre das nur in D2. Mit diesem einfachen Prinzip lassen sich bestimmte Wörter oder Kombinationen von Wörtern in Dokumenten schnell finden.

Dokument	Text
D1	Google ist eine Suchmaschine.
D2	Suchmaschinen erstellen eine Trefferliste.
D3	Suchmaschinen bestehen aus drei Komponenten.
D4	Eine Suchmaschine ist eine Webanwendung.
D5	Die Trefferliste enthält das gesuchte Wort.

Tabelle 4.1: Einfaches Beispiel von Dokumenten mit verschiedenen Inhalt

Term	Dokument
aus	D3
bestehen	D3
das	D5
die	D5
drei	D3
eine	D1; D2; D4
enthält	D5
erstellen	D2
gesuchte	D5
Google	D1
ist	D1; D4
Komponenten	D3
Suchmaschine	D1; D4
Suchmaschinen	D2; D3
Trefferliste	D2; D5
Webanwendung	D4
Wort	D5

Tabelle 4.2: Einfacher Invertierter Index

Mit dem Prinzip des einfachen invertierten Index kann nur festgestellt werden, ob ein Suchbegriff in einem Dokument vorhanden ist. Für komplexere invertierte Indices spielen viele weitere Faktoren wie z.B. Worthäufigkeit und Positionsangeben innerhalb des Dokuments eine wichtige Rolle. In der Tabelle 4.3 ist so ein invertierter Index mit Worthäufigkeiten und Positionsangaben dargestellt. Diese Häufigkeit von Wörtern lässt nun Rückschlüsse auf die Relevanz des Dokumentes zu. Aufgrund der kurzen Beispielsätze kommen die Wörter in diesem Beispiel größtenteils nur einmal vor. Die Verteilung der Häufigkeiten von Wörtern lässt sich bei längeren Dokumenten besser ausnutzen. Ein weiterer wichtiger Faktor für das spätere Ranking der Dokumente ist die Positionsangabe der Wörter im Dokument. Je weiter

ein Wort am Anfang steht, desto relevanter kann dies bei der späteren Suche sein. Das Wort „Trefferliste“ im D2 befindet sich an vierter Stelle und im D5 an zweiter Stelle. Mit der Suche nach dem Wort „Trefferliste“ könnte nun zuerst D5 und danach D2 in der Ergebnisliste der Suchmaschine angezeigt werden, da das Wort im D5 weiter am Anfang steht. Dieses Beispiel zeigt, wie prinzipiell ein Ranking erstellt werden kann. Moderne Ranking-Systeme berücksichtigen noch sehr viele weitere Faktoren (vgl. [Dir15], S48-53).

Term	Worthäufigkeiten	Positionsangaben
aus	D3:1	D3,3
bestehen	D3:1	D3,2
das	D5:1	D5,4
die	D5:1	D5,1
drei	D3:1	D3,4
eine	D1:1; D2:1; D4:2	D1,3; D2,3; D4,4
enthält	D5:1	D5,3
erstellen	D2:1	D2,2
gesuchte	D5:1	D5,5
Google	D1:1	D1,1
ist	D1:1; D4:1	D1,2; D4,3
Komponenten	D3:1	D3,5
Suchmaschine	D1:1; D4:1	D1,4; D4,2
Suchmaschinen	D2:1; D3:1	D2,1; D3,1
Trefferliste	D2:1; D5:1	D2,4; D5,2
Webanwendung	D4:1	D4,5
Wort	D5:1	D5,6

Tabelle 4.3: Invertierter Index mit Worthäufigkeiten und Positionsangaben

4.2 MapReduce

Der MapReduce Algorithmus dient zur Bearbeitung von großen Datenmengen auf verteilten Rechnern. Es wurde erstmals 2003 von Google eingeführt, um Daten mit hoher Flexibilität zu speichern. Dank großen Clustern von Rechnern erreichen Suchmaschinen hohe Performance beim Durchsuchen der Daten und bei der Indexierung. Der MapReduce-Algorithmus hat zwei wesentliche Funktionen. Die Map- und die Reduce-Funktion. Abgearbeitet wird immer zuerst die Map- und danach die Reduce-Funktion. Diese Funktionen werden mehrfach parallel auf zusammengeschalteten Cluster-Rechnern ausgeführt.

In der Abb. 4.2 soll dieses Prinzip verdeutlicht dargestellt werden. Zuerst werden in der Phase 1 die Eingabedaten eingelesen. Diese werden in ungefähr gleich große Blöcke unterteilt. Diese Blöcke werden dann in Phase 2 zu einem Key in der Form (k_1, v_1) zusammengefasst und an verschiedene Cluster zugeteilt. Danach wird die Map-Funktion auf jedem Cluster parallel ausgeführt. Diese Funktion produziert aus dem eingegebenen Wertepaar ein Zwischenergebnis. Wie in der Formel 4.1 dargestellt, besteht diese wiederum aus Paaren der Form $list(k_2, v_2)$. Die Map-Funktion teilt die Sätze in eine Liste von mehreren Wörtern ein. Dadurch kann ein großes Dokument effizient und in minimaler Zeit aufgeteilt werden. Nachdem alle Zwischenergebnisse berechnet sind, werden diese in Phase 3 verteilt. Die Listen in der Form

list(k_2, v_2) werden analysiert und die Einträge mit gleichem Key k_2 werden in dieser Form *(k_2, list(v_2))* zusammengefasst. Diese dient nun als Eingabeparameter für die unterschiedlichen Reduce-Prozesse. In der Phase 4 werden die zusammengefassten Keys komprimiert und in Partitionen zerlegt. Die Reduce-Funktion wird für jeden Key aufgerufen und das endgültige Ergebnis wird berechnet. Dadurch werden die Listen reduziert und der benötigte Speicherplatz wird gering gehalten. Die Phase 5 beschreibt die Datenausgabe. Die Ergebnisse werden hier in einem verteilten Dateisystem, wie das GFS oder HDFS, gespeichert. Diese Daten können dann gemeinsam in eine Datei geschrieben werden.

Dieses Modell beschreibt eines der effizientesten Verfahren zur Verarbeitung von großen Datenmengen. Mit einem minimalen Aufwand lassen sich die vielen Daten effizient und parallel auf vielen Clustern verarbeiten (vgl. [Mar151]).

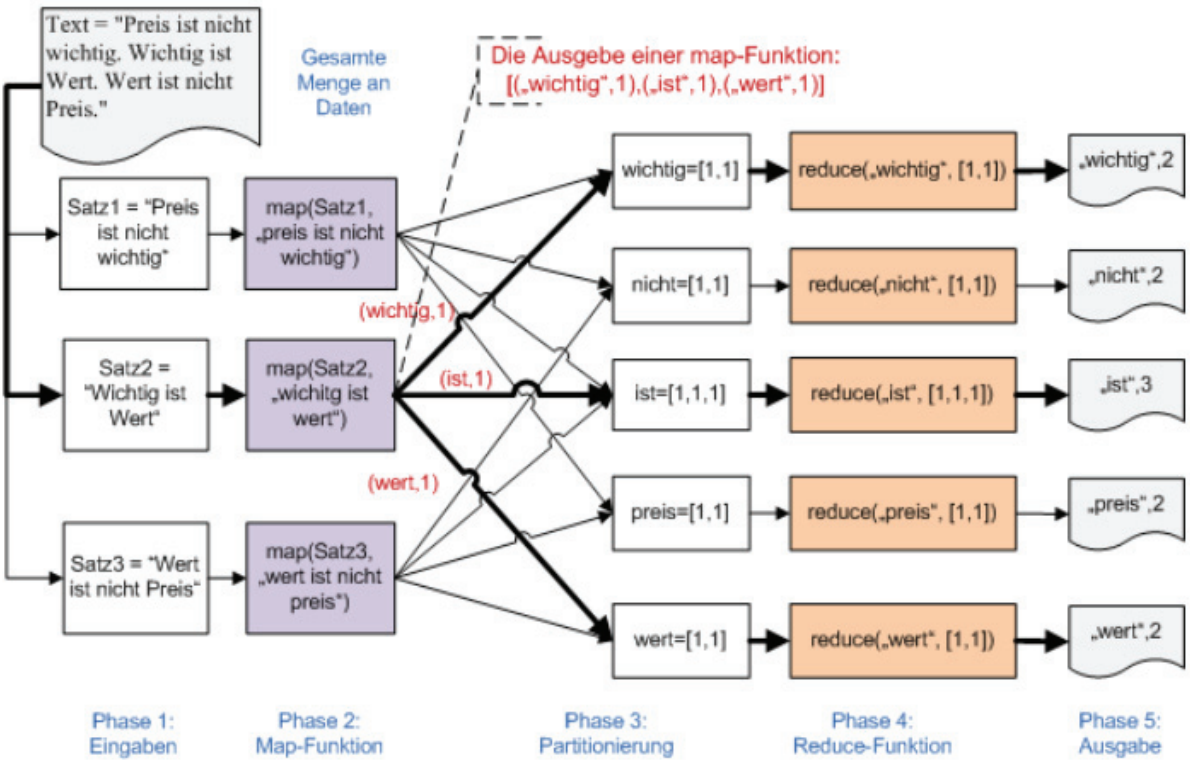


Abb. 4.2: Beispiel für den MapReduce-Algorithmus (übernommen von [Mar151])

$$map : (k_1, v_1) \rightarrow list(k_2, v_2)$$
$$reduce : (k_2, list(v_2)) \rightarrow list(k_3, v_3)$$

Formel 4.1: Darstellung der MapReduce Funktionen (nach [Ber15])

4.3 PageRank

Der bekannteste Algorithmus zum Bewerten von Webseiten ist der PageRank. Dieser wurde an der Stanford University von Larry Page und Sergei Brin entwickelt und im Jahre 1997 patentiert. Der Algorithmus bewertet eine Seite anhand der Beziehungen einzelner Webseiten zueinander. Eine Seite wird umso höher gewichtet, je mehr Hyperlinks auf diese Seite führen. Der Inhalt spielt bei den Webseiten mit diesem Algorithmus keine Rolle. Stattdessen stellt dieser Algorithmus ein Verhältnis zwischen einzelnen Webseiten dar.

Im nachfolgenden wird die Formel 4.2 für den PageRank dargestellt. Die Terme aus der Formel werden in der Tabelle 4.4 erklärt. Der PageRank einer Seite A ergibt sich bei einer rekursiven Berechnung aus den jeweiligen verlinkten Seiten T. Diese Seiten von T fließen nicht gleichmäßig in die Berechnung hinein. Die Anzahl der Links C(T) nimmt Einfluss auf die Gewichtung des PageRanks von der Seite A. Dies ergibt sich aus dem Term PR(T) durch C(T). Alle verlinkten Seiten von T werden addiert. Jede zusätzliche Seite, die auf die Seite A verlinkt, nimmt damit Einfluss auf den PageRank. Die resultierende Summe wird dann mit dem Dämpfungsfaktor d multipliziert. Dadurch wird die Weitergabe des PageRanks von einer Seite zu einer anderen verringert. Dieser Faktor hat einen Wert von 0 bis 1. Dieser Wert geht auf das Prinzip des Zufallssurfers (Random Surfer Modell) zurück. Dieses Prinzip besagt, dass mit einer bestimmten Wahrscheinlichkeit ein Nutzer die Hyperlinks von Webseite zu Webseite verfolgt. Nach einer gewissen Wahrscheinlichkeit bricht der Nutzer die Linkverfolgung ab und beginnt weiter bei einer neuen zufälligen Webseite.

Beim PageRank gibt es noch eine weitere abweichende Formulierung. Diese wird in der Formel 4.3 dargestellt und enthält zusätzlich den Faktor N. Wobei N die Anzahl aller Seiten im Web darstellt. Diese Formel unterscheidet sich nicht grundsätzlich von der anderen Formel. In dieser Formel wird lediglich auf die Wahrscheinlichkeit aller Seiten im Internet eingegangen und bildet eine Wahrscheinlichkeitsverteilung über alle Seiten. Die Summe aller PageRank-Werte entspricht hier dem Wert 1 (vgl. [OnP151]), (vgl. [eFa15]), (vgl. [Dir15], 101-105).

$$PR(A) = (1-d) + d \sum_{i \in \{1, \ldots, n\}} \frac{PR(T_i)}{C(T_i)}$$

Formel 4.2: Erste Version des PageRank-Algorithmus (nach [eFa15])

$$PR(A) = \frac{(1-d)}{N} + d \sum_{i \in \{1, \ldots, n\}} \frac{PR(T_i)}{C(T_i)}$$

Formel 4.3: Zweite Version des PageRank-Algorithmus (nach [eFa15])

Term	Erklärung
PR(A)	Gibt den PageRank einer Seite an.
d	Dieser Faktor ist ein Dämpfungsfaktor und kann zwischen 0 und 1 liegen. Ein großer Wert wie die 1 bedeutet, dass der PageRank komplett weitergeleitet wird. Kleinere Werte verursachen eine Dämpfung des PageRanks.
PR(T_i)	Gibt den PageRank der externen Seiten T an, die auf die Seite A verlinkt sind.
C(T_i)	Gibt die Anzahl der externen Links auf der Seite T an.
N	Anzahl aller Seiten im Internet.

Tabelle 4.4: Erklärung der PageRank-Terme

4.3.1 Berechnung durch ein Gleichungssystem

Wenn eine Seite mit einem hohen PageRank auf eine andere Seite verlinkt, wird diese besser bewertet als von einer privaten Homepage mit einem geringeren PageRank. Anhand eines Beispiels mit der Formel 4.2 soll das verdeutlicht werden. Dazu werden wie in Abb. 4.3 drei Seiten A,B und C dargestellt. Die Seite A verlinkt auf die beiden Seiten B und C. Die Seite B verlinkt nur auf die Seite C. Und die Seite C verlinkt wiederum nur auf die Seite A. Üblicherweise beträgt der Dämpfungsfaktor d nach Angaben von Larry Page und Sergey Brin den Wert 0,85. Nach dem Einsetzen in die Formel entsteht für jede Seite eine eigene Gleichung. Diese Gleichungen sind in der Formel 4.4 dargestellt und werden dann als Gleichungssystem gelöst. Die Berechnungsschritte für das Lösen des Gleichungssystems befinden sich in Anhang A – Berechnung des PageRanks durch ein Gleichungssystem.

Das Beispiel zeigt, dass der PageRank der Seite C am höchsten mit 1,19 ist. Am schlechtesten ist der PageRank der Seite B mit 0,64. Der PageRank für die Seite A ist fast genauso hoch wie die Seite C mit 1,16. Alle PageRanks zusammen ergeben den Wert 3. Dies entspricht der Anzahl der Seiten. Dieses Beispiel verdeutlicht, dass je mehr Webseiten auf die eigene Webseite verweisen, desto höher steigt der PageRank (vgl. [eFa15]), (vgl. [Hol15]).

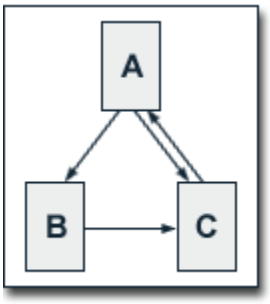


Abb. 4.3: Verlinkung der drei Seiten untereinander (übernommen von [eFa15])

$$PR(A) = (1-d) + d(PR(C)) = \frac{2058}{1769} = 1{,}163369$$

$$PR(B) = (1-d) + d\left(\frac{PR(A)}{2}\right) = \frac{1140}{1769} = 0{,}644432$$

$$PR(C) = (1-d) + d\left(\frac{PR(A)}{2} + PR(B)\right) = \frac{2109}{1769} = 1{,}192199$$

Formel 4.4: Aufstellen der Gleichungen für dieses Beispiel und Lösung des Gleichungssystems

4.3.2 Berechnung durch Iteration

In dem gezeigten Beispiel mit drei Seiten lässt sich das Gleichungssystem unproblematisch lösen. Das Internet besteht aber aus Milliarden von Webseiten. Dadurch ist das Lösen eines Gleichungssystems nicht möglich. Die Lösung des Problems kann durch eine iterative Berechnung gelöst werden. Dadurch wird der PageRank näherungsweise berechnet. Jede Seite wird ein Anfangswert zugewiesen. In diesem Beispiel beträgt der Anfangswert die Zahl 1. Mit mehreren Berechnungsdurchläufen wird dann der PageRank aller Seiten ermittelt. Anhand des Beispiels aus der Abb. 4.3 soll das verdeutlicht werden. Die Tabelle für die iterative Berechnung befindet sich im Anhang B – Iterative Berechnung des PageRank. Der Dämpfungsfaktor d beträgt auch hier 0,85. In der iterativen Berechnung ist der PageRank mit 6 Nachkommastellen bereits nach 27 Iterationsschritten erreicht. Für das komplette Internet sind mehr Iterationsschritte notwendig. Nach Larry Page und Sergey Brin beträgt der Wert zirka 100 Iterationsschritte (vgl. [eFa15]).

4.3.3 Google PageRank

Der PageRank von Google wird in einer Skala von 0 bis 10 gemessen. Diese Werte werden in der Tabelle 4.5 erklärt. Oft wird fälschlicherweise angenommen, dass nur nach dem PageRank die Dokumente bei einer Suche sortiert werden. Der PageRank ist aber nur eins von vielen Verfahren, um Dokumente in der Ergebnisliste zu sortieren. Außerdem hat jede Unterseite einer Seite einen eigenen PageRank. Die Startseite hat in der Regel den höchsten PageRank, da externe Links größtenteils auf die Startseite verweisen. Hier gibt es eine Faustregel. Pro Navigationsebene tiefer wird der PageRank um den Wert 1 vermindert (vgl. [Mic15]), (vgl. [Dir15], S.101-105).

In der Tabelle 4.6 wird der PageRank von einigen Seiten dargestellt.[6] Sehr bekannte Seiten haben einen hohen Wert. Und Seiten die weniger bekannt sind einen niedrigeren Wert. Die Seite der Hochschule Neubrandenburg hat einen PageRank von 6. Dies entspricht einem exzellentem Wert. Währenddessen eine Unterseite von der Hochschule Neubrandenburg im Studiengang Geoinformatik nur einen PageRank von 4 aufweist. Eine andere Unterseite im Studiengang Geodäsie und Geoinformatik erreicht nur einen PageRank von 3. Wie bereits erläutert, besagt die Faustregel, dass der PageRank pro Navigationsebene abnimmt. Darum hat die Unterseite der Hochschule Neubrandenburg einen schlechteren PageRank als die Startseite.

[6] Der PageRank wurde unter folgender Seite errechnet: http://www.gaijin.at/olsgprank.php

Page-Rank	Bedeutung
0	Sehr Niedrig, noch nicht zugewiesen, nicht im Index oder von Google abgestraft.
1-2	Niedrig
3	In Ordnung
4	Gut
5	Sehr Gut
6-7	Exzellent
8-10	Dieser Wert wird meist nur von ganz Großen erreicht wie z.B. Microsoft, Amazon, Adobe usw.

Tabelle 4.5: Erklärung der PageRank-Skala (übernommen von [Mic15])

Page-Rank	URL
3	http://www.hs-nb.de/studiengang-gg/
4	http://www.hs-nb.de/studiengang-gi/
4	http://www.hs-nb.de/studiengang-gm/
4	http://www.hs-nb.de/fachbereich-lg/
5	http://dmoz.de/
6	http://www.hs-nb.de/
7	http://www.fh-stralsund.de/
8	http://de.wikipedia.org/
9	https://www.youtube.com/

Tabelle 4.6: Der PageRank von Seiten im Internet

4.4 TF-IDF

Der TF-IDF Algorithmus dient zum Bewerten von Webseiten und ist sehr komplex. Der Algorithmus ist in der Formel 4.5 dargestellt. Dieser besteht aus einzelnen Bestandteilen. Die Bestandteile *tf*, *idf*, *boost* und *norm* sind bezogen auf einzelne Terme und *coord* und *queryNorm* auf die gesamte Suchanfrage.

$$score(q,d) = coord(q,d) * queryNorm(q) * \sum_{t\ in\ q} \left(tf(t\ in\ q) * idf(t)^2 * t.getBoost() * norm(t,d) \right)$$

Formel 4.5: TF-IDF Algorithmus (nach [Klo14], S.195)

Die Anteile der Terme einer Suchanfrage sind im Faktor *coord* enthalten. Dieser beschreibt die Anzahl der gefundenen Suchterme (docTerms) dividiert durch die Anzahl der gesuchten Terme (queryTerms) im Dokument. Hat beispielsweise eine Suchanfrage für ein Dokument 3 von 4 Suchbegriffen enthalten, ergibt sich wie in der Formel 4.6 dargestellt ein Faktor von 0,75 für die Berechnung.

$$coord(q,d) = \frac{docTerms}{queryTerms} = \frac{3}{4} = 0{,}75$$

Formel 4.6: Berechnung des coord-Faktors (nach [Klo14], S.195)

Der Normalisierungsfaktor *queryNorm* ist zum Vergleich unterschiedlicher Suchanfragen notwendig, damit Teilsuchanfragen in Bezug auf die Gesamtbewertung einer Abfrage vergleichbar gemacht werden können. Mit der Formel 4.7 kann dieser rechnerisch durch die Gewichtung einzelner Terme berücksichtigt werden. Der Faktor hat dabei keine Auswirkung auf den berechneten Score-Wert, da bei allen Dokumenten mit dem gleichen Wert gerechnet wird.

$$queryNorm = \frac{1}{sumOfSquaredWeights} = \frac{1}{q.getBoost()^2 * \sum \left(idf(t) * t.getBoost() \right)^2}$$

Formel 4.7: Berechnung des Normalisierungsfaktors (nach [Klo14], S.196)

Die „Term Frequency“ *tf* bestimmt die Häufigkeit eines Terms im Dokument. Folgende Formel 4.8 wird dafür angegeben. Ist beispielsweise in einem Dokument ein Term neunmal vorhanden, so ergibt sich der Wert 3 dafür.

$$tf(t\ in\ d) = \sqrt{frequency} = \sqrt{9} = 3$$

Formel 4.8: Berechnung der Term Frequency (nach [Klo14], S.196)

Die Häufigkeit des Terms im Dokumentenbestand wird mit der „Inverse Document Frequency“ *idf* beschrieben. Dieser besagt, je seltener der Term in den Dokumenten vorhanden ist, desto wichtiger ist der Term. Die Formel 4.9 zeigt die Anzahl der Dokumente, bei dem der Term enthalten ist *docFreq*. Und die Anzahl aller Dokumente im Datenbestand *numDocs*. Diese Formel wird dann in Relation mit dem Logarithmus gesetzt. Sind beispielsweise 10 Dokumente im Datenbestand vorhanden und ein Term wird in fünf Dokumenten gefunden, so ergibt sich ein Wert von 1,22.

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq + 1}\right) = 1 + \log\left(\frac{10}{5+1}\right) = 1{,}22$$

Formel 4.9: Berechnung der Inverse Document Frequency (nach [Klo14], S.196)

Jeder Term kann mit einem Term-Boost *t.getBoost()* während der Suchanfrage ausgestattet werden. Der Faktor wird direkt in die Formel übernommen. Der letzte Faktor *norm(t,d)* besteht aus drei Einzelfaktoren, die bei der Indexierung berechnet werden. So wie in der Formel 4.10 dargestellt, sind das der Dokument-Boost *doc.getBoost()*, der Feld-Boost *f.getBoost()* und die Längennormalisierung *lengthNorm*. Der Dokument-Boost ist ein Wert, der beim Indexieren mitgegeben wird. Der Feld-Boost wird zu einem Feld zugeordnet. Diese beiden werden für die Berechnung direkt als Faktoren übernommen.

$$norm(t,d) = doc.getBoost() * \prod f.getBoost() * lengthNorm$$

Formel 4.10: Berechnung des Normalisierungsfaktors (nach [Klo14], S.197)

Der Wert der Längennormalisierung wird während der Indexierung berechnet und hängt von der Anzahl der Terme in den Feldern ab. Dabei gilt, je mehr Terme ein Feld enthält, umso geringer wird der Score ausfallen. Die Berechnung der Längennormalisierung ist in Formel 4.11 dargestellt. Hat ein Feld beispielsweise 100 Terme im Index, dann beträgt der Wert für die Längennormalisierung 0,1.

$$lengthNorm = \frac{1}{\sqrt{numTerms}}$$

Formel 4.11: Berechnung der Längennormalisierung (nach [Klo14], S.197)

Der Algorithmus hängt von vielen Faktoren ab, dennoch haben die beiden Faktoren „Term Frequency“ *tf(t in d)* und „Inverse Document Frequency“ *idf(t)* den größten Einfluss. Dies ist der Grund, weshalb der Algorithmus nach diesen beiden Faktoren benannt wurde (vgl. [Klo14], S.194-197).

5 Suchergebnisse

Dieses Kapitel beschäftigt sich mit den Suchergebnissen. Zunächst wird auf die stetige Entwicklung der Ergebnisseiten eingegangen. Danach werden Eingabemethoden dargestellt, die eine effizientere Suche mit einer Suchmaschine ermöglichen.

5.1 Aufbau der Suchergebnisseiten

5.1.1 Entwicklung

Seit dem Beginn der Suchmaschinen haben sich diese stetig entwickelt. Darunter gehört auch die Darstellung der Suchergebnisse. Wie in der Abb. 5.1 (a) dargestellt, bestand die Suchergebnisseite nur aus dem Suchfeld, Optionen und den Treffern. Dies entspricht der ursprünglichen Form einer Suchmaschine. Diese wurde dann im Laufe der Zeit durch komplexere Darstellungen verdrängt. Die zweite Suchergebnisseite (b) enthält zusätzlich noch eingefügte Werbung. Diese stand entweder am Anfang der Trefferliste oder/und rechts in einer weiteren Liste. Durch diese zusätzliche Spalte auf der rechten Seite hat sich die Suchergebnisdarstellung wesentlich verändert. Die dritte Suchergebnisseite (c) enthält weiterhin noch Werbung. Hinzugekommen sind die Ergebnisse der „Universal Search“. Diese sind in der linken Spalte zwischen einzelnen Treffern angeordnet. Die letzte Suchergebnisseite (d) zeigt die aktuelle Anordnung. Diese wurde durch einen Knowledge-Graph erweitert (vgl. [Dir15], S.126-130).

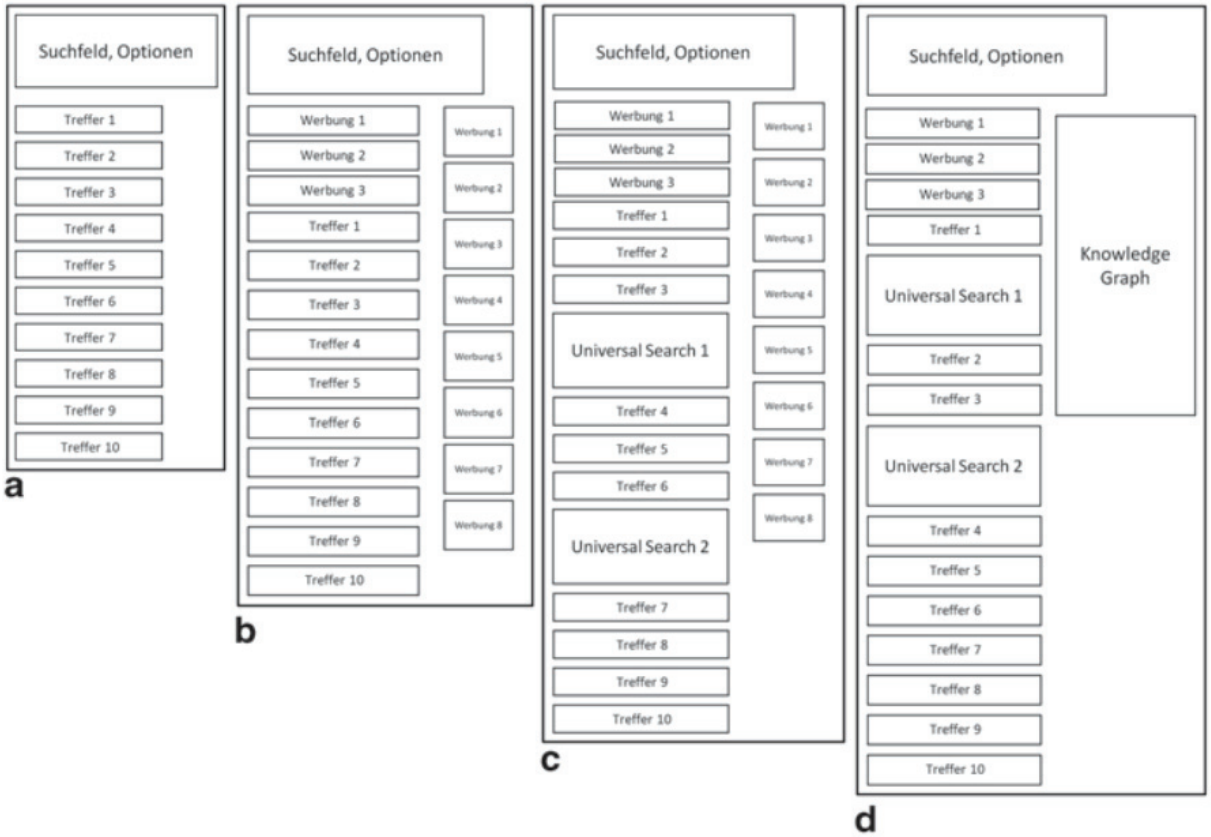


Abb. 5.1: Entwicklung der Layouts von Suchmaschinen

5.1.2 Horizontale und vertikale Suche

Bei Suchmaschinen lässt sich die Suche in horizontale und vertikale Suche unterscheiden. Die horizontale Suche soll das Internet in der Breite abdecken. Das ganze Internet wird möglichst weit abgesucht und die Ergebnisse werden nicht spezialisiert. Alle Inhalte sollen in Verbindung mit der Suchanfrage gefunden werden. Im Idealfall soll bei einer Suchanfrage alles zum Suchbegriff passende gefunden werden. Hingegen soll die vertikale Suche spezialisiert suchen. Es können nur bestimmte Typen von Inhalten oder thematisch spezialisierte Inhalte gesucht werden. Viele Suchmaschinen, unter anderem Google und Bing, stellen beide Formen zur Verfügung. So kann nicht nur horizontal, sondern auch vertikal gesucht werden. Diese Kategorien können wie in der Abb. 5.2 dargestellt auf Orte, Bilder, News, Videos und vieles mehr spezialisiert sein (vgl. [Dav15]).

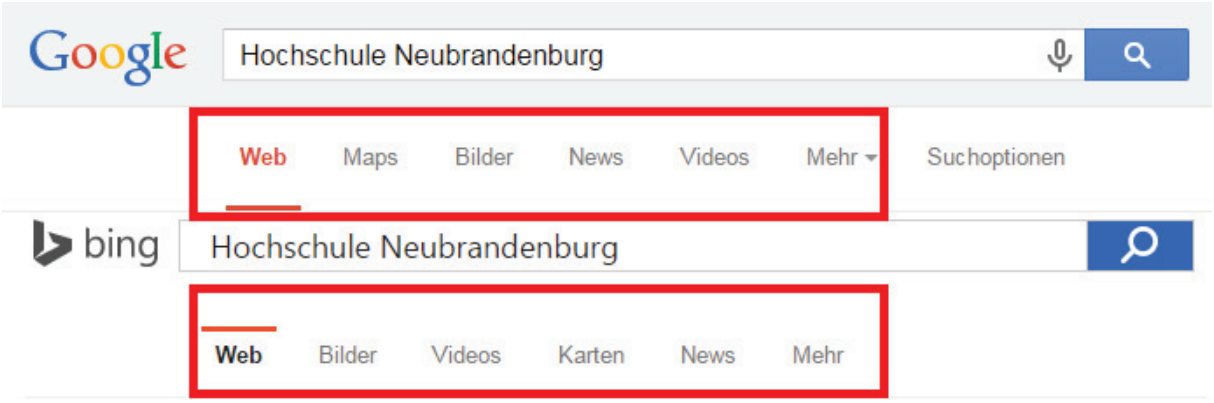


Abb. 5.2: Vertikale Suche am Beispiel Google und Bing

5.1.3 Knowledge-Graph

Der Knowledge-Graph ist ein Kasten, bei dem nur die wichtigsten Informationen zu einer Person, einem Bauwerk, einem Unternehmen, einem Ort o.ä. angezeigt werden. Die Suchmaschine erstellt diese Informationen automatisiert aus mehreren Quellen zusammen. Bei einer Suchanfrage stellt der Knowledge-Graph die Fakten komprimiert dar, sodass der Nutzer nicht auf eine andere Webseite gehen muss (vgl. [Dir15], S.138).

5.1.4 Universal Search

Die „Universal Search“ (Universalsuche) kombiniert beide Arten der horizontalen und vertikalen Suche. Zu der horizontalen Suche werden vertikale Suchergebnisse miteinbezogen und angezeigt. Hier werden die Ergebnisse einer Suchanfrage direkt in die Websuche integriert, ohne dass vorher auf die entsprechende Kategorie gewechselt wird. Je nach dem Suchbegriff werden die besten Ergebnisse anderer Kategorien in einer sogenannten „Onebox“ präsentiert. Dieser Kasten enthält Informationen zum Suchbegriff wie z.B. Bilder, Videos oder News. Wie in der Abb. 5.3 dargestellt, erscheinen zum Suchbegriff „Neubrandenburg“ drei weitere Kästen mit zusätzlichen Informationen anderer Kategorien. Der rechte Kasten (der Knowledge-Graph) enthält dabei allgemeine Informationen zum Suchbegriff, der obere mar-

kierte Kasten enthält Informationen der Kategorie News und der untere Kasten vier Bilder aus der Kategorie Bilder. Je nach Eingabe des Suchbegriffes können sich die Kästen in der Position und dem Inhalt ändern. Weiterhin kann sich die Anordnung pro Tag stetig verändern (vgl. [Dav151]).

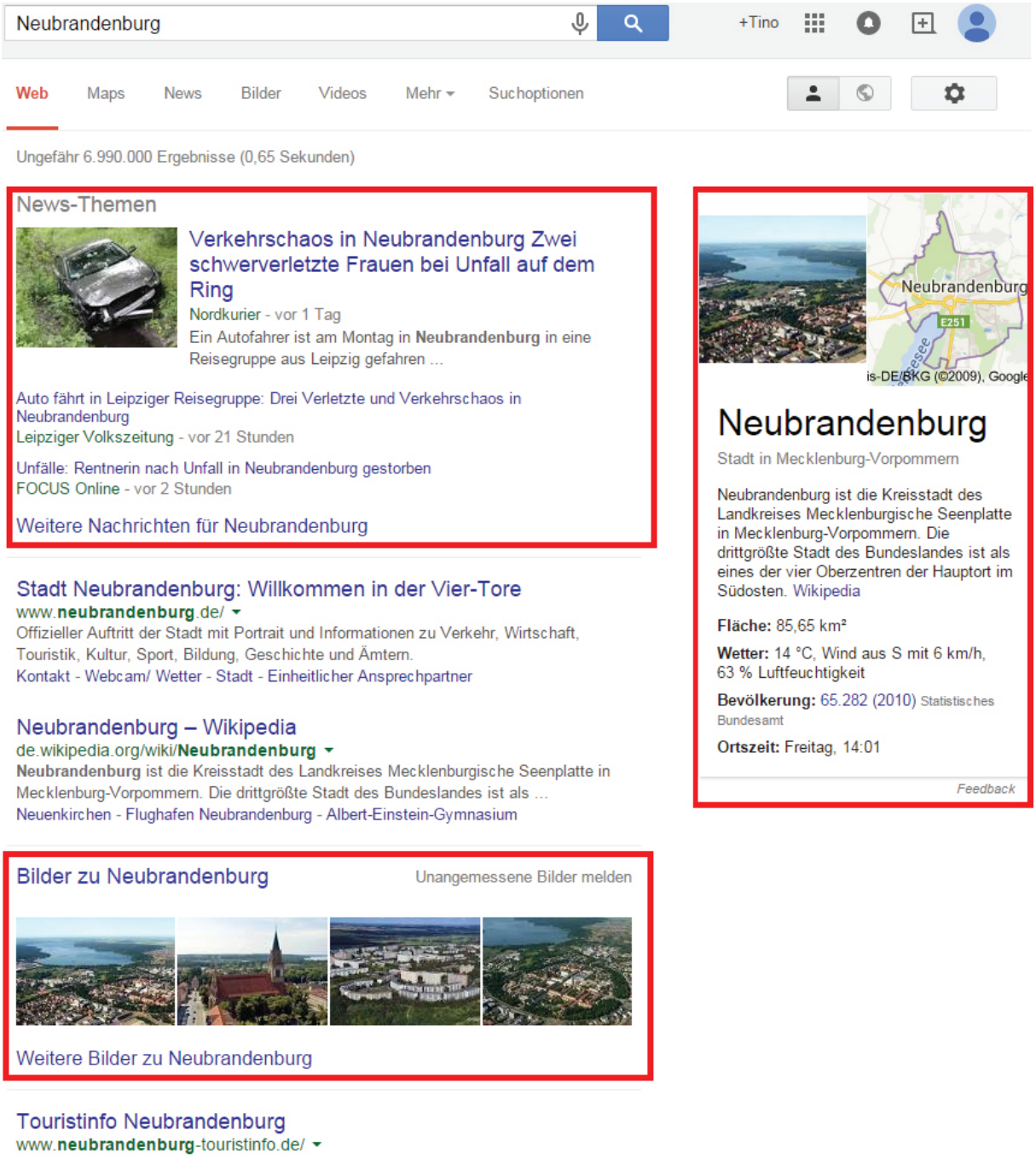


Abb. 5.3: Horizontale Suche mit vertikalen Suchergebnissen am Beispiel Google

5.1.5 Textanzeigen

Damit diese Suchmaschinen existieren können, müssen die Kosten für den Betrieb erwirtschaftet werden. Ein guter Ansatz hierfür ist die Einblendung von Werbung. Diese Werbung wird je nach Suchanfrage angepasst und als Textanzeige bei den Suchergebnissen dargestellt. Der Werbetreibende kann bei diesem Modell leichter entscheiden, wann der Nutzer die Werbung sehen soll. Dadurch steigt die Wahrscheinlichkeit, dass ein Nutzer auf die Werbung reagiert (vgl. [Dir15], S.4).

5.2 Eingabemethoden

Bei der Nutzung einer Suchmaschine gibt es verschiedene Möglichkeiten die Suchanfrage genauer zu formulieren. Dazu gehören der Einsatz von booleschen Operatoren, das erweiterte Suchformular und weitere Befehle im Suchfeld.

5.2.1 Boolesche Operatoren

Die booleschen Operatoren können die Ergebnisse einer Suchanfrage gezielt einschränken oder erweitern. Dazu gehören die Operatoren AND, OR und NOT. Es lassen sich beliebig viele Operatoren miteinander kombinieren, sodass Suchanfragen sehr komplex formuliert werden können. In der Abb. 5.4 werden diese Operatoren mit zwei Suchanfragen A und B dargestellt. Der AND-Operator verknüpft zwei Suchbergriffe miteinander. Dadurch werden nur Dokumente gefunden, die beide Suchbegriffe beinhalten. Damit kann die Treffermenge gezielt verringert werden. Hingegen der OR-Operator erhöht die Anzahl der Treffermenge. Bei zwei Suchbegriffen können somit Dokumente gefunden werden, die entweder den ersten oder den zweiten Suchbegriff beinhalten. Der letzte Operator NOT schließt einen Suchbegriff aus den Dokumenten aus. Das heißt, dass dieser Suchbegriff in keinem der Dokumente vorhanden sein darf. Dadurch verringert sich die Treffermenge. Oft werden bei einer Suchanfrage mehrere Wörter mit einem Leerzeichen getrennt eingegeben. Die Suchanfrage fügt dann automatisch den Operator AND zwischen den Wörtern ein. Dadurch kann die Treffermenge mit mehreren Wörtern automatisch stark eingeschränkt werden, ohne dass der Nutzer diese Befehle kennen muss (vgl. [Dir15], S.195-199).

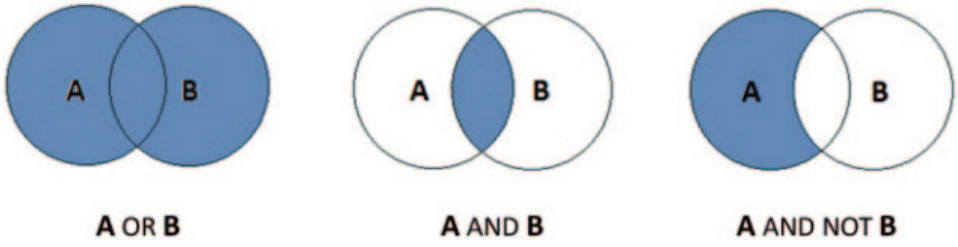


Abb. 5.4: Boolesche Operatoren bei der Mengenangabe (übernommen von [Dir15], S.196)

5.2.2 Erweitertes Suchformular

Um komplexere Suchanfragen stellen zu können, gibt es weitere Möglichkeiten. Eine alternative Möglichkeit ist die Eingabe über ein erweitertes Suchformular. Diese helfen bei der Eingabe komplexer Suchanfragen an die Suchmaschine. Dieses Hilfsmittel zur Eingabe präziser Suchanfragen kann beispielsweise durch einen Menüpunkt in der Suchmaschine aufgerufen werden. Bei diesem Suchformular sind dann vordefinierte Felder wie z.B. Sprache, URL, Dateityp usw. vorhanden.[7] Diese Felder bieten eine Zusammenstellung der wichtigsten Funktionen und sind für die meisten Fälle ausreichend, um die Suchanfrage zu verfeinern (vgl. [Dir15], S.200).

5.2.3 Befehle

Eine weitere Möglichkeit ist die Eingabe der Befehle direkt im Suchfeld. Diese Befehle variieren je nach Suchmaschine. In der Tabelle 5.1 sind einige ausgewählte Symbole und in der Tabelle 5.2 sind einige Befehle der Suchmaschine Google dargestellt. Durch die Anwendung von Befehlen kann die Suchanfrage sehr stark spezifiziert werden. Weiterhin kann die Suchanfrage zu den Befehlen auch mit booleschen Operatoren verknüpft werden. Dadurch lassen sich noch exaktere Suchanfragen formulieren (vgl. [Dir15], S.202).

Symbol	Funktion	Beispiel
#	Schränkt die Treffermenge auf bestimmte Hashtags ein.	#Neubrandenburg
-	Ergebnisse mit dem Suchbegriff nach einem Bindestrich werden entfernt. Dadurch kann die Treffermenge gezielt durch Begriffe beschränkt werden.	-Hochschule Neubrandenburg
“	Beschränkt die Treffermenge auf Webseiten, bei denen nur diese Wörter in der Reihenfolge vorkommen.	„Hochschule Neubrandenburg“
*	Der Stern dient als Platzhalter für unbekannte Begriffe.	"Die * Neubrandenburg ist"

Tabelle 5.1: Mögliche Operatoren bei einer Suchanfrage am Beispiel Google (vgl. [Goo15])

Befehl	Funktion	Beispiel
site:	Beschränkt die Treffermenge auf eine ausgewählte URL.	site:hs-nb.de
link:	Beschränkt die Treffermenge auf Webseiten, die auf diese Seite verweisen.	link:hs-nb.de
related:	Beschränkt die Treffermenge auf Webseiten, die dieser ähnlich sind.	related:hs-nb.de
intitle:	Beschränkt die Treffermenge auf die Wörter die im Titel vorkommen.	intitle:Neubrandenburg
filetype:	Beschränkt die Treffermenge auf ein bestimmtes Dateiformat.	filetype:pdf

Tabelle 5.2: Mögliche Befehle bei einer Suchanfrage am Beispiel Google (vgl. [Goo15])

[7] Bei der Suchmaschine Google.de unter folgender URL https://www.google.de/advanced_search erreichbar.

5.3 Beeinflussung durch Nutzerverhalten

Der Nutzer kann die Suchergebnisse einer Suchmaschine beeinflussen. Im nachfolgenden werden diese nutzerspezifischen Einflussfaktoren dargestellt.

5.3.1 Personalisierte Faktoren

Diese Faktoren beziehen sich auf jeden einzelnen und individuellen Nutzer. Beispielsweise können Informationen vom aktuellen Standort die Suchanfrage wesentlich beeinflussen, indem Suchergebnisse vom eigenen Land, der Stadt oder der Landessprache deutlich bevorzugt werden. Ein weiterer Faktor sind personenbezogene Daten des Nutzers über einen längeren Zeitraum. Es kann unter anderem ein Dokument bevorzugt werden, dass früher oft mehrfach aufgerufen wurde (vgl. [Dir15], S.171-172).

5.3.2 Soziale Netzwerke

Auch die sozialen Netzwerke nehmen Einfluss auf das Ranking der Suchergebnisse. Die Häufigkeit eines erwähnten oder geteilten Dokumentes von Personen wird bemessen, sowie Dokumente, die als Favorit markiert sind oder die Favoriten der Freunde. So kann der Anbieter der Inhalte, durch das Verbreiten seiner Angebote, einen Einfluss nehmen (vgl. [Dir15], S.171-172).

6 Suchmaschinen im Web

Dieses Kapitel beschäftigt sich mit den Suchmaschinen auf dem derzeitigen Markt und bietet einen ersten Überblick über die verschiedenen Suchmaschinen. Zunächst soll einmal der Markt erfasst und die Marktanteile der Suchmaschinen anschaulich dargestellt werden. Anschließend werden die Abhängigkeiten der einzelnen Suchmaschinen untereinander beschrieben. Danach werden die Volltextsuchmaschinen und andere Arten von Suchmaschinen gegenübergestellt und einem einfachen Suchtest unterzogen. Zum Schluss wird auf die möglichen Trends des aktuellen Suchmaschinenmarktes näher eingegangen.

6.1 Marktanteil der letzten Monate

Die Nutzer können verschiedene Suchmaschinen nutzen. Welche Suchmaschinen in Deutschland und weltweit genutzt werden, sollen die folgenden Abbildungen Abb. 6.1 und Abb. 6.2 darstellen. Die genauen Prozentwerte der Abbildungen werden in Tabelle 6.1 und Tabelle 6.2 gezeigt.[8] Diese Abbildungen zeigen die 5 am häufigsten genutzten Suchmaschinen für den Zeitraum Dezember 2014 bis Juli 2015 und wurden mit einer logarithmischen Y-Achse für die Prozente skaliert, um die Nutzungszahlen besser darstellen zu können. Beim Betrachten der Abbildungen zeigt sich, dass die Suchmaschine Google mit großem Abstand den höchsten Marktanteil aller Suchmaschinen besitzt und das sowohl national als auch international. Dieser Marktanteil liegt konstant bei ca. 90%. Andere Suchmaschinen schneiden deutlich schlechter ab und erreichen weltweit nicht mehr als 5%.

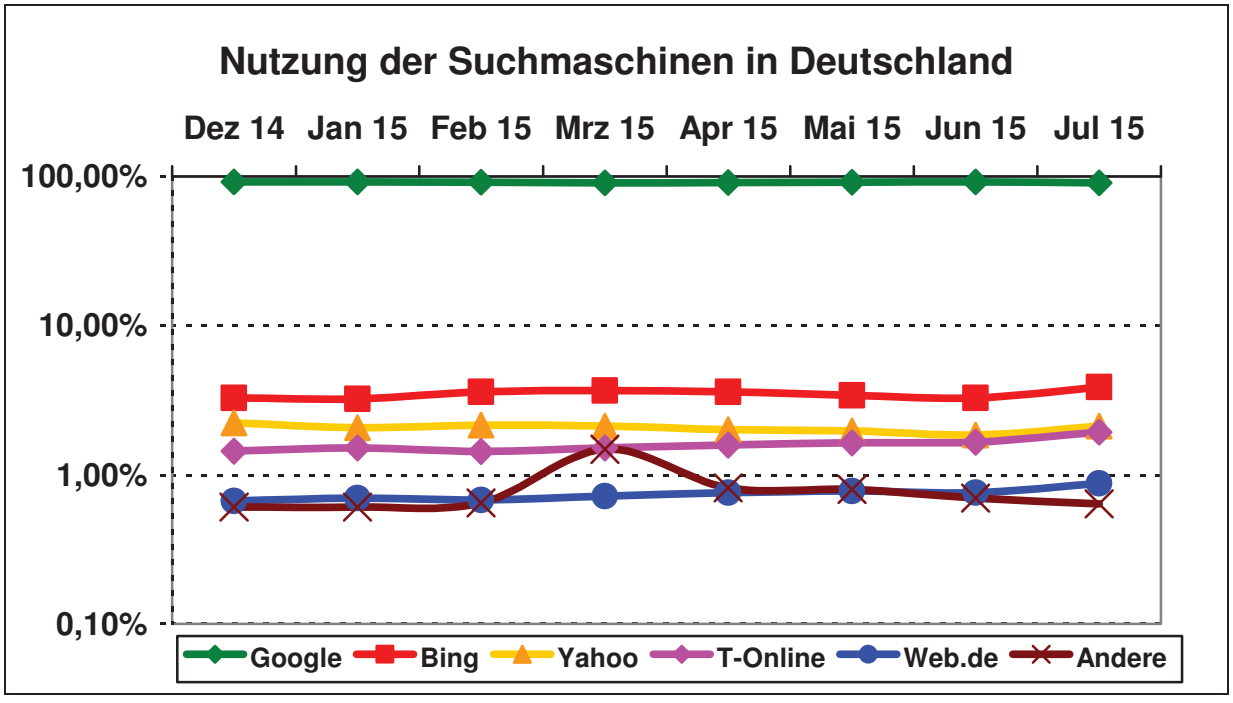


Abb. 6.1: Suchmaschinennutzung in Deutschland von Dezember 2014 bis Juli 2015

[8] Die Daten für die Nutzung der Suchmaschinen stammen von der Seite gs.statcounter.com.

Suchmaschine	Dez 14	Jan 15	Feb 15	Mrz 15	Apr 15	Mai 15	Jun 15	Jul 15
Google	91,79%	91,89%	91,49%	90,51%	91,24%	91,40%	91,77%	90,56%
Bing	3,27%	3,22%	3,60%	3,66%	3,60%	3,41%	3,28%	3,87%
Yahoo	2,23%	2,07%	2,15%	2,12%	2,01%	1,97%	1,84%	2,12%
T-Online	1,44%	1,51%	1,43%	1,51%	1,58%	1,64%	1,65%	1,93%
Web.de	0,67%	0,70%	0,68%	0,72%	0,76%	0,78%	0,76%	0,88%
Andere	0,61%	0,61%	0,65%	1,48%	0,82%	0,80%	0,70%	0,64%

Tabelle 6.1: Suchmaschinennutzung in Deutschland von Dezember 2014 bis Juli 2015 (übernommen von [Tay15])

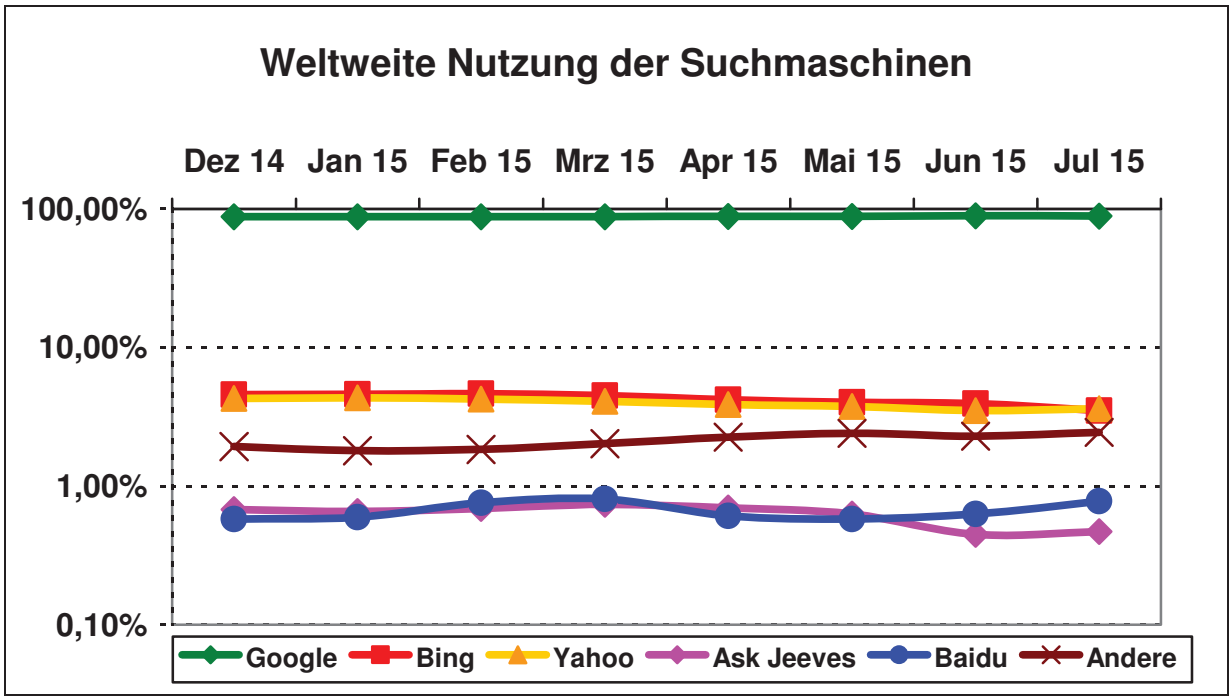


Abb. 6.2: Weltweite Nutzung der Suchmaschinen von Dezember 2014 bis Juli 2015

Suchmaschine	Dez 14	Jan 15	Feb 15	Mrz 15	Apr 15	Mai 15	Jun 15	Jul 15
Google	87,93%	88,01%	87,80%	87,82%	88,35%	88,61%	89,19%	89,17%
Bing	4,59%	4,61%	4,65%	4,50%	4,21%	4,03%	3,94%	3,51%
Yahoo	4,29%	4,33%	4,25%	4,10%	3,89%	3,75%	3,50%	3,63%
Ask Jeeves	0,68%	0,66%	0,69%	0,74%	0,70%	0,63%	0,45%	0,47%
Baidu	0,58%	0,60%	0,76%	0,81%	0,61%	0,58%	0,63%	0,78%
Andere	1,93%	1,80%	1,84%	2,03%	2,25%	2,40%	2,29%	2,44%

Tabelle 6.2: Weltweite Nutzung der Suchmaschinen von Dezember 2014 bis Juli 2015 (übernommen von [Tay15])

6.2 Marktanteil der letzten Jahre

Die Marktanteile der letzten Monate haben sich kaum verändert. Um ein Bild über den Suchmaschinenmarkt zu bekommen, werden nun die Marktanteile der letzten Jahre in den Abbildungen Abb. 6.3 und Abb. 6.4 dargestellt. Die genauen Prozentwerte der Abbildungen werden in Tabelle 6.3 und Tabelle 6.4 gezeigt. Diese Abbildungen verdeutlichen, dass die Suchmaschine Google im Laufe der Jahre einige wenige Prozente verloren hat. In Deutschland hatte diese am Anfang 2010 noch 97% und sank bis 2015 auf 91%. Dies liegt wahrscheinlich an der zunehmenden Konkurrenz anderer Suchmaschinen auf dem Markt. Während Google geringfügig absteigt, steigen die Konkurrenten Bing und Yahoo auf.

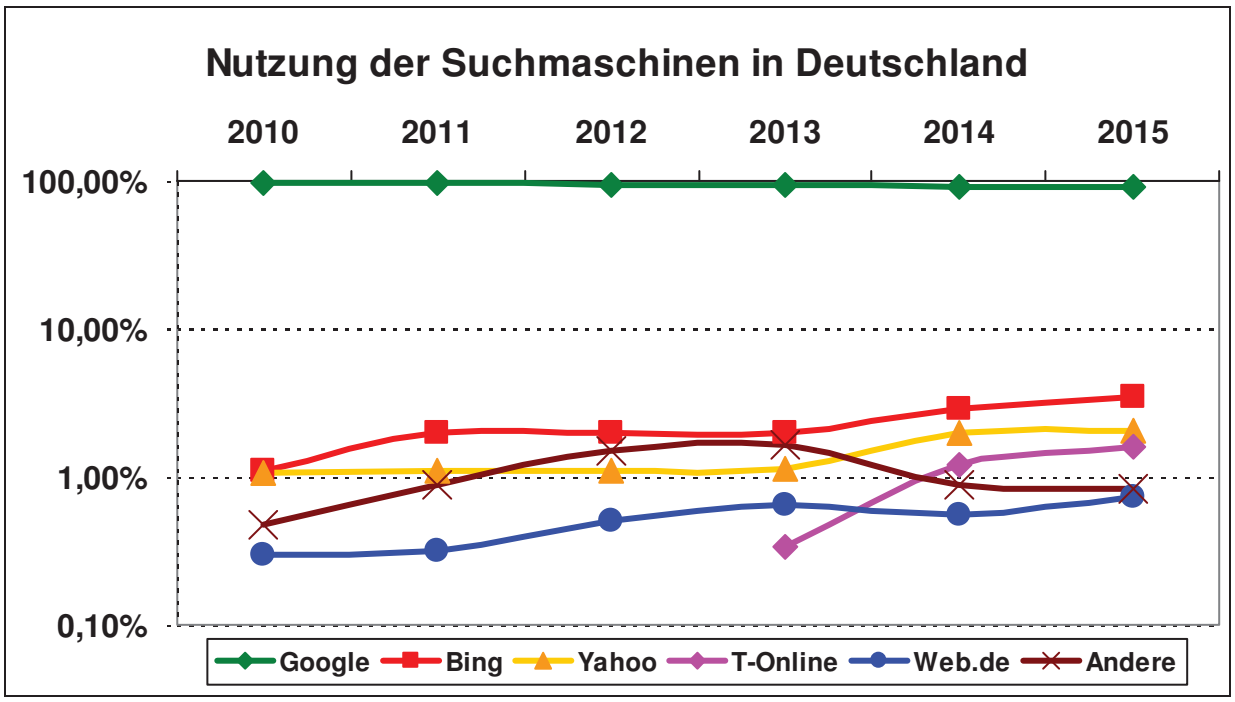


Abb. 6.3: Suchmaschinennutzung in Deutschland von 2010 bis 2015

Suchmaschine	2010	2011	2012	2013	2014	2015
Google	97,07%	95,73%	94,86%	94,26%	92,49%	91,28%
Bing	1,10%	1,99%	2,01%	1,99%	2,86%	3,52%
Yahoo	1,06%	1,09%	1,10%	1,12%	2,01%	2,04%
T-Online	-	-	-	0,34%	1,21%	1,60%
Web.de	0,30%	0,32%	0,50%	0,65%	0,56%	0,74%
Andere	0,47%	0,87%	1,52%	1,64%	0,88%	0,82%

Tabelle 6.3: Suchmaschinennutzung in Deutschland von 2010 bis 2015 (übernommen von [Tay15])

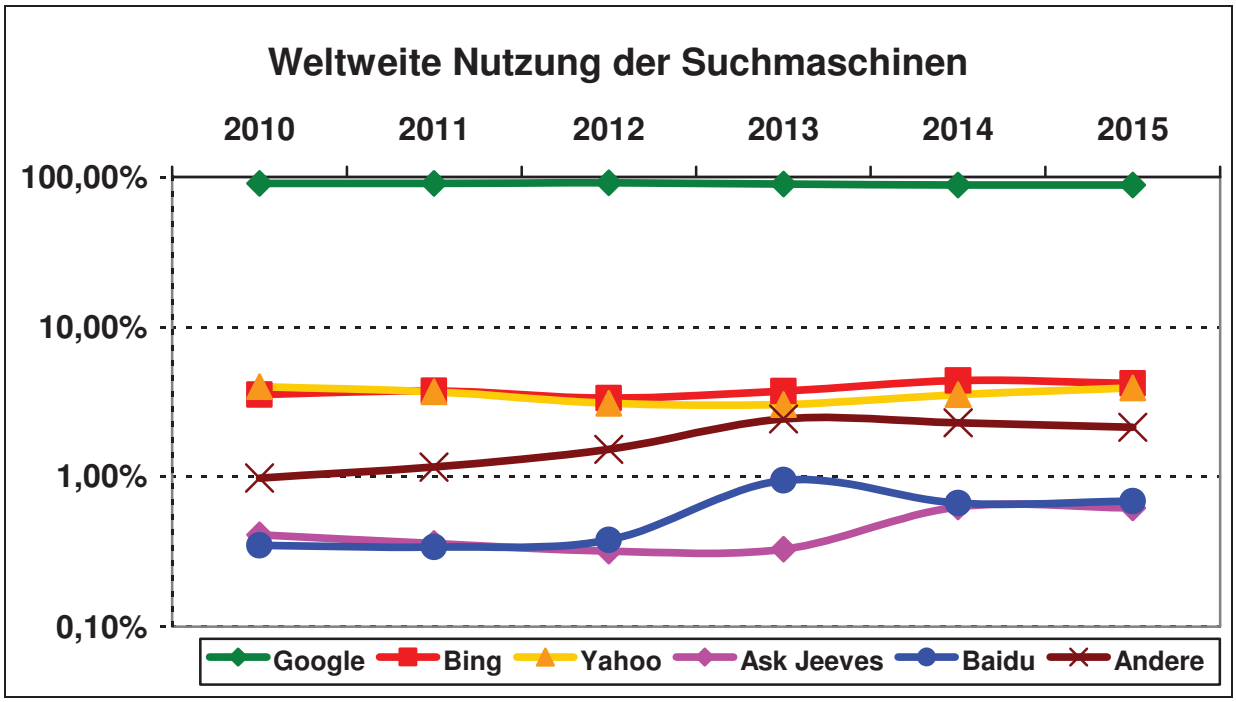


Abb. 6.4: Weltweite Nutzung der Suchmaschinen von 2010 bis 2015

Suchmaschine	2010	2011	2012	2013	2014	2015
Google	90,73%	90,70%	91,35%	89,52%	88,50%	88,44%
Bing	3,53%	3,74%	3,34%	3,73%	4,38%	4,19%
Yahoo	4,00%	3,70%	3,09%	3,04%	3,53%	3,92%
Ask Jeeves	0,41%	0,36%	0,32%	0,33%	0,63%	0,62%
Baidu	0,35%	0,34%	0,38%	0,95%	0,67%	0,69%
Andere	0,98%	1,16%	1,52%	2,43%	2,29%	2,14%

Tabelle 6.4: Weltweite Nutzung der Suchmaschinen von 2010 bis 2015 (übernommen von [Tay15])

6.3 Beziehungsgeflecht der Suchmaschinen

Es gibt eine Vielzahl an Suchmaschinen auf dem Markt, doch nur wenige weisen einen eigenen Index auf. Beispielsweise geben Google und Bing ihre Suchergebnisse auch an Partner weiter. Die Suchmaschine Yahoo hat seit Jahren ihre eigene Suchmaschine aufgegeben und liefert stattdessen nur noch Suchergebnisse von Bing. Dieses Teilen der Suchergebnisse wird auch das sogenannte Partnerindex-Modell genannt.

Viele Portale wie GMX oder T-Online greifen auf dieses Modell zurück. Mit dieser Methode haben beide Seiten Vorteile. Die Gewinne durch die Textanzeigen werden geteilt. Dem Suchmaschinenbetreiber entstehen durch das Liefern der Suchergebnisse an Partner nur geringe Kosten und die Kosten für den Betrieb einer Suchmaschine entfallen für den jeweiligen Partner. Dies ist ein Grund, warum es nur noch wenige alternative Suchmaschinen auf dem Markt gibt. Viele Suchportale setzen daher auf dieses Modell, da es für sie lukrativer ist.

Die folgende Abb. 6.5 zeigt das Beziehungsgeflecht der deutschen Suchmaschinen. In dieser Abbildung sind zunächst anhand der Umrandungen die Suchmaschinen mit eigenem Index zu erkennen. Diese Suchmaschinen sind Google, Bing, Exalead und Yandex. Google

und Bing sind die beiden zentralen Lieferer für Suchergebnisse an viele andere Suchmaschinen bzw. Suchportale. Neben den Suchergebissen (schwarze durchgehende Pfeile) werden auch Textanzeigen (graue Pfeile) für die Refinanzierung mit dem Partnerindex-Modell geliefert. Verdeutlicht zeigt sich, dass hinter der Vielzahl an Suchmaschinen bzw. Suchportalen nur in wenigen Fällen eine eigene Suchmaschine steht. Viel mehr wird auf Partnerschaften größerer Suchmaschinen gesetzt (vgl. [Dir15], S.156-158).

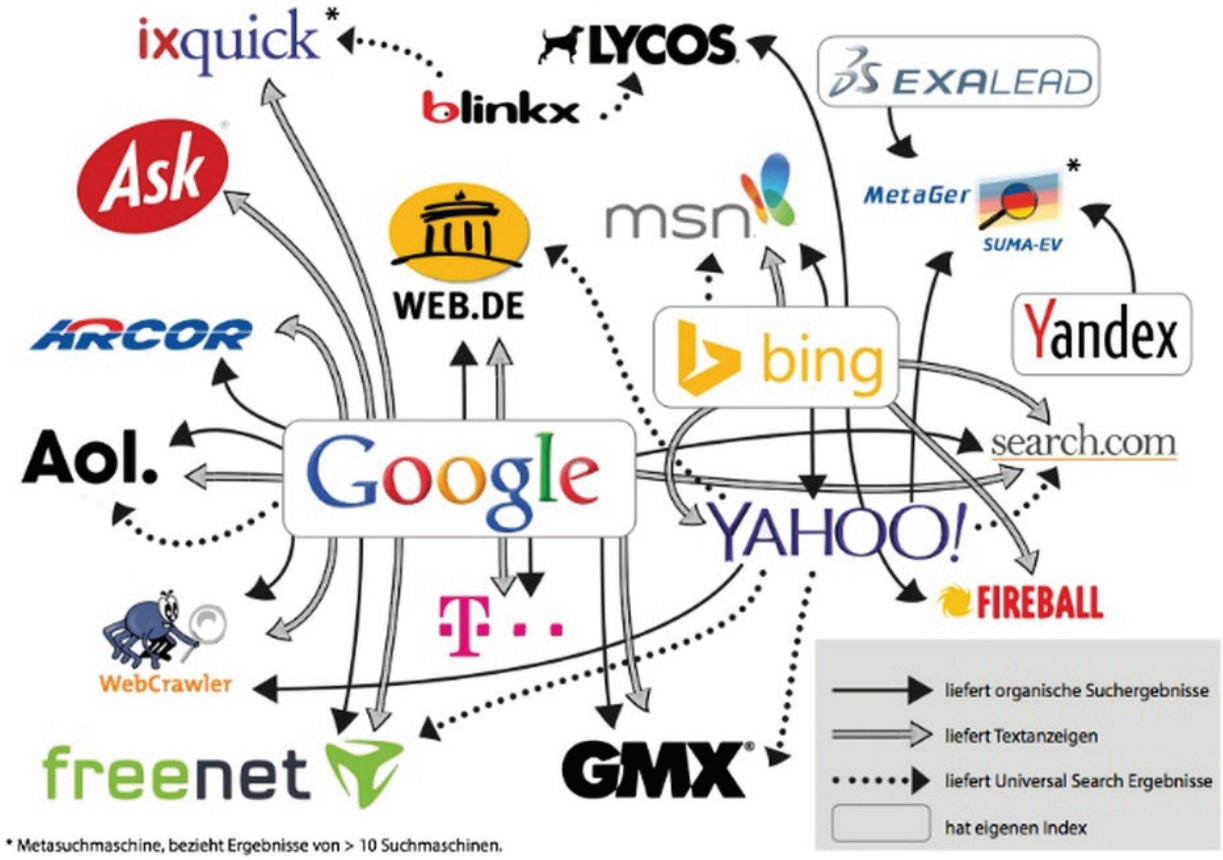


Abb. 6.5: Beziehnungsgeflecht der Suchmaschinen in Deutschland (übernommen von [Dir15], S.158)

6.4 Vergleich der Suchmaschinen

Im Folgenden werden die Suchmaschinen getestet und ihre Ergebnisse ausgewertet. Zunächst werden die Volltextsuchmaschinen untersucht, da diese unabhängig von anderen Suchmaschinendiensten sind und einen eigenen unabhängigen Datenbestand besitzen. Getestet werden die sechs Volltextsuchmaschinen Google, Bing, Exalead, Yandex, DeuSu und Hotbot. In diesem Test wird nach den folgenden Seiten in der Tabelle 6.5 gesucht. Dieser Test soll die genannten Suchmaschinen gegenüberstellen und zeigen, an welcher Position sich die Seiten in der Ergebnisliste befinden und wie viele Treffer die jeweilige Suchmaschine erreicht. Die deutlich markierten Text-Anzeigen werden bei der Position der Suchbegriffe nicht gewertet. Die Ergebnisse des Tests sind in der Tabelle 6.6 dargestellt.

Die drei Suchmaschinen Google, Bing und Hotbot haben in diesem Test sehr gute Ergebnisse erzielt. Die gesuchten Webseiten standen in der Ergebnisliste an den ersten vier Positionen. Die Treffermenge ist bei Google höher für die Eingabe eines Suchbegriffes als bei Bing. Bei der Eingabe von drei Suchbegriffen haben Google und Bing ungefähr gleich viele Treffer. Nach der Eingabe von drei oder mehr Suchbegriffen gibt es einen Unterschied. Die Treffer von Google werden reduziert, während hingegen die Treffer von Bing steigen. Auch bei Hotbot schwankt die Anzahl der Ergebnisse mit der Eingabe von mehreren Suchbegriffen.

Die Suchmaschinen Exalead, Yandex und DeuSu konnten im Test nicht überzeugen. Exalead fand, trotz der enormen Ergebnismenge, nur wenige der gesuchten Webseiten. Zu den Suchbegriffen Hochschule Neubrandenburg fand diese Suchmaschine die Webseite erst an der Position 142. Hier zeigt sich, dass die Suchmaschine das Suchergebnis nicht optimal gerankt hat. Die Stundenplan Webseite fand diese Suchmaschine erst beim Weglassen des Suchbegriffes „Master“ an einer guten Position. Yandex hatte zwar deutlich bessere Ergebnisse als die Suchmaschine Exalead, fand aber nur die Hälfte der gesuchten Webseiten. Trotzdem waren die gesuchten Webseiten stets auf den ersten zwei Positionen. DeuSu fand nur zwei der Webseiten auf den ersten fünf Positionen, dies liegt an dem kleinen Datenbestand der Suchmaschine.

Test	Webseite	Suchbegriffe
a	www.neubrandenburg.de	Neubrandenburg
b	www.hs-nb.de	Hochschule Neubrandenburg
c	www.hs-nb.de/studiengang-gi/	Hochschule Neubrandenburg Geoinformatik
d	www.hs-nb.de/studiengang-gg/	Hochschule Neubrandenburg Geoinformatik Master
e	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan
f	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan 1.Semester
g	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Stundenplan

Tabelle 6.5: Die erwartete Webseite und die verwendeten Suchbegriffe für die Suche

Dienst	Besonderheiten	Position	Ergebnisse
Google	• PageRank 7 • Die beliebteste Suchmaschine weltweit • Nutzt die eigene NoSQL-Datenbank Bigtable	a) 1 b) 1 c) 1 d) 1 e) 3 f) 1 g) 2	a) 7.120.000 b) 257.000 c) 5.830 d) 3.820 e) 750 f) 355 g) 24.100
Bing	• PageRank 8 • Verknüpft die Suchbegriffe mit dem Befehl AND	a) 1 b) 1 c) 1 d) 2 e) 1 f) 1 g) 3	a) 5.910.000 b) 133.000 c) 5.670 d) 91.400 e) 398.000 f) 65.900 g) 66.800
Exa-lead	• Besitzt keinen öffentlichen PageRank • Verknüpft die Suchbegriffe mit dem Befehl AND • Stellt ein Bild zur Webseite neben dem Suchtreffer dar • Viele Sucheinstellungen (Sprache, Dateityp, Jahresdatum)	a) 142 b) - c) - d) - e) - f) - g) 2	a) 1.238.990 b) 16.955 c) 500 d) 242 e) 1 f) 1 g) 5
Yan-dex.com	• Besitzt keinen öffentlichen PageRank • Internationale Suchmaschine von Yandex.ru	a) 2 b) 1 c) - d) 2 e) - f) - g) -	a) 796.000 b) 45.000 c) unbekannt d) unbekannt e) unbekannt f) unbekannt g) unbekannt
DeuSu	• Besitzt keinen öffentlichen PageRank • Verknüpft die Suchbegriffe mit dem Befehl AND • Werbefrei, finanziert durch Spenden • Keine vertikale Suche, nur reine Websuche • Deutsche Suchmaschine	a) 1 b) 5 c) - d) - e) - f) - g) -	a) 74.038 b) 2.922 c) 193 d) 46 e) 1 f) 1 g) 1
Hotbot	• PageRank 8 • Verknüpft die Suchbegriffe mit dem Befehl AND • Bietet nur Websuche, News und Wetter • Suchmaschine ist nicht auf deutsch	a) 3 b) 1 c) 1 d) 4 e) 3 f) 1 g) 3	a) 1.450.000 b) 65.600 c) 86.900 d) 51.700 e) 4.540 f) 51.700 g) 9.070

Tabelle 6.6: Ergebnisse des Suchtests mit Volltextsuchmaschinen

Als Nächstes werden die anderen Arten von Suchmaschinen im gleichen Test gegenübergestellt. Da nicht alle Suchmaschinen die Ergebnismenge anzeigen, wurde in diesem Test darauf verzichtet. Nur die Positionen der Suchergebnisse sind hier entscheidend. Die Ergebnisse des Tests sind der Tabelle 6.7 dargestellt.

Die Meta- und Hybridsuchmaschinen bieten in diesem Test sehr gute Ergebnisse. Nur die dezentrale Suchmaschine YaCy konnte nicht überzeugen, dafür sind derzeit noch zu wenige Benutzer beteiligt, um wirklich gute Ergebnisse erzielen zu können. Das Webverzeichnis DMOZ, mit eingebauter Suchfunktion, konnte auch nicht überzeugen. Zwar werden hier die ersten beiden Webseiten gefunden, bei der Suche nach Unterseiten oder der Stundenplan Seite scheitert diese aber auch.

Abschließend ist zu sagen, dass viele der Suchmaschinen auf dem Markt akzeptable Ergebnisse hervorbringen. Dennoch sind sehr viele abhängig von anderen großen Suchmaschinen wie z.B. Google oder Bing, um gute Ergebnisse erzielen zu können. Ohne einen eigenen sehr großen Datenbestand können diese Suchmaschinen nur so gut sein, wie die größten Suchmaschinen auf dem Markt.

Dienst	a	b	c	d	e	f	g	Besonderheit	Page-Rank
Yahoo	1	1	1	3	1	1	1	• Hybridsuchmaschine • Nutzt Suchtreffer von Bing	5
T-Online	1	1	1	1	2	1	2	• Hybridsuchmaschine • Nutzt Suchtreffer von Google	6
Web.de	1	1	1	1	2	1	2	• Hybridsuchmaschine • Nutzt Suchtreffer von Google	7
Fireball	1	1	1	3	1	1	1	• Hybridsuchmaschine • Nutzt Suchtreffer von Bing	6
Metager	2	1	5	5	1	2	1	• Metasuchmaschine	6
Ixquick	1	1	1	5	2	1	2	• Metasuchmaschine • Steht für viel Datenschutz	6
StartPage	1	1	1	1	2	1	2	• Metasuchmaschine • Nutzt nur Suchtreffer von Google • Steht für viel Datenschutz	7
OneSeek	3	1	1	1	2	1	3	• Metasuchmaschine	4
Lycos	1	1	1	3	1	1	1	• Hybridsuchmaschine	6
Duck-DuckGo	1	1	1	3	1	1	1	• Hybridsuchmaschine • Steht für Datenschutz	7
Qwant	1	1	1	3	1	1	1	• Hybridsuchmaschine • Stellt Suchergebnisse in mehreren Spalten dar	5
YaCy	-	-	-	-	-	-	-	• Dezentrale Suchmaschine	4
DMOZ	7	1	-	-	-	-	-	• Webverzeichnis	5

Tabelle 6.7: Ergebnisse des Suchtests mit anderen Arten von Suchmaschinen

6.5 Trends

6.5.1 Google als weltweiter Marktführer

Anschaulich wird in der Abb. 6.1 und Abb. 6.2 gezeigt, dass der Marktführer in Deutschland und weltweit Google mit einem Anteil von ca. 90% ist. Alle anderen Suchmaschinen liegen bei ca. 0,5% bis 6% Marktanteil. Bing liegt mit ca. 4% vor Yahoo mit einem Marktanteil von ca. 3%. In der dargestellten Zeitspanne von 8 Monaten hat sich kaum etwas beim Nutzerverhalten, für die Nutzung der Suchmaschinen, verändert. Auch aus der Abb. 6.3 und Abb. 6.4 lässt sich entnehmen, dass Google stets die Nummer 1 der genutzten Suchmaschinen darstellt.

Daher wird der Trend weitergehend so sein, dass Google in Deutschland und weltweit als klarer Sieger der genutzten Suchmaschinen hervorgeht. Google ist schon sehr lange auf dem Markt und kann auf einen sehr großen eigenen Datenbestand zurückgreifen, neuere Suchmaschinen haben es schwer diesen Datenbestand aufzuholen. Ohne eine neuartige oder disruptive Idee kann keine Suchmaschine gegen Google bestehen.

6.5.2 Konkurrenz in einigen Ländern

Google ist nicht in jedem Land Marktführer. Die beiden Suchmaschinen Baidu und Yandex machen der Suchmaschine Google deutlich Konkurrenz. In China nutzen viele die Suchmaschine Baidu. Diese Suchmaschine ist marktführend in China. Im Jahre 2010 hat sich Google aufgrund der niedrigen Marktanteile vom chinesischen Markt zurückgezogen. In Russland ist die Suchmaschine Yandex der Marktführer. Mit rund 4.000 Angestellten hat sich das Unternehmen von Yandex im Heimatland gut etabliert (vgl. [Zei15]).

Ein Trend könnte dahin gehen, dass landesspezifische Suchmaschinen eingeführt werden. Somit hätte jedes Land seine eigene spezifische Suchmaschine.

6.5.3 Partnerindex-Modell

Frühere große Suchmaschinen wie Yahoo sind von der Idee als reine Volltextsuchmaschine zurückgegangen und nutzen nun das Partnerindex-Modell. Viele kleinere Suchmaschinen werden auf das Partnerindex-Modell der größeren Suchmaschinen zurückgreifen, da dies zunehmend lukrativer und preisgünstiger für diese ist. Daher wird ein möglicher Trend in diese Richtung gehen, dass es nur noch wenige große Suchmaschinen gibt und viele kleinere, die auf dieses Modell zurückgreifen werden (vgl. [Dir15], S.156-158).

6.5.4 Datenschutz

Suchmaschinen wie DuckDuckGo, Ixquick und StartPage stehen für besonderen Datenschutz. So sollen bei der Nutzung keine personenbezogenen Daten erfasst oder Suchprofile erstellt werden. Besonders für Nutzer der Suchmaschine Google ist die Suchmaschine StartPage interessant. Diese Suchmaschine nutzt ausschließlich die Suchergebnisse von

Google unter Einhaltung der Privatsphäre und des Datenschutzes. Dafür ausgezeichnet wurde diese Suchmaschine mit dem europäischen Datenschutz-Gütesiegel.
So könnte ein möglicher Trend dahin gehen, dass immer mehr Nutzer auf Privatsphäre achten und Suchmaschinen mit besserem Datenschutz nutzen werden.

6.5.5 Mobile Suche mit Smartphone

Immer mehr Benutzer von Suchmaschinen nutzen diese auf mobilen Endgeräten wie dem Smartphone. Die Suchmaschine Google wird daher Webseiten höher im Ranking einstufen, die eine nutzerfreundlichere mobile Ansicht ihrer Webseite bieten. Dies soll die Webseiten-Betreiber künftig bemühen ihre mobilen Ansichten der Webseiten zu verbessern[9] (vgl. [Mak15]).
Ein weiterer Trend besteht daher bei der Verbesserung von mobilen Ansichten der Webseiten und bei der Nutzung von Suchmaschinen auf mobilen Geräten. Weitere Suchmaschinen könnten auf den Trend aufspringen und die Webseiten mit einer guten mobilen Ansicht besser in der Ergebnisliste ranken.

[9] Ob eine Webseite für Mobilgeräte optimiert wurde, kann auf der folgenden Webseite überprüft werden: https://www.google.com/webmasters/tools/mobile-friendly/

7 Open-Source-Suchmaschine

In diesem Kapitel werden zwei Open-Source-Suchmaschinen dargestellt, mit denen eine eigene Suchmaschine aufgebaut werden kann. Nach dem Aufbau der Suchmaschine, werden die Webseiten der Hochschule Neubrandenburg indexiert. Anschließend wird nach den Webseiten im Index gesucht.

7.1 Aufbau einer Suchmaschine mit Lucene, Solr und Nutch

Die erste Open-Source-Suchmaschine wird mit den folgenden Projekten aufgebaut. Dies sind Apache Lucene, Solr und Nutch. Durch die Kombination aller drei Projekte kann eine vollwertige Suchmaschine aufgesetzt werden. Zunächst werden die einzelnen Projekte erklärt und dann wird eine Solr-Instanz aufgesetzt und mit dem Web-Crawler Nutch ein Index erstellt.

7.1.1 Apache Lucene

Lucene ist ein Projekt der Apache Software Foundation und hat eine frei verfügbare Programmbibliothek für die Indexierung und Suche von Daten und ist damit eine leistungsstarke, vollfunktionsfähige Text-Suchengine. Das „Application Programming Interface“ (API) bietet die dafür benötigten Schnittstellen und ist bei Internet-Suchmaschinen und webseitenübergreifenden Suchfunktionen weit verbreitet. Entwickelt wurde das Projekt mit der Programmiersprache Java und ist dadurch plattformübergreifend auf vielen Betriebssystemen nutzbar (vgl. [fhw15]).

Lucene besteht aus vier wesentlichen Hauptkomponenten:

- **Lucene Core**: Indizierung, Suche, Rechtschreibprüfung, Hervorheben von Treffern und Tokenisierung
- **Solr**: APIs für Extensible Markup Language (XML), Hypertext Transfer Protocol (HTTP), Javascript Object Notation (JSON), Python und Ruby on Rails sowie Trefferhervorhebung
- **Open Relevance Project**: Frei verfügbares Material für Performance-Tests und Relevanz-Evaluierung.
- **PyLucene**: Python-Portierung von Lucene Core (vgl. [Rou15])

Die Vor- und Nachteile von Lucene sind in der folgenden Tabelle 7.1 zusammengefasst:

Vorteile	Nachteile
+ Open-Source API + Modular, leicht zu integrieren, definierte Schnittstellen + Anpassung von Indexierungs- und Suchfunktionen + Reine Indexierungs- und Suchsoftware + Gut realisiert, effizient und schnell	– Keine Möglichkeit Web-Crawler-ähnliche Funktionalitäten einzubauen – Keine fertige Suchmaschine

Tabelle 7.1: Vor- und Nachteile von Apache Lucene

7.1.2 Apache Solr

Solr ist ein Projekt der Apache Software Foundation und stellt einen Java-Server mit dem entwickelten Suchindex Lucene zur Verfügung. Solr ist sozusagen eine „Serverversion“ von Lucene. Lucene stellt dabei das technische Gerüst dar und wird für den Aufbau, die Ausführung und die Optimierung von Abfragen genutzt. Ansprechende Suchlösungen lassen sich mit seinen REST-Schnittstellen und der XML-basierten Konfiguration realisieren. Der Begriff REST (Representational State Transfer) bezeichnet einen Architekturstil, der dem HTTP-Protokoll zugrunde liegt und mit dem Webservice realisiert werden kann (vgl. [ITW15]).
Auf Basis von Standard-Webtechnologien bietet Solr eine mächtige und leicht integrierbare Suchmaschine für Entwickler. Seit der Version 3.0 wurden Lucene und Solr zusammen entwickelt und seit der Version 3.4 sind die Versionsnummern beider Projekte zusätzlich synchron gehalten (vgl. [Sch151]), (vgl. [com15]).
Solr bietet folgende Funktionalitäten:

- Arbeitet mit HTTP-Anfragen (REST-Schnittstelle)
- Definieren eigener Felder
- Indexieren von XML-Dokumente mit einem POST-Request oder der Datei „post.jar“
- Indexieren sonstiger Dokumente wie z.B. Office Dateien, PDFs, Textdateien, HTML-Dateien und andere Dateiformate
- Indexieren von Datenbanken mit dem DataImportHandler
- Rechtschreibüberprüfungsfunktion
- Autovervollständigung während der Eingabe

7.1.3 Apache Nutch

Nutch ist ein in Java geschriebener Web-Crawler, mit dem die Webseiten im Internet oder Intranet gecrawlt und indexiert werden können. Folgende Projekte werden dabei genutzt:

- Lucene für die Indexierung
- Solr für die Suchfunktionalitäten
- Tika für das Parsen von verschiedenen Dokumenten
- Hadoop zur Skalierung großer Datenmengen
- Gora zur Verbindung mit Solr und Hadoop (vgl. [Gol151])

7.1.4 Systemvoraussetzungen

Folgende Systemvoraussetzungen müssen für den Einsatz von Solr und Nutch erfüllt sein:

- Java (Java Laufzeitumgebung JRE bzw. Java Entwicklungsumgebung JDK 7.0 oder höher)
- Ein Servlet Contrainer wie Tomcat, Jetty, WildFly oder das integrierte Jetty-Servlet
- Eine funktionierende Internetverbindung
- Einen Browser zur Administration

7.1.5 Installation

Für den Einsatz der Suchmaschine werden die beiden Projekte Solr (Version 4.10.4) und Nutch (Version 1.9) benötigt. Eingerichtet werden die beiden Projekte auf dem Betriebssystem Debian 8. Zunächst wird Solr als Java-Applikation installiert. Diese Installation kann in einem Java Servlet Container erfolgen, wie z.B. Jetty oder Tomcat.

Für die Installation mit Jetty muss nur der Service gestartet werden. In Linux erfolgt dies mit dem Shell-Skript „solr“ und in Windows mit dem Batch-Skript „solr.cmd“, die sich im Ordner „bin“ befinden. Zum Ausführen muss der Parameter „start“ übergeben werden.

- `bin/solr start`

7.1.6 Solr und Nutch Konfiguration

Nach der Installation von Solr werden einige Konfigurationen an Solr und Nutch vorgenommen. In Solr wird das bereits existierende Core „collection1“ verwendet. Ein Core ist ein eigenständiger Solr-Index und besteht aus Konfigurationsdaten und dem Lucene Index. Diese Konfigurationen legen fest, wie die Daten zu indexieren sind. Es können Vorgaben für definierte Felder, ein Standardoperator für die Eingabe mehrerer Wörter und viele weitere Einstellungen verändert werden. Jedes angelegte Core besitzt ein eigenes Schema (vgl. [Klo14], S.11 ff.).

Als Erstes muss die Datei „schema-solr4.xml“, die sich im „conf“ Verzeichnis von Nutch befindet, in das Verzeichnis von Solr „solr/example/solr/collection1/conf“ kopiert und in „schema.xml“ umbenannt werden. Außerdem muss diese Datei zwischen dem Feld „fields“ um folgende Zeile ergänzt werden:

- `<field name="_version_" type="long" indexed="true" stored="true"/>`

Die XML-Datei „nutch-default.xml“ beinhaltet alle Standardeinstellungen über den Crawler. Diese Datei sollte nicht verändert werden. Um diese Einträge verändern zu können, muss stattdessen die Konfigurationsdatei „nutch-site.xml“ angepasst werden. Die Tabelle 7.2 zeigt die einzutragenden Konfigurationen an, die zwischen dem Feld „configuration“ eingetragen werden. Der Name des Crawlers wird hier zu „Nutch“ geändert. Greift Nutch im Crawling-Prozess auf eine Webseite zu, so wird der Name in den Log-Files des Servers angezeigt. Außerdem wird eine Wartezeit von 500 Millisekunden festgelegt, die zwischen der erfolgreichen Abfrage einer Webseite gewartet werden muss. Eine zu geringe Zahl könnte einen Server zu stark belasten. Bei der Verwendung einer zu hohen Zahl könnte der Crawling-Prozess sehr lange dauern. Außerdem wird der Timeout-Wert auf 30 Sekunden gesetzt. Dadurch wird versucht möglichst lange eine Webseite zu crawlen, bevor das Crawlen der Webseite abgebrochen wird.

	Code	Erklärung

Name	`<property>`
	`<name>http.agent.name</name>`
	`<value>Nutch</value>`
	`</property>`
		Name des Crawlers muss hier zwingend eingetragen werden.

Warte-zeit [sek.]	`<property>`
	`<name>fetcher.server.delay</name>`
	`<value>0.5</value>`
	`</property>`
		Wartezeit in Sekunden, die nach der erfolgreichen Abfrage einer Webseite gewartet werden muss.

Timeout [ms.]	`<property>`
	`<name>http.timeout</name>`
	`<value>30000</value>`
	`</property>`
		Maximale Zeit in Millisekunden, in der versucht wird eine Webseite zu crawlen, bevor abgebrochen wird.

Menge [bytes]	`<property>`
	`<name>http.content.limit</name>`
	`<value>131072</value>`
	`</property>`
		Mit dieser Option wird die gesamt Menge einer Webseite in Bytes bestimmt.

Tabelle 7.2: Einstellungen in der Konfigurationsdatei von Nutch

Danach muss eine Seed-Liste erstellt werden, in dem die URLs stehen, die als Ausgangsmenge gecrawlt werden sollen. Dazu wird im Verzeichnis von Nutch der Ordner „urls“ erstellt. In diesem Ordner wird dann eine „seed.txt“ Text-Datei angelegt. Danach wird in dieser Text-Datei die URL „http://www.hs-nb.de“ eingetragen. Hier sind mehrere URLs pro Zeile möglich. Mit dieser Einstellung würde der Crawler auch Hyperlinks zu externen Webseiten verfolgen. Um zu erreichen, dass der Crawler ausschließlich Webseiten von der eigenen Domain crawlt, können Filter eingesetzt werden. In der Datei „regex-urlfilter.xml“ wird ein regulärer Ausdruck verwendet, um nur Webseiten mit der Domain „hs-nb.de“ zu erlauben. Dieser reguläre Ausdruck ist folgendermaßen aufgebaut und die Bestandteile des Ausdrucks sind in der Tabelle 7.3 erklärt:

- `+^(http://|https://)(a-z0-9]*\.)*hs-nb.de/`

Zum Schluss muss noch der Pfad zu Java in den Umgebungsvariablen richtig eingetragen werden. Dazu wird folgender Befehl genutzt:

- `export JAVA_HOME=/usr/lib/jvm/default-java`

Ausdruck	Erklärung
+	Der gesamte Ausdruck muss in der URL vorkommen.
^	Der nachfolgende Ausdruck muss am Anfang stehen.
(http://|https://)	Am Anfang muss entweder http:// oder https:// stehen.
([a-z0-9]*\.)*	Ermöglicht zusätzlich Subdomains.
hs-nb.de/	Diesen Ausdruck muss diese Domain zwingend beinhalten.

Tabelle 7.3: Einstellungen eines Regex-Filter zur Konfiguration des Crawlers von Nutch

7.1.7 Web-Crawler

Bevor der Web-Crawler zum Einsatz kommt, muss Solr neu gestartet werden. Damit werden die geänderten Konfigurationen in Solr übernommen. Anschließend kann der Web-Crawler in Nutch gestartet werden. Dies geschieht mit folgendem Befehl:

- `bin/crawl urls hsnb http://localhost:8983/solr/ 999`

Danach beginnt der Web-Crawler die Webseiten zu crawlen und alle Hyperlinks von derselben Domain zu verfolgen. Dabei ließt dieser die „seed.txt“ Text-Datei im Ordner „urls“ aus, erstellt die gecrawlten Daten im Ordner „hsnb“ und schickt diese Daten an die angegebene Solr-Instanz weiter. Der Wert am Ende des Befehls gibt die Crawl-Tiefe an. Ein sehr hoher Wert garantiert, dass alle verlinkten Webseiten gefunden werden können.
In der Tabelle 7.4 sind die Anzahl der Einträge und die benötigte Laufzeit für das Crawling mit einer Crawl-Tiefe von 1, 2, 3 und 999 dargestellt. Bei einer Crawl-Tiefe von 1 wird nur die eingetragene URL in der Seed-Liste gecrawlt. Bei einer Crawl-Tiefe von 2 werden zusätzlich die verlinkten Webseiten von der eingetragenen URL gecrawlt. Dieser Vorgang dauerte 2 Minuten und brachte 45 Einträge im Index. Eine Crawl-Tiefe von 3 hingegen dauerte über 1 Stunde und brachte 3060 Einträge. Eine Crawl-Tiefe von 999 stellt sicher, dass alle verlinkten Webseiten im Crawling-Prozess erreicht werden. Dieser Vorgang dauerte 3 Stunden und brachte anschließend insgesamt 6396 Einträge.

Crawl-Tiefe	Einträge	Laufzeit
1	1	20 Sekunden
2	45	2 Minuten
3	3060	1 Stunde und 20 Minuten
999	6396	3 Stunden

Tabelle 7.4: Anzahl der Einträge und Laufzeit bei variabler Crawl-Tiefe

7.1.8 Suche nach den Indexierten Seiten

Eine Abfrage kann in der Benutzeroberfläche von Solr erfolgen. Dazu wird das User-Interface von Solr, mit der URL „http://localhost:8983/solr“, aufgerufen. Danach wird das Core „collection1“ und der Menüpunkt „Query“ ausgewählt. Durch die vordefinierten Felder können die Anfragen erstellt werden. Diese Abfragen erfolgen dann durch das Aufrufen der URL mit den in der Tabelle 7.5 beschriebenen Parametern.
Eine gültige Abfrage kann folgendermaßen aussehen und liefert eine JSON-Datei mit den Daten für die Suchbegriffe Hochschule und Neubrandenburg zurück:

- http://localhost:8983/solr/collection1/select?q=Hochschule+AND+Neubrandenburg&start=0&rows=50&fl=url+title+content&wt=json&indent=true

Parameter	Erklärung
q	Query String, für die Suchbegriffe.
sort	Hier kann nach einem Feld sortiert werden (Feldname+asc oder desc).
fl	Dieser Parameter schränkt die Ausgabe auf bestimmte Felder ein.
start	Startzahl der Einträge.
rows	Maximale Anzahl der auszugebenden Einträge.
wt	Legt das Rückgabeformat fest (JSON, XML, PYTHON, RUBY, PHP, CSV).
indent	Mit true wird das Ausgabeformat für Menschen leserisch ausgegeben. Die Angabe von false lohnt sich im Produktiven Einsatz, um die Datenmenge geringer halten zu können.

Tabelle 7.5: Parameter für eine GET-Abfrage in Solr

Nun erfolgt, wie im Kapitel 6.4 zuvor, die Suche nach den Webseiten. Dazu wird wieder mit den Begriffen in der Tabelle 7.6 nach den erwarteten Webseiten gesucht. Die Ergebnisse dieser Suche sind in der Tabelle 7.7 dargestellt.

Die Begriffe werden jeweils mit dem Operator AND verknüpft, um die Treffermenge besser einschränken zu können. Standardmäßig werden in Solr die Suchbegriffe mit OR-Operator verknüpft. In der „schema.xml“ von Solr lässt sich dies bei „defaultOperator“ anpassen.

Da nur die Domain „hs-nb.de“ gecrawlt wurde, wird das erwartete Ergebnis für den Suchbegriff Neubrandenburg nicht in der Trefferliste angezeigt. Bei der Suche nach der Hochschule Neubrandenburg befindet sich die erwartete Webseite an der Position 1 in der Trefferliste. Die beiden Unterseiten zum Studiengang GI und GG sind in der Treffermenge an den Positionen 15 und 10. Zum Schluss beginnt die Suche nach der Stundenplan-Webseite. Diese befindet sich in den Treffern lediglich an den Positionen 29, 40 und 222.

Die Hauptseite wurde in Solr schnell gefunden, die beiden Unterseiten waren nicht sehr weit oben vertreten. Die Stundenplan-Webseite war nicht bei den relevanten Treffern dabei, sondern wurden erst ab Treffer 29 angezeigt. Das Ranking von Solr müsste hier noch individuell für die eigenen Bedürfnisse angepasst werden.

Test	Webseite	Suchbegriffe
a	www.neubrandenburg.de	Neubrandenburg
b	www.hs-nb.de	Hochschule Neubrandenburg
c	www.hs-nb.de/studiengang-gi/	Hochschule Neubrandenburg Geoinformatik
d	www.hs-nb.de/studiengang-gg/	Hochschule Neubrandenburg Geoinformatik Master
e	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan
f	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan 1.Semester
g	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Stundenplan

Tabelle 7.6: Die erwartete Webseite und die verwendeten Suchbegriffe für die Suche

Zeitpunkt	Position	Ergebnisse
Nach dem Indexieren von www.hs-nb.de (Begriffe wurden in der Suche mit dem Operator AND verknüpft)	a) - b) 1 c) 15 d) 10 e) 40 f) 29 g) 222	a) 4.888 b) 4.560 c) 535 d) 127 e) 50 f) 39 g) 234

Tabelle 7.7: Ergebnisse des Suchtests nach dem Indexieren mit Solr

7.2 YaCy

7.2.1 Anwendung

Bei der Suchmaschinensoftware YaCy (Yet Another Cyberspace) handelt es sich um eine dezentrale Suchmaschine mit User-Interface, Index-Administration und Monitoring zum Überwachen. Diese Software basiert auf die beiden Projekte Solr und Lucene. Der Anwender wird mit dieser Software gleichzeitig zum Betreiber und Nutzer der eigenen Suchmaschine. Die Indexierungs-Komponenten sind in der Abb. 7.1 dargestellt. Dazu gehört der eigene Web-Crawler, um Dokumente aus dem Internet erfassen, parsen und speichern zu können. Dieser Suchindex wird lokal in einer Datenbank gespeichert. Mit der Websuche kann dann wiederum auf diesen lokalen Suchindex und auch auf einem anderen Suchindex von YaCy-Peers im Netzwerk gesucht werden. Dadurch können die Informationen durch eine Peer-to-Peer-Verbindung mit anderen Nutzern ausgetauscht werden (vgl. [YaC15]).

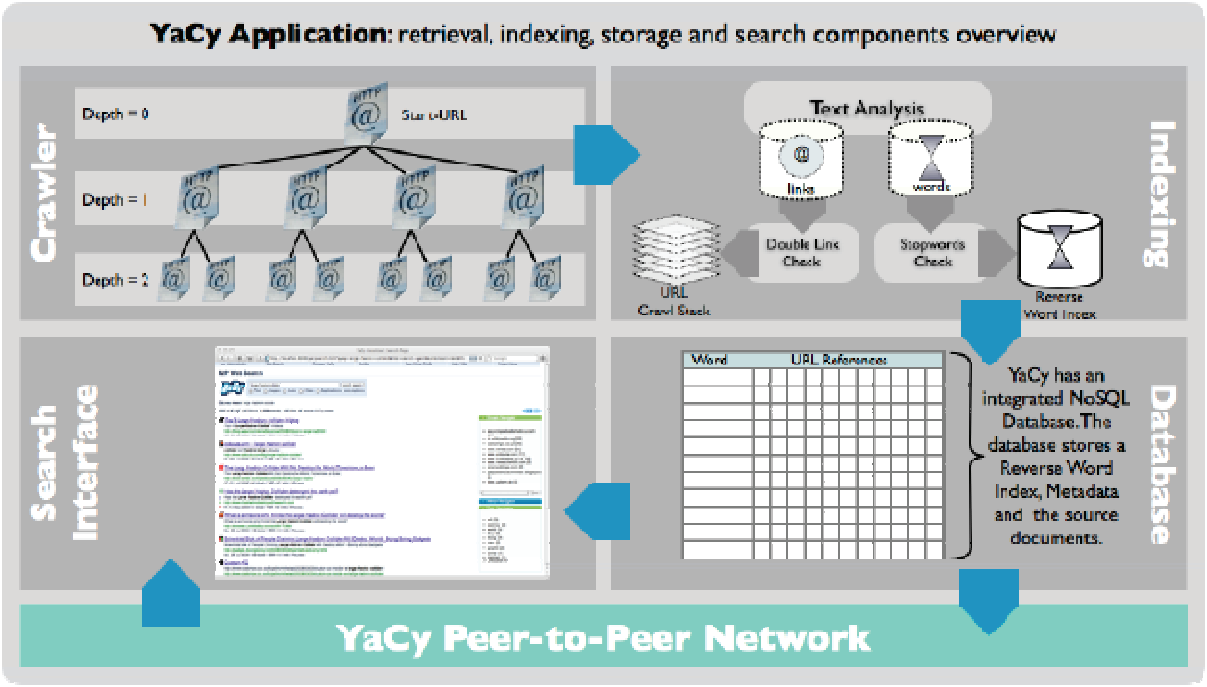


Abb. 7.1: Prinzip der YaCy-Suchmaschine (übernommen von [YaC15])

7.2.2 Systemvoraussetzungen

Folgende Systemvoraussetzungen müssen für den Einsatz von YaCy erfüllt sein:

- Mindestens 1 GB Arbeitsspeicher (empfehlenswert sind 2 GB oder mehr)
- Mindestens 20 GB freier Speicherplatz
- Mindestens 1 GHz Taktfrequenz
- Eine funktionierende Internetverbindung
- Zugriff auf Firewall und Router, um den eigenen Knotenpunkt für andere nutzbar machen zu können

- Java (Java Laufzeitumgebung JRE bzw. Java Entwicklungsumgebung JDK 6.0 oder höher)
- Einen Browser zur Administration (Optimal den IE7/8/9, Firefox ab Version 2, Safari oder Chrome) (vgl. [YaC152])

7.2.3 Installation

Installiert wird diese Suchmaschinensoftware auf dem eigenen Rechner. Für die Installation der Software stehen folgende Betriebssysteme zur Verfügung:[10]

- Windows
- GNU/Linux (außerdem wird OpenJDK7 benötigt)
- Mac OS

Nach der Installation und dem Starten der Anwendung ist diese mit dem Web-Browser auf dem Port 8090 erreichbar.

7.2.4 Konfiguration

Bei der Konfiguration stehen drei verschiedene Möglichkeiten zur Verfügung. Wie in der Abb. 7.2 dargestellt, kann YaCy als „Gemeinschafts-basierte Web Suche“, „Suchportal für eigene Internetseiten“ oder als „Intranet Indexierung“ genutzt werden.

Die „Gemeinschafts-basierte Web Suche“ ermöglicht die Einrichtung einer Suchmaschine mit dem Austausch anderer Peers im Netzwerk „freeworld“. Bei einer Suchanfrage werden die Indices anderer Peers im Netzwerk geladen und mit der Suchanfrage verglichen. Es besteht die Möglichkeit den eigenen Index für andere Peers im Netzwerk freizugeben. Dazu muss der Router so konfiguriert werden, dass der jeweilige Port von außerhalb erreichbar ist.

Bei der Nutzung als „Suchportal für eigene Internetseiten“ dient die YaCy-Installation ausschließlich als Suchfunktion für die eigenen indexierten Webseiten. Es wird nur auf den eigenen Index gesucht. Andere Peer-Nutzer des öffentlichen Netzwerks „freeworld“ erhalten zusätzlich die Ergebnisse dieser Webseiten.

Die „Intranet Indexierung“ ermöglicht das Erreichen aller Seiten des Intranets mit der Suchfunktion. Dadurch können Dokumente auf lokalen Datenträgern gefunden und indexiert werden. Außerdem sperrt diese Option die Verbindung zum Peer-to-Peer Netzwerk, sodass nur auf dem lokalen Index gesucht werden kann (vgl. [YaC151]).

[10] Eine genaue Installationsanleitung, für das jeweilige Betriebssystem, befindet sich auf der Homepage von YaCy: http://www.yacy.net/de/index.html

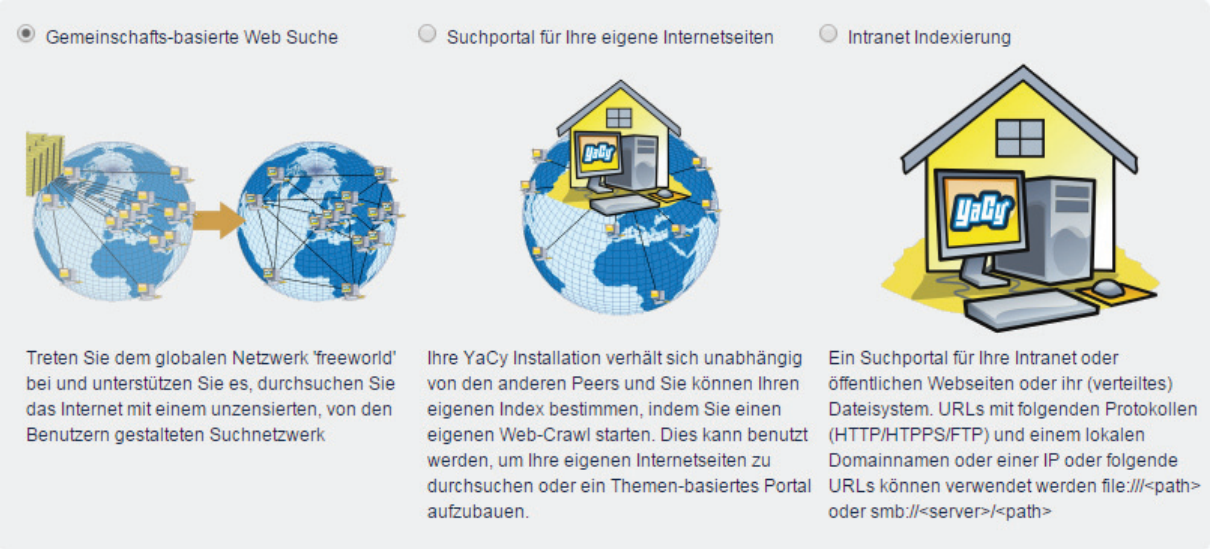


Abb. 7.2: YaCy-Anwendungsfall

7.2.5 Web-Crawler

Mit dem integrierten Web-Crawler von YaCy können die Webseiten indexiert werden. Im Menüpunkt „Webseiten laden mit Crawler“ kann die URL für den Crawler-Prozess angegeben werden.

Vor dem Start können noch weitere Einstellungen vorgenommen werden. Es kann eine Einschränkung für die Anzahl der maximalen zu indexierenden Dokumente angegeben werden und es kann entschieden werden, ob alle Dateien der URL oder nur Dateien der Unterpfade indexiert werden sollen. In der Einstellung „Parser Configuration“ können außerdem die verschiedenen Dateiformate für den Parser eingestellt werden. Mit dem einfachen Crawling werden nur die Webseiten von einer angegebenen Domain gecrawlt.

Weiterhin existiert ein „Experten Crawler“, mit dem noch weitere Feineinstellungen vorgenommen werden können. Es kann eine Crawl-Tiefe bestimmt werden. Bei der Crawl-Tiefe 0 wird damit nur diese eine Webseite indexiert, während bei einer höheren Crawl-Tiefe auch die verlinkten Seiten indexiert werden. Außerdem können Filter erstellt werden, um den Crawler auf bestimmte URLs zu beschränken oder um diese auszuschließen (vgl. [YaC153]).

7.2.6 Crawling der Webseiten

Im nachfolgenden Schritt wird die Webpräsenz der Hochschule Neubrandenburg gecrawlt. Dazu wird zunächst der Web-Crawler auf die Seite „www.hs-nb.de“ eingestellt und der Crawling-Prozess gestartet. Pro Sekunde werden maximal 2 Seiten geladen, dieser Prozess dauert je nach Größe der Webpräsenz einige Zeit.

In diesem Test wurden 7.722 Dokumente gefunden und der Crawler produzierte einen Daten-Traffic von 681,57 MB. Der Crawling-Prozess hat im Test etwa 2 Stunden gedauert.

7.2.7 Ranking-Faktoren

In der Kategorie „Ranking und Heuristiken“ können die Faktoren für das Ranking verändert werden. In der „Boost Abfrage“ sind standardmäßig die Werte „crawldepth_i:0^0.8 crawldepth_i:1^0.4“ eingetragen. Durch das Verändern der Werte können Einträge mit einer geringeren Crawling-Tiefe besser gerankt werden als höhere. Außerdem können Webseiten mit bestimmten Keywords, Titel, Wörter usw. besser gerankt werden. Weiterhin können viele weitere Faktoren, die Einfluss auf das Ranking nehmen, angepasst werden. So kann die Gewichtung für den Titel höher gewichtet werden, als die Überschriften auf der Webseite.
Außerdem gibt es noch sogenannte „Pre-Ranking-Faktoren“. Diese nehmen auch Einfluss auf das Ranking. Alle Faktoren erhalten einen Wert von 0 bis 15. Durch Faktoren wie die Domain- und URL-Länge mit dem Wert 15, können Webseiten mit einer kürzeren URL bevorzugt werden.

7.2.8 Suche nach den Indexierten Seiten

Es wird, wie im Kapitel 6.4 zuvor, nach den Webseiten in der Tabelle 7.8 mit den dazugehörigen Suchbegriffen gesucht. Diese Suche geschieht einmal vor und nach dem Crawling-Prozess. Die dazu gehörigen Ergebnisse befinden sich in der Tabelle 7.9.
Zunächst ist zu erwähnen, dass sich aufgrund der laufend veränderten Anzahl von Peers, auch die Suchmenge verändern und die Position der Suchtreffer in der Ergebnisliste variieren kann. Um dies verhindern zu können, müsste die Suche auf den eigenen Index beschränkt oder auf die Verbindung zum öffentlichen Peer Netzwerk verzichtet werden.
In diesem Test wurde das öffentliche Peer-Netzwerk „freeworld“ verwendet. Vor dem Indexieren der Webpräsenz wurde nur die Startseite der Hochschule Neubrandenburg an der Position 14 gefunden. Diese befand sich im Index der anderen Peers. Alle anderen Webseiten wurden nicht mit den eingegebenen Suchbegriffen gefunden.
Erst nach dem Crawling-Prozess wurden weitere Webseiten bei der Suche entdeckt. Dazu gehören die beiden Unterseiten vom Studiengang GI und GG. Beide Webseiten sind unter den ersten 10 Ergebnissen zu finden. Die Startseite der Webseite rutschte danach in der Ergebnisliste stark ab. Dies ist zurückzuführen auf ein schlechtes Ranking und die große Anzahl an neuen Webseiten im Index. Die Stundenplan-Webseite wurde erst im dritten Suchvorgang gefunden. Zu viele Suchbegriffe führten dazu, dass diese nicht entdeckt wurde. Im dritten Suchvorgang war diese Webseite stets auf der ersten Position.
Als Nächstes werden kleine Änderungen an den Ranking-Faktoren vorgenommen. Die beiden Faktoren „Domain Length“ und „URL Length“ werden auf den Wert 15 erhöht. Danach wird wieder nach den Webseiten gesucht. Es zeigt sich, dass nun die Startseite auf Position 3 angezeigt wird, da diese eine sehr kurze URL besitzt. Auch die Position der zweiten Unterseite vom Studiengang GG hat sich geringfügig verbessert. Die Stundenplan-Webseite blieb weiterhin im dritten Suchvorgang auf der ersten Position.

Test	Webseite	Suchbegriffe
a	www.neubrandenburg.de	Neubrandenburg
b	www.hs-nb.de	Hochschule Neubrandenburg
c	www.hs-nb.de/studiengang-gi/	Hochschule Neubrandenburg Geoinformatik
d	www.hs-nb.de/studiengang-gg/	Hochschule Neubrandenburg Geoinformatik Master
e	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan
f	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Master Stundenplan 1.Semester
g	user.hs-nb.de/~stundenplan/	Hochschule Neubrandenburg Geoinformatik Stundenplan

Tabelle 7.8: Die erwartete Webseite und die verwendeten Suchbegriffe für die Suche

Zeitpunkt	Position	Ergebnisse
Vor dem Indexieren	a) - b) 14 c) - d) - e) - f) - g) -	a) 724 b) 93 c) 8 d) 8 e) 1 f) 1 g) 1
Nach dem Indexieren von www.hs-nb.de	a) - b) 64 c) 4 d) 10 e) - f) - g) 1	a) 11.422 b) 10.828 c) 1.029 d) 262 e) 54 f) 9 g) 145
Nach dem Indexieren von www.hs-nb.de mit geänderten Ranking-Faktoren • **„Domain Length“ auf 15 erhöht** • **„URL Length“ auf 15 erhöht**	a) - b) 3 c) 5 d) 4 e) - f) - g) 1	a) 11.422 b) 10.828 c) 1.029 d) 262 e) 54 f) 9 g) 145

Tabelle 7.9: Ergebnisse des Suchtests mit YaCy

7.3 Fazit

Durch die Kombination von Solr und Nutch kann sehr einfach und schnell eine eigene Suchmaschine aufgebaut werden. Diese Suchmaschine ist komplett in Java gehalten und kann dadurch auf vielen Betriebssystemen eingesetzt werden. Solr ist hier die tragende Komponente und kümmert sich um die Suche im Index, während Nutch lediglich für das Crawlen der Webseiten zuständig ist. Nutch bietet jede Menge Einstellungen, mit denen die Webseiten sehr einfach indexiert werden können. Der Index für die Webpräsenz der Hochschule Neubrandenburg wurde mit Nutch nach 3 Stunden erstellt und durch die REST-Schnittstelle von Solr konnten die Webseiten schnell gefunden werden. Die Ergebnisse zeigen eine solide Platzierung der Suchergebnisse an. Nur bei der Suche nach der Stundenplan-Webseite könnte das Ranking noch optimiert werden. Außerdem werden bei der Suche jede Menge Operatoren wie z.B. AND, OR und NOT unterstützt, wodurch noch gezielter nach Webseiten gesucht werden kann. Solr bietet keine integrierte Trefferliste an, sondern lediglich die gesuchten Daten als Dateiformat. Daher müsste eine eigene Trefferausgabe dafür implementiert werden.

Die Suchmaschine YaCy basiert auf Solr und lässt sich sehr leicht installieren. Auch diese Suchmaschine ist komplett in Java gehalten und kann auf vielen Betriebssystemen eingesetzt werden. Nach der Installation kann auch sofort nach den Webseiten gesucht werden. Diese Indices befinden sich lediglich auf anderen Peers im Netzwerk „freeworld“. Da ständig neue Peers hinzukommen, kann sich die Treffermenge täglich unterscheiden. Im Test zeigte sich, dass das Projekt noch nicht weit fortgeschritten ist, was die Anzahl an aktiven Peers betrifft. Nur eine der gesuchten Webseiten wurde im Test gefunden. Erst nach dem Indexieren der Webpräsenz der Hochschule Neubrandenburg wurden bessere Ergebnisse erzielt. YaCy eignet sich gut als Suchmaschine, jedoch müssen viel mehr Peers aktiv zum Suchmaschinen-Netz beitragen, damit diese auch viele Webseiten im Internet finden kann. Unterstützt werden bei der Suche keine Operatoren wie AND, OR und NOT, wodurch die Suche nach Webseiten nicht gezielt eingeschränkt werden kann. Jedoch gibt es Einschränkungen an der Seite, die direkt gewählt werden können. Dazu gehört, dass Suchergebnisse auf eine bestimmte Domain, Datentyp, Sprache oder Autor beschränkt werden können.

Beide Open-Source-Suchmaschinen eigenen sich gut, um eine eigene Suchmaschine aufzubauen. YaCy ist dafür jedoch empfehlenswerter, da es die Komplettlösung in einem Paket anbietet. Dadurch ist diese Suchmaschine schnell einsatzfähig und kann durch viele verschiedene Einstellungen stark angepasst werden. Darüber hinaus bietet diese Software viele weitere Funktionen und individuelle Einsatzmöglichkeiten an.

8 Zusammenfassung und Ausblick

Im abschließenden Kapitel werden noch einmal die Ergebnisse der Arbeit zusammengetragen und ein Ausblick erstellt.

8.1 Faktoren für die Wichtigkeit einer Suchmaschine

Das Prinzip einer Suchmaschine ist notwendig, um Informationen im Web gezielt suchen zu können. Eine gute Suchmaschine zeichnet sich durch die Kombination aus einem großen Index und einem guten Ranking aus. Je mehr Daten im Index sind und je vollständiger das Internet damit abgedeckt ist, desto hochwertiger sind die Suchergebnisse. Ein weiterer entscheidender Faktor ist das Ranking. Gut gerankte Suchergebnisse ermöglichen, dass relevante Webseiten auf den vorderen Positionen stehen. Bei einem schlechten Ranking können relevante Webseiten nicht sofort gefunden werden. Das Ranking der Suchergebnisse wird durch präzise Algorithmen realisiert. Diese Bewertung hängt von einer Menge bestimmter Faktoren ab. Auch das Verlinken der Webseiten untereinander hat einen Einfluss auf die Relevanz einer Webseite. Je häufiger auf eine Webseite verlinkt wurde, desto stärker ist diese Relevanz. Diese Algorithmen sind der Kern einer Suchmaschine und werden von vielen Suchmaschinen streng geheim gehalten, um das Ausnutzen ihrer Suchmaschinen zu verhindern. Wie gut eine Suchmaschine ist, hängt also nicht immer nur von der Menge der Webseiten im Index, sondern auch immer von deren Algorithmen und Ranking ab.

8.2 Steuern von Suchmaschinen für Webseitenbetreiber

Webseitenbetreiber können sich an Richtlinien orientieren, damit Suchmaschinen gezielt ihre Webseiten indexieren. Dies ist erforderlich, falls Barrieren auf den Webseiten vorhanden sind, die den Crawler-Prozess verhindern würden. Außerdem kann dadurch verhindert werden, dass Suchmaschinen bestimmte Webseiten indexieren. Diese Richtlinien dienen nur zur Orientierung und müssen von Suchmaschinenbetreibern nicht eingehalten werden. Die meisten und größten Suchmaschinen halten sich jedoch an diese Richtlinien.

8.3 Der Suchmaschinenmarkt

Der Suchmaschinenmarkt bietet viele verschiedene Arten von Suchmaschinen. Die Volltextsuchmaschine besitzt einen eigenen Index und ist von keiner anderen Suchmaschine abhängig. Eine Volltextsuchmaschine mit großem Index und guten Algorithmen eignet sich am Besten für eine Websuche. Werden Informationen zu einem speziellen Themengebiet gesucht, bietet sich die Benutzung einer Spezialsuchmaschine an. Metasuchmaschinen vereinen viele verschiedene Suchmaschinen und könnten nützlich sein, wenn nach einer Vielfalt an Informationen gesucht wird. Diese könnten aber wiederrum im Ranking schlechtere Ergebnisse liefern. Hybridsuchmaschinen bestehen aus mehreren Arten von Suchmaschinen und bieten, ähnlich wie Metasuchmaschinen, mehr oder weniger gute Ergebnisse. Dezentra-

le Suchmaschinen gehen einen komplett anderen Weg. Diese Suchmaschinenart besteht aus vielen Teilnehmern, die jeweils einen kleinen Teil vom Internet indexiert haben. Je größer die Anzahl an Mitwirkenden, desto bessere Suchergebnisse lassen sich erzielen. Diese Suchmaschinenart hängt sehr stark von der Anzahl aktiver Anwender ab und bietet lediglich eine Alternative.

Die Marktanalyse zeigt, dass 90% der Menschen auf die Suchmaschine Google vertrauen. Im Test hatte diese Suchmaschine auch sehr gute Ergebnisse geliefert. Die Suchmaschine Bing hat ähnlich gute Ergebnisse gezeigt. Dennoch nutzen hier nur ca. 5% aller Menschen diese Suchmaschine. Andere Volltextsuchmaschinen konnten insgesamt nicht überzeugen. Entweder waren zu wenige Webseiten im Index oder das Ranking war nicht überzeugend. Suchmaschinenarten wie Meta- und Hybridsuchmaschinen haben im Test, bei der Position der Suchtreffer, überzeugt. Dies ist aber nicht weiter verwunderlich, da diese unter anderem auch auf die Suchergebnisse von Google oder Bing zugreifen.

Die größten Lieferer für Suchmaschinentreffer auf dem Markt sind Google und Bing. Wer nach Informationen im Internet sucht, nutzt deshalb am Besten die Suchmaschine Google oder Bing. Auch Meta- und Hybridsuchmaschinen sind dafür gut geeignet. Bei der Suche nach einem bestimmten Themengebiet bietet sich eine Spezialsuchmaschine an. Der Suchmaschinenmarkt verändert sich ständig. Immer wichtiger ist die Frage nach dem Datenschutz. Nutzer, die unsicher bei der Weitergabe der Daten sind, nutzen am besten die Suchmaschine Startpage, Ixquick oder DuckDuckGo.

8.4 Eigene Suchmaschine nutzen

Eine weitere Möglichkeit nach Informationen im Internet zu suchen, ist eine selbstaufgebaute Suchmaschine. Hier wurden die beiden Open-Source-Suchmaschinen Solr in Kombination mit Nutch und die Suchmaschinensoftware YaCy verwendet. Beide eignen sich für den Einsatz einer eigenen Suchmaschine. Während Solr und Nutch getrennt in Suchsoftware und Web-Crawler sind und die Trefferliste nur als Dateiformat ausgibt, bietet YaCy das Komplettpaket. Bei Solr eignet sich die Suche mit Operatoren wie z.B. AND, OR und NOT, diese werden bei YaCy jedoch nicht unterstützt. Stattdessen können Einschränkungen auf die Domain, den Datentyp, die Sprache oder dem Autor unternommen werden. Beim Ranking der Suchergebnisse war die Suchmaschine YaCy deutlich besser und bietet jede Menge Einstellmöglichkeiten bei den Ranking-Faktoren an.

YaCy eignet sich als Suchsoftware durch die Vielfalt an Einstellungen, dem Komplettpaket und die schnelle Einsetzbarkeit. Solr und Nutch sind für Entwickler interessant, die selbst eine eigene Suchlösung erstellen wollen.

8.5 Ausblick

Die Ergebnisse und Vergleiche der Suchmaschinen beschränken sich auf wenige gesuchte Webseiten der Hochschule Neubrandenburg und der Stadt Neubrandenburg. Für einen deutlichen Unterschied in den Ergebnissen und der Qualität der Suchergebnisse müsste nach vielen weiteren thematisch unterschiedlichen Webseiten gesucht werden. Dabei müsste auch der Informationsgehalt der gesuchten Webseite analysiert werden. Weiterhin wäre noch zu untersuchen, wie sehr sich die Nutzung einer Suchmaschine mit mobilen Geräten zu einer Suche am PC unterscheidet.

Glossar

API	Das „Application Programming Interface“ ist eine Schnittstelle für den Zugriff auf die eigenen Funktionen.
Cluster	Ein Verbund von Rechnern zur Steigerung der Rechenleistung und Ausfallsicherheit.
Crawling	Bezeichnet den automatischen Vorgang, bei dem ein Computerprogramm Dokumente im Web findet.
Deep Web	Bezeichnet ein verstecktes Web im Internet, das für Suchmaschinen nicht zugänglich ist.
Dokument	Ein Text, Bild o. ä. das sich im Binärcode auf einem Dateisystem abgelegt befindet.
Domain	Ist ein weltweit eindeutiger Name einer Webpräsenz.
Dubletten	Gleiche Inhalte unter verschiedenen Adressen (URLs).
GFS	Das Cluster-Dateisystem (Global File System), ermöglicht die gleichzeitige Nutzung des Speichers auf mehreren Rechnern.
Hashtag	Ein Schlüssel- oder Schlagwort um Themen zu finden.
HDFS	Ein Dateisystem zur Speicherung von sehr großen Datenmengen auf mehreren Rechnern.
HTML	Textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente.
Hyperlinks	Querverweis zu einem anderen elektronischen Dokument oder Webseite.
Indexierung	Inhaltliche Erschließung von Dokumenten, um diese schneller finden zu können.
Java	Objektorientierte und plattformunabhängige Programmiersprache.
Monitoring	Bezeichnet das Überwachen von Vorgängen.
NoSQL	Bezeichnet Datenbanken, die nicht dem relationalen Ansatz verfolgen.
N-Gramm	Zeichenfolgen, die sich im Text wiederholen.
Onebox	Separate Anzeigebox, in der Ergebnisse einer anderen Kategorie zu einem Suchbegriff passend angezeigt werden.
Open Directory Projekt	Das größte von Menschen gepflegte Webverzeichnis des Internets.

PageRank	Dient als Maß für die Verlinkung von Webseiten. Der Google PageRank kann zwischen einem Wert von 0 und 10 liegen.
Parsen	Daten werden analysiert, segmentiert und codiert.
Peer-to-Peer	Eine Rechner zu Rechner Verbindung, bei der alle Computer gleichberechtigt sind. Dienste können zur Verfügung gestellt, sowie beansprucht werden.
Ranking	Suchergebnisse werden in einer Ergebnisliste nach Relevanz sortiert.
REST	„Representational State Transfer“ bezeichnet eine Schnittstelle eines Webservices.
Robots.txt	Eine Datei, die Anweisungen zur Indexierung und zum Ausschluss von Inhalten einer Webseite für Suchmaschinen Crawler enthält.
Seet Set	Sammlung von Webseiten, dienen als Startpunkte für den Crawling-Prozess.
Servlet	Bezeichnet das Nutzen von Java-Klassen innerhalb des Webservers.
Sitemaps	Alle URLs einer Webpräsenz zusammengefasst in einem Dokument. Detaillierte Empfehlung für den Crawling-Prozess, um Barrieren durch JavaScript o.ä. zu überwinden.
Snippet	Kurzer Textauszug aus einer Webseite zum Anzeigen in der Ergebnisliste einer Suchmaschine.
Spider Traps	Fallen durch dynamische Inhalte im Web für den Crawling-Prozess.
SQL	Bezeichnet eine Datenbanksprache für relationale Datenbanken.
Textanzeigen	Anzeigen-Werbung die von einem Werbenden geschaltet wurde.
URL	Bezeichnung für Netzwerkressourcen.
Webpräsenz	Bezeichnung für einen Internetauftritt, einen virtuellen Platz im Internet, an dem sich mehrere Webseiten und Dokumente befinden.
Webseite	Eine Seite die mit Angabe einer URL aufgerufen werden kann.
Wildcard	Platzhalter für ein oder mehrere andere Zeichen.
XML	Die „Extensible Markup Language“ ist eine Auszeichnungssprache zur Darstellung von hierarchisch strukturierten Daten.

Quellenverzeichnis

[Ber15] Bertran, Pere Ferrera; Datasalt; http://www.datasalt.com/2012/02/tuple-mapreduce-beyond-the-classic-mapreduce/ (21.05.2015).

[Brä15] Brätz, Marcel; Kryptographiespielplatz; https://www.kryptographiespielplatz.de/index.php?aG=4c5a277c9581f415b57e54bfa90a2feba468bdc6 (02.07.2015).

[com15] comundus; http://www.comundus.com/produkte/solr-enterprise-search/ (06.07.2015).

[Dav15] Linden, David; SeoSweet; https://www.seosweet.de/blog/2014/06/03/horizontale-und-vertikale-suche-seo-grundlagen/ (06.05.2015).

[Dav151] Linden, David;. SeoSweet; https://www.seosweet.de/blog/2012/01/05/google-trefferliste-universal-search/ (08.05.2015).

[Die15] Dietl, Marcell; Seibert-Media; https://blog.seibert-media.net/blog/2011/08/31/nosql-datenbanken-theorie-und-praxis/ (12.08.2015).

[Dir15] Lewandowski, Dirk; Suchmaschinen verstehen; s.l. : Springer-Verlag, 2015. 978-3-662-44013-1.

[eFa15] eFactory; http://pr.efactory.de/d-pagerank-algorithmus.shtml (09.05.2015).

[Enr15] Lauterschlag, Enrico; Webschmoeker; http://www.webschmoeker.de/grundlagen/was-ist-eine-suchmaschine/ (20.04.2015).

[fhw15] fh-wedel; http://www.fh-wedel.de/~si/seminare/ws02/Ausarbeitung/e.lucene/1.html (06.07.2015).

[Gol15] Golem; http://www.golem.de/1111/88040.html (29.04.2015).

[Gol151] Golem; http://www.golem.de/news/nutch-2-0-freie-suchmaschine-mit-flexiblem-datenbank-backend-1207-93099.html (06.07.2015).

[Goo15] Google Support; https://support.google.com/websearch/answer/2466433?p=adv_operators&hl=de&rd=1 (24.05.2015).

[Hol15] Dambeck, Holger; Spiegel Online; http://www.spiegel.de/wissenschaft/mensch/numerator-wie-google-mit-milliarden-unbekannten-rechnet-a-646448-2.html (11.05.2015).

[ITW15] ITWissen; http://www.itwissen.info/definition/lexikon/representational-state-transfer-REST.html (09.07.2015).

[ITW151] ITWissen; http://www.itwissen.info/definition/lexikon/Internet-Internet.html (10.07.2015).

[Joo15] Joos, Thomas; Bigdata-insider; http://www.bigdata-insider.de/infrastruktur/articles/472678/ (12.08.2015).

[Kla15] Patzwaldt, Klaus; at-web; http://www.at-web.de/blog/20080612/suchmaschinen-mehr-gemeinsamkeiten-fur-robots-richtlinien.htm (20.04.2015).

[Klo14] Klose, Markus und Wrigley, Daniel; Einführung in Apache Solr; s.l. : O'Reilly Verlag, 2014. 978-3-95561-421-8.

[Mak15] Makino, Takaki, Jung, Chaesang und Phan, Doantam; Google Webmaster Center Blog; http://googlewebmastercentral.blogspot.de/2015/02/finding-more-mobile-friendly-search.html (02.06.2015).

[Mar15] Bari, Mario Di; Seo-Summary; http://www.seo-summary.de/suchmaschinen-liste-marktanteile-deutschland/#abh_about (09.04.2015).

[Mar151] Martens, Andreas; Universität Paderbohn; http://is.uni-paderborn.de/fileadmin/Informatik/AG-Engels/Lehre/WS0809/SE/Sonstiges/Seminar/Version1.0/Seminar.NAQ.Andreas.Martens.v1.0.pdf (20.05.2015).

[Mic15] Sattler, Michael; Suchmaschinenoptimiertung leicht gemacht; http://suchmaschinenoptimierung.michaelsattler.de/pagerank.html (12.05.2015).

[Onl15] Online Shop Manager; http://www.onlineshopmanager.de/online-marketing-tipps/suchmaschinenoptimierung-seo-tipps/welche-anforderungen-sind-suchmaschinen-zu-stellen/ (20.04.2015).
[OnP15] OnPageWiki; https://de.onpage.org/wiki/Wayback_Machine (27.04.2015).
[OnP151] OnPageWiki; https://de.onpage.org/wiki/PageRank (09.05.2015).
[Rou15] Rouse, Margaret; Searchenterprisesoftware; http://www.searchenterprisesoftware.de/definition/Apache-Lucene (06.07.2015).
[Sch15] Schreiben10; http://www.schreiben10.com/referate/Informatik/4/Aufbau-einer-normalen-Suchmaschine-reon.php (05.05.2015).
[Sch151] Schlichtholz, Daniel; Mayflower Blog; https://blog.mayflower.de/755-Schnelle-Volltextsuche-mit-Solr.html (06.07.2015).
[Tay15] Lane, Taylor's; StatCounter; http://gs.statcounter.com/ (09.04.2015).
[Taz15] Taz; http://www.taz.de/!82759/ (29.04.2015).
[Uni15] Universitätsbibliothek Bielefeld; http://www.ub.uni-bielefeld.de/biblio/search/help/nutzen.htm (11.04.2015).
[YaC15] YaCy; http://yacy.de/de/Technik.html (29.04.2015).
[YaC151] YaCy-Websuche; http://www.yacy-websuche.de/wiki/index.php/De:Anwendungen (01.07.2015).
[YaC152] YaCy; http://www.yacy-websuche.de/wiki/index.php/De:Requirements (03.07.2015).
[YaC153] YaCy; http://www.yacy-websuche.de/wiki/index.php/De:CrawlStart_p (03.07.2015).
[Zei15] Zeit; http://www.zeit.de/digital/internet/2013-02/baidu-yandex-expansion (02.06.2015).

Abbildungsverzeichnis

Formelverzeichnis

Tabellenverzeichnis

Anhang

Anhang A – Berechnung des PageRanks durch ein Gleichungssystem

Dies ist der Anhang für das Beispiel aus dem Kapitel 4.3.1. Folgendes Gleichungssystem entsteht in diesem Beispiel (1). Für d wird der Dämpfungsfaktor 0,85 eingesetzt (2). Danach wird das Gleichungssystem vereinfacht dargestellt (3). Der Ausdruck für PR(A) wird in die Gleichung von PR(B) und in PR(C) eingesetzt (4). Danach wird der Ausdruck für PR(B) in die Gleichung von PR(C) eingesetzt (5). Zum Schluss wird das Gleichungssystem gelöst und der PageRank für die Seiten A,B und C sind berechnet (6).

$$PR(A) = (1-d) + d(PR(C))$$

1. $$PR(B) = (1-d) + d\left(\frac{PR(A)}{2}\right)$$

$$PR(C) = (1-d) + d\left(\frac{PR(A)}{2} + PR(B)\right)$$

$$PR(A) = 0{,}15 + 0{,}85PR(C)$$

2. $$PR(B) = 0{,}15 + 0{,}85\left(\frac{PR(A)}{2}\right)$$

$$PR(C) = 0{,}15 + 0{,}85\left(\frac{PR(A)}{2} + PR(B)\right)$$

$$PR(A) = 0{,}15 + 0{,}85PR(C)$$

3. $$PR(B) = 0{,}15 + 0{,}425PR(A)$$

$$PR(C) = 0{,}15 + 0{,}425PR(A) + 0{,}85PR(B)$$

$$PR(A) = 0{,}15 + 0{,}85PR(C)$$

4. $$PR(B) = 0{,}15 + 0{,}425(0{,}15 + 0{,}85PR(C))$$

$$PR(C) = 0{,}15 + 0{,}425(0{,}15 + 0{,}425PR(C)) + 0{,}85PR(B)$$

$$PR(A) = 0{,}15 + 0{,}85PR(C)$$

5. $$PR(B) = 0{,}15 + 0{,}425(0{,}15 + 0{,}85PR(C))$$

$$PR(C) = 0{,}15 + 0{,}425(0{,}15 + 0{,}85PR(C)) + 0{,}85(0{,}15 + 0{,}425(0{,}15 + 0{,}85PR(C)))$$

$$PR(A) = \frac{2058}{1769} = 1{,}163369$$

6. $$PR(B) = \frac{1140}{1769} = 0{,}644432$$

$$PR(C) = \frac{2109}{1769} = 1{,}192199$$

Anhang B – Iterative Berechnung des PageRank

Dies ist der Anhang für das Beispiel aus dem Kapitel 4.3.2. Diese Tabelle soll die Iterationsschritte von 0 bis 40 mit dem Anfangswert 1 darstellen.

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0,575	1,425
2	1,36125	0,575	1,06375
3	1,054188	0,728531	1,217281
4	1,184689	0,598030	1,217281
5	1,184689	0,653493	1,161818
6	1,137545	0,653493	1,208962
7	1,177618	0,633457	1,188926
8	1,160587	0,650487	1,188926
9	1,160587	0,643249	1,196164
10	1,166739	0,643249	1,190011
11	1,161510	0,645864	1,192626
12	1,163732	0,643642	1,192626
13	1,163732	0,644586	1,191682
14	1,162929	0,644586	1,192484
15	1,163612	0,644245	1,192143
16	1,163322	0,644535	1,192143
17	1,163322	0,644412	1,192267
18	1,163427	0,644412	1,192162
19	1,163337	0,644456	1,192206
20	1,163375	0,644418	1,192206
21	1,163375	0,644435	1,192190
22	1,163362	0,644435	1,192204
23	1,163373	0,644429	1,192198
24	1,163368	0,644434	1,192198
25	1,163368	0,644432	1,192200
26	1,163370	0,644432	1,192198
27	1,163369	0,644432	1,192199
28	1,163369	0,644432	1,192199
29	1,163369	0,644432	1,192199
30	1,163369	0,644432	1,192199
31	1,163369	0,644432	1,192199
32	1,163369	0,644432	1,192199
33	1,163369	0,644432	1,192199
34	1,163369	0,644432	1,192199
35	1,163369	0,644432	1,192199
36	1,163369	0,644432	1,192199
37	1,163369	0,644432	1,192199
38	1,163369	0,644432	1,192199
39	1,163369	0,644432	1,192199
40	1,163369	0,644432	1,192199

Anhang C – Inhalt der CD

- Kopie der Bachelorarbeit im PDF-Format

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Neubrandenburg, den

Unterschrift