DFN Projekt WebSearchBench

Entwicklung und Erprobung innovativer Basistechnologien für eine skalierbare, intelligente Internet Suchmaschine

**Prof. Dr.-Ing. Christoph Lindemann,
Marco Lohmann, Oliver P. Waldhorst,
Christian Welz**

Abschlußbericht

Universität Dortmund
Informatik IV
-Rechnersysteme und Leistungsbewertung-
August-Schmidt-Str. 12
44227 Dortmund
http://www4.cs.uni-dortmund.de/~Lindemann/

29. Oktober 2003

Inhaltsverzeichnis

1 Einleitung

Aufgrund des explosionsartigen Wachstums in den letzten Jahren hat sich das World Wide Web (WWW) zu einer Informationsmatrix von nahezu unvorstellbarer Größe entwickelt. Schätzungen zufolge besteht das öffentlich verfügbare (d.h. das von einer Suchmaschine indizierbare) Web aus mehr als 10 Milliarden Dokumenten. Das gesamte Datenvolumen der derzeit öffentlich verfügbaren Web Dokumente wird auf über 100 Terabyte geschätzt. Jeden Monat ändert sich ein Datenvolumen von ca. 700 Gigabytes. Die Transformation der nahezu unendlichen Vielfalt von Daten und multimedialen Dokumenten aus dem Web in produktives Wissen stellt eine der größten wissenschaftlichen Herausforderungen unserer Zeit dar! Intelligente Internet Suchmaschinen sind Softwaresysteme, welche diese Transformation durchführen können. Deshalb besteht nahezu bei allen kommerziellen und akademischen Internet Portalen ein Bedarf an einer intelligenten Suchmaschine. Jedoch sind derzeit frei verfügbare Softwaresysteme (z.B. [6]) nicht in der Lage die anfallenden Datenmengen effizient zu bewältigen. Darüber hinaus werden die relevantesten Suchergebnisse oftmals nicht an erster Stelle gelistet (Ranking). Softwarelösungen, wie z.B. der Google-Prototyp, die den modernen Anforderungen genügen, sind aufgrund der kommerziellen Nutzung im lukrativen Suchmaschinengeschäft nicht mehr frei verfügbar.

Projektziele

Der Erwerb und Einsatz kommerzieller Suchmaschinen Software ist extrem kostspielig und aufwendig. Das Ziel dieses Projekts bildet die Realisierung einer innovativen, skalierbaren Suchmaschine für das World Wide Web, die exzellente Suchergebnisse liefert und Gigabit Netzwerktechnologie effektiv nutzt. Daher sollte mit *WebSearchBench* ein Open-Source Software-Baukasten entwickelt werden, der es ermöglicht Suchdienste für Web Portale oder großer Teile des WWW zu bereit zu stellen. Die Suchmaschine besteht aus Indexer, Suchsoftware für Stichwort und Ähnlichkeitssuche, Web Crawler sowie einer Software-Komponente zum Auffinden wissenschaftlicher Interessengemeinschaften als Basissoftware Komponenten. Neben der reinen Softwareentwicklung wird für jede Komponente eine Software Qualitätssicherung durchgeführt und eine umfassende Software Dokumentation erstellt. Ein wichtiges Ziel dieses Projekts besteht darin, das Wissen in der deutschen Forschungslandschaft über die Suchmaschinentechnologie voranzutreiben. Weiterhin sollte WebSearchBench leistungsfähiger als das derzeit an deutschen Universitäten weit verbreitete Open Source Produkt ht://Dig sein, um mittelfristig WebSearchBench als Suchsoftware an deutschen Universitäten zu etablieren. Zu diesem Zweck sollte eine spezialisierte Open Source Distribution der WebSearchBench Software namens *Portal Search* angefertigt werden, die zusätzlich zum üblichen Leistungsumfang sehr einfach zu installieren und zu benutzen ist. Diese Distribution richtet sich gezielt an Rechenzentrumsleiter deutscher Universitäten. Andererseits sollte mit der Entwicklung der *Exploratory Web Search* Distribution der WebSearchBench Software ein Softwarebaukasten entwickelt werden, der die Vertiefung des Fachwissens über Aufbau und Funktionsweise einer Internet

Suchmaschine für die Lehre und Forschung sowie Akzeptanz und Nutzung bei anderen Forschergruppen gewährleistet. Neben dem Open Source Charakter der WebSearchBench, sollte dabei die effektive Verwaltung und Ausnutzung der Graphstruktur des WWW in allen Teilen der Software eine entscheidende Rolle spielen. Der innovative Schwerpunkt des Projekts liegt in der effektiven Parallelisierung der Basissoftware Komponenten der Suchmaschine für ein PC Cluster.

Die beiden Unterauftragnehmer chemie.de und bildungsserver.de sollten jede der Basissoftware Komponenten der Suchmaschine auf ihrem Web Portal erproben und Erfahrungsberichte, die Vorschläge zur Verbesserung und Optimierung der von der Universität Dortmund realisierten Software Komponenten enthalten, erstellen.

Zusammenfassend ergeben sich die folgenden Zielsetzungen für den Entwurf der WebSearchBench:

- Open-Source Software-Baukasten
- Modularer Aufbau der Software-Architektur und Möglichkeit zur einfachen Erweiterung der Software-Komponenten
- Skalierbarkeit der entwickelten Software-Komponenten mit steigender Anzahl zu verwaltender Dokumente
- Detaillierte Dokumentation aller Software-Komponenten
- Software Qualitätssicherung aller Software-Komponenten
- Effiziente Verwaltung eingebrachter Web Dokumente in einem Dokumenten Repository
- Speichereffizienter Indexer und Sortierer für HTML Dokumente
- Leistungsfähiger Web Crawler optimiert für Gigabit Technologie mit effektiver Relevanzbewertung zum zielgerichteten Crawling
- Modulare Integration neuartiger Verfahren zum zielgerichteten Crawlen von Dokumenten
- Effektive Strategie für das erneute Einbringen von Web Dokumenten
- Optimierung der Crawler Software durch Intranode und Internode Parallelisierung
- Effizientes Suchmodul für die Stichwortsuche auf Web Dokumenten
- Effektives Verfahren zur Relevanzbewertung basierend auf Vektormodellen
- Bedarfsspezifische Modifikation der Relevanzgewichte
- Aufbau und effektive Verwaltung der Graph-Struktur des WWW

- Modulare Integration neuartiger graphbasierter Verfahren zur Relevanzbewertung von Dokumenten
- Modulare Integration neuartiger Verfahren zum Auffinden ähnlicher Dokumente sowie wissenschaftlicher Interessengemeinschaften basierend auf der Graphstruktur und innovativer Klassifikationsalgorithmen
- Parallelisierung der Software-Komponenten zur Relevanzbewertung
- Parallelisierung der Software-Komponenten zur Stichwortsuche und Ähnlichkeitssuche
- Spezialisierte Verarbeitung deutsch- und englischsprachiger Dokumente
- Hinzufügen von Attributen zu indizierten Dokumenten

Zusätzlich erforderliche Arbeiten

In Diskussionen im Rahmen des Projektreffreffens am 06.08.2001 stellte sich in der Diskussion mit den Unterauftragnehmern heraus, dass zur Integration der Internet Suchmaschine in Web-Portale wie *chemie.de* und *Deutscher Bildungsserver* geeignete Schnittstellen (sogenannte *Application Programming Interfaces*, *APIs*) benötigt werden. Diese APIs sollten die einfache Administration und Konfiguration der Softwarekomponenten ermöglichen und stellen die gute Handhabbarkeit der Softwarekomponenten Crawler, Indexer, Suchsoftware und Bewertungsverfahren sicher. Daher sind die APIs für die breite Akzeptanz der WebSearchBench Software dringend erforderlich. Nach einer vorab Schätzung würde die Realisierung der Editor-, Search-, und Admin-API's von der Universität Dortmund einen Aufwand von 2, 3 und 5 Personenmonaten (PM) erfordern. Dies ergibt einen Gesamtaufwand von 10 PM wissenschaftlicher Mitarbeiter BAT IIa; d.h. derselbe Aufwand wie die im Projektvorschlag angebotene Entwicklung der Spezialsuchsoftware zum Auffinden von Communities. Daher wurde vereinbart, den Arbeitsplan des Projektvorschlag dahingehend zu modifizieren, dass diese APIs anstelle der ursprünglich geplanten Spezialsuchsoftware zum Auffinden von Communities realisiert werden soll.

Weiterhin wurde ein leistungsstarkes graphisches Front-End realisiert, das den Einsatz der entwickelten APIs praktisch demonstriert. Mit Hilfe des graphischen Front-Ends können bedarfsspezifische Modifikationen der Relevanzbewertung erprobt, indizierte Dokumente gelöscht sowie Attribute zu bereits indizierten Dokumenten hinzugefügt werden.

Ablauf des Projekts

Bereits zu Beginn des Projekts wurde die Software-Architektur mittels der Unified Modeling Language (UML) detailliert geplant. Dies ermöglichte es die Schnittstellen aller Software-Komponenten geeignet aufeinander abzustimmen und bei Bedarf einfach zu erweitern. Die

UML-Diagramme aller WebSearchBench sind dem Abschlussbericht als Anlage beigefügt. Im Rahmen des Projekts wurden insgesamt mehr als 300.000 Lines of Code in der Programmiersprache C++ entwickelt. Die ausführliche Dokumentation des Programmcodes ist daher in jeder Phase des Projekts von entscheidender Bedeutung gewesen. Für andere Forschungsgruppen bietet diese detaillierte Dokumentation, die als HTML-basierte, grafisch anschauliche Dokumentation vorliegt, die Möglichkeit bereits realisierte Verfahren einfach und schnell zu verstehen und gegebenenfalls nach eigenen Vorstellungen zu erweitern.

Während der Entwicklung und der Konzeption aller Softwaremodule wurde besonderer Wert auf die Skalierbarkeit, Robustheit und Fehlerfreiheit gelegt. Dabei soll die Skalierbarkeit den effektiven Einsatz aller Komponenten auch mit zunehmender Anzahl zu verarbeitender Dokumente und steigendem Lastaufkommen beim Crawlen und Suchen ermöglichen. Die *mehrfähdige* Ausführung aller Komponenten ist dabei entscheidend und wurde konsequent in allen Komponenten umgesetzt. Alle Komponenten ermöglichen die robuste Verarbeitung von Daten. Insbesondere beim Crawlen und Indexieren von Dokumenten kommt es sehr häufig vor, dass seitens der Web-Seiten Betreiber sowie der Web-Seiten Autoren allgemein anerkannte Protokolle und Spezifikationen (z.B. HTML, HTTP, Robots Exclusion Protokoll, etc.) nicht eingehalten und/oder modifiziert werden. Die Fehlertoleranz der Komponenten insbesondere gegenüber derartigen Gegebenheiten macht den praktischen Einsatz der Software überhaupt möglich. Gesteigerter Wert wurde außerdem auf die Fehlerfreiheit der WebSearchBench Software gelegt. Dazu wurden mittels geeigneter Werkzeuge (mpatrol, gdb, valgrind, strace, gprof, etc.) Testumgebungen geschaffen, die es ermöglichen die Evaluation der Komponenten durchzuführen und so einen hohen Grad an Fehlerfreiheit auch bei steigendem Lastaufkommen zu garantieren. Das Erstellen zahlreicher Showcases während des gesamten Projekts dokumentierte so zu jedem Zeitpunkt den momentanen Entwicklungsstand und ermöglichte den Unterauftragnehmern die jeweilige Software intensiv zu testen und geeignete Verbesserungsvorschläge zu formulieren. Weiterhin wurden im Rahmen der Projekttreffen seitens der Unterauftragnehmer Verbesserungsvorschläge für die praktische Einsetzbarkeit der Software gemacht und Schnittstellen für die Integration der Software in die Web-Portale der Unterauftragnehmer vereinbart.

Projektabschluss

Ab Mitte Juli steht die Website www.websearchbench.de mit Beispielen zur Portalsuche über ausgewählte, attraktive Websites und einem Show Case „Exploratory Web“ für die Suche im akademischen deutschen Web zur öffentlichen Nutzung bereit. Über die Website stehen auch die Dokumentationen zur entwickelten Software und die Lizenzvereinbarung zur kostenlosen Nutzung der Software durch wissenschaftliche Einrichtungen bereit. Es werden aktiv Nutzer gesucht, die die Software auf ihrem Portal einsetzen. Weiterhin wurde die WebSearchBench Software erfolgreich beim Fraunhofer Institut für Medienkommunikation (IMK) installiert. Herr Lindemann und der DFN-Verein werden im Rahmen der nächsten Betriebstagung (November 2003) einen Workshop durchführen, auf dem die Ergebnisse, insbesondere die

Vorteile der WebSearchbench Software gegenüber htdig den Betreibern von Webportalen im Wissenschaftsbereich vorgestellt werden.

Projektergebnisse

Mit WebSearchBench wurde ein Open-Source Software-Baukasten entwickelt, der es ermöglicht Suchdienste für Web Portale oder großer Teile des WWW zu bereit zu stellen. WebSearchBench ist leistungsfähiger als das derzeit an deutschen Universitäten weit verbreitete Open Source Produkt ht://Dig. Zu diesem Zweck wurde eine spezialisierte Open Source Distribution der WebSearchBench Software namens *Portal Search* angefertigt, die zusätzlich zum üblichen Leistungsumfang sehr einfach zu installieren und zu benutzen ist. Diese Distribution richtet sich gezielt an Rechenzentrumsleiter deutscher Universitäten. Für andere Forschungseinrichtungen bietet sich mit dieser Distribution die Möglichkeit einen Suchdienst auf der Web Site des jeweiligen Instituts bereitzustellen, der relevante Suchergebnisse liefert. Das Ranking ist hierbei deutlich besser als das bei anderen frei verfügbaren Suchmaschinenprodukten. Der praktische Einsatz dieser Distribution wurde mit folgenden Showcases demonstriert:

- Portalsuche auf www.uni-dortmund.de und www.imk.fhg.de
- Portalsuche auf Web Site des Unterauftragnehmers DBS

Zusammenfassend die wichtigsten Leistungsmerkmale der Portal Search Distribution:

- Softwareprodukte Web Crawler und Search Engine (Repository, Indexer, Sortierer und Suchsoftware) lauffähig auf einem Standard Linux PC.
- HTML-basierte, grafisch anschauliche Dokumentation der Softwarekomponenten
- Erzeugung und Verwaltung eines Indexes von bis zu 1 Millionen Web Dokumenten
- Verarbeitungsgeschwindigkeit des Web Crawlers/Indexers: ca. 15.000 Web Dokumente Stunde aus .de oder .com
- Eincrawlen einer vorgegebenen Menge von Websites bzw. Hosts
- Unterstützung eines Lexikons in deutscher und englischer Sprache
- Stichwortsuche in HTML und TXT Dokumenten
- Zusammengesetzte Suchanfragen mit Boolschen Operatoren (UND, ODER, Negation, Phrasensuche)
- Anzeige der Suchergebnisse mit Kontext des vorgegebenen Suchstrings (Snippets).
- Ausgabe der Suchergebnisse in alphabetischer Reihenfolge
- IR-basierte Verfahren zur Relevanzbewertung (relative Häufigkeit des Stichworts, Fett, Kursivdruck, Bestandteil der URL, etc.)

- Application Programming Interfaces zur Modifikation/Erweiterung/Wartung des Web Crawlers, des Indexes, der Suchmodul und der Relevanzbewer-tungsverfahren mit benutzerfreundlicher GUI.
- Application Programming Interface zur einfachen Integration in Web Site über eine PHP Schnittstelle.

Des weiteren wurde mit der *Exploratory Web Search* Software ein Baukasten entwickelt, der die Vertiefung des Fachwissens über Aufbau und Funktionsweise einer Internet Suchmaschine für die Lehre und Forschung sowie Akzeptanz und Nutzung bei anderen Forschergruppen gewährleistet.

- Exploratory Web Search kann dabei gezielt in Lehrveranstaltungen an Universitäten eingesetzt werden, um Studenten einen praxisnahen Einblick in die Funktionsweise einer innovativen Internetsuchmaschine zu geben. Außerdem können Studenten in speziellen Projektgruppen durch den Einsatz der WebSearchBench Software eigene Ideen und Erweiterungen im Bereich Internetsuchmaschinen direkt umsetzen und evaluieren.
- Wissenschaftlern bietet sich durch die Anwendung der WebSearchBench Software erstmalig die Möglichkeit, selbständig große und vor allem aktuelle Benchmarks erstellen zu können, und sind daher nicht mehr auf die Verwendung von Dritten bereitgestellter Daten (z.B. TREC Benchmarks, [8]) angewiesen.

Mit Hilfe der Indizes aus Dokumenten- und Linkstruktur können neuartige Verfahren aus dem Information Retrieval, wie dem Maschinellem Lernen und semantische Webanwendungen, getestet werden. Dabei gestaltete sich der Entwurf und die Realisierung der parallelen Software-Komponenten als höchst schwierig. Trotzdem konnte ein akademischer Prototyp der parallelen Software-Komponenten realisiert werden. Der praktische Einsatz wurde mit folgenden Showcases demonstriert:

- Suche im deutschen Web auf www.websearchbench.de
  (Showcase Exploratory Web Search)
- Ähnlichkeitssuche im Index der Informatik Fachbereiche deutscher Universitäten auf www.websearchbench.de

Für den Anwender im akademischen Bereich sind die folgenden Punkte interessant:

- Softwareprodukte Web Crawler und Search Engine (Indexer, Repository Sortierer und Suchsoftware) lauffähig auf einem High-end Linux PC.
- HTML-basierte, grafisch anschauliche Dokumentation der Softwarekomponenten
- Erzeugung und Verwaltung eines Indexes von bis zu 1 Millionen Web Dokumenten
- Verarbeitungsgeschwindigkeit des Web Crawlers/Indexers: ca. 15.000 Web Dokumente Stunde aus .de oder .com

- Eincrawlen einer vorgegebenen Menge von Websites bzw. Hosts
- Unterstützung eines Lexikons in deutscher und englischer Sprache
- Stichwortsuche in HTML und TXT Dokumenten
- Zusammengesetzte Suchanfragen mit Boolschen Operatoren (UND, ODER, Negation, Phrasensuche)
- Anzeige der Suchergebnisse mit Kontext des vorgegebenen Suchstrings (Snippets).
- Ausgabe der Suchergebnisse in alphabetischer Reihenfolge
- IR-basierte Verfahren zur Relevanzbewertung (relative Häufigkeit des Stichworts, Fett, Kursivdruck, Bestandteil der URL, etc.)
- Application Programming Interfaces zur Modifikation/Erweiterung/Wartung des Web Crawlers, des Indexes, der Suchmodul und der Relevanzbewertungsverfahren mit benutzerfreundlicher GUI.
- Application Programming Interface zur Integration in Web Site über PHP und C++ Schnittstellen.
- Zusätzliche Features
- graph-basierte Relevanzbewertung mit PageRank (optimierte Default-Einstellungen für Portal-Index und für Web-Index)
- graph-basierte Suche nach Web Dokumenten, die zu einem vorgegebenen Dokument ähnlich sind, mit dem Cocitation-Verfahren

PR Arbeit

Um das Projekt in den Medien und der überregionalen Presse bekannt zu machen, wurden mehrere Pressemitteillungen verfasst. Als Resonanz entstanden mehrere Artikel in Fachzeitschriften, Radio-Interviews sowie ein Auftritt von Prof. Lindemann im WDR Computerclub.

2 Aufbau und Zusammenspiel der Softwarekomponenten

WebSearchBench besteht aus den zwei Softwarebausteinen Web Crawler und Search Engine (Repository, Indexer und Suchsoftware). In Abbildung 1 sind die wesentlichen Komponenten und deren Zusammenspiel schematisch dargestellt.

Der erste Schritt zum Aufbau eines Suchdienstes besteht aus dem Sammeln von Webdokumenten, aus denen der Suchindex erstellt werden soll. Diese Aufgabe wird vom *Crawler* übernommen. Beginnend mit einer Liste von Links, der *URL Startliste*, fordert der Crawler die entsprechenden Dokumente aus dem Web an. Ob es sich hierbei um Webseiten des eigenen Portals oder um externe Seiten handelt, ist hierbei für den Ablauf nicht von

Bedeutung. Die eingebrachten Webdokumente werden im *Dokumenten Repository* gespeichert. Der *Indexer* fordert die gespeicherten Dokumente zur weiteren Verarbeitung an. Hierbei werden aus den Dokumenten neue URLs extrahiert, die anschließend vom Crawler eingebracht werden. Die von diesen URLs gebildete Verweisstruktur wird im *Index für Verweisstruktur* gespeichert. Schließlich werden alle Wörter und Metainformationen aus den Dokumenten extrahiert und im *Index für Stichworte* abgelegt. Eine effiziente Suche wird durch eine nachfolgende Umstrukturierung des Stichwortindex ermöglicht. Das *Suchmodul* nimmt über einen Web Server Suchanfragen von Benutzern entgegen. Aus diesen Sucheingaben wird eine Anfrage an den Index für Stichworte generiert, und eine Ergebnisliste von Dokumenten berechnet. In einem zweiten Schritt wird diese Liste mit Hilfe der *Relevanzbewertung* absteigend nach einem Relevanzkriterium umsortiert, als HTML-Seite aufbereitet und an den Benutzer zurück gesendet. Die Komponente zur Relevanzbewertung hat Zugriff auf die Indizes für Stichworte und der Verweisstruktur. Somit können in WebSearchBench auch leicht graphbasierte Verfahren zur Relevanzbewertung oder zum Auffinden ähnlicher Dokumente integriert werden.

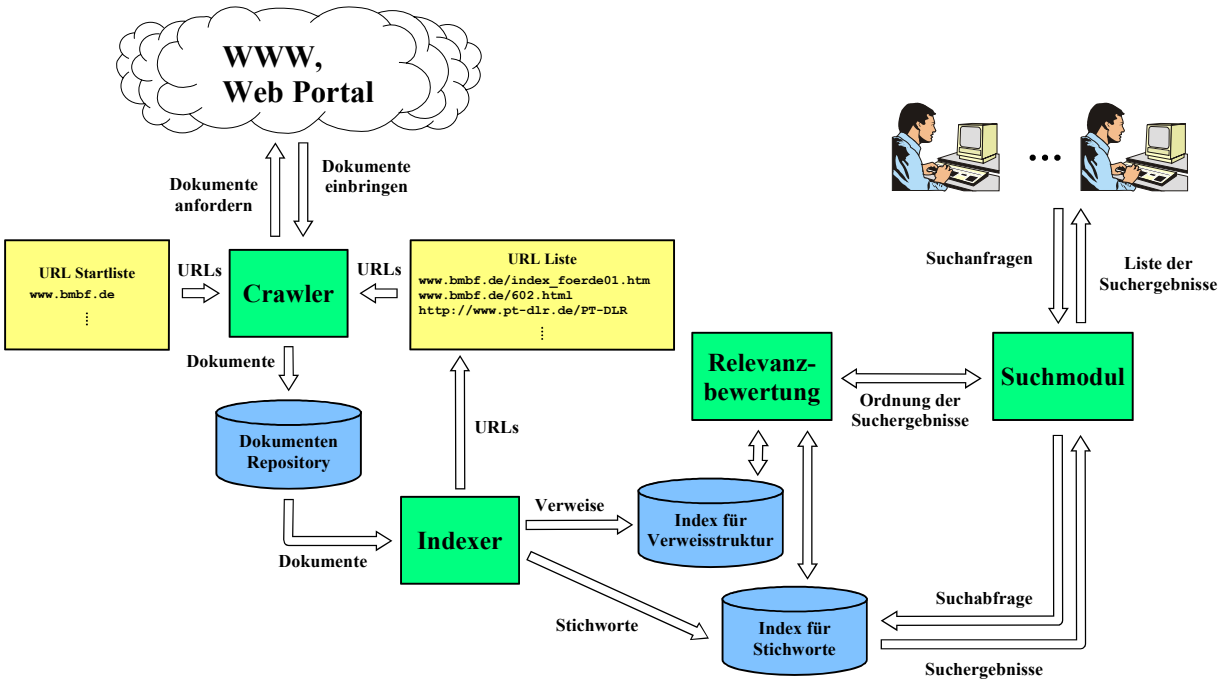

Abbildung 1. Komponenten und Arbeitsablauf von WebSearchBench

WebSearchBench bietet aufgrund seines modularen Aufbaus auch die Möglichkeit, einen externen Web Crawler, wie z.B. wget [4] zu verwenden. In diesem Fall erfolgt die Extrahierung von Links aus bereits eingebrachten Webseiten direkt durch den Crawler. Die heruntergeladenen Webdokumente werden üblicherweise in einer Verzeichnisstruktur des Dateisystems abgelegt. Durch ein separates Programm können diese Seiten eingelesen und in das Dokumenten Repository eingefügt werden. Anschließend kann der Indexer die Dokumente anfordern und auswerten, wobei die Rückkopplung der extrahierten URLs, wie in Abbildung 1 dargestellt, entfällt.

3 Leistungsumfang von WebSearchBench

Technische Eckdaten

- WebSearchBench besteht aus den zwei Softwarekomponenten Web Crawler und Search Engine (Repository, Indexer und Suchsoftware)
- lauffähig auf einem Standard Intel Pentium PC unter dem Betriebssystem Linux und zugehöriger Software wie in der Installations- und Bedienungsanleitung spezifiziert
- Web Crawler / Indexer verarbeiten mindestens 15.000 Dokumente pro Stunde
- Index kann bis zu 100.000 Dokumente enthalten
- Bei bis zu 1 Suchanfrage/Sekunde werden Antwortzeiten kleiner als 1 Sekunde garantiert
- Betrachtung von HTML und TXT Dokumenten. Bei Bedarf erweiterbar auf beliebige andere Dokumentformate, wie z.B. PDF, PS, Word oder PowerPoint

Unterstützung folgender Suchanfragen

- Einwort- und Mehrwortsuchanfragen
- UND / ODER Verknüpfung von Stichwörtern
- Negationssuche (Ausschluss von Stichworten)
- Phrasensuche zum Beispiel für die Suche nach Eigennamen wie „Ad Hoc Netze“
- Verwaltung einer individuell konfigurierbaren Stoppwortliste in deutscher, englischer und französischer Sprache
- Anzeige der Suchergebnisse im Kontext der vorgegebenen Stichwörter (Snippets)

Relevanzbewertung durch Analyse der In-Page und Off-Page Information

- Ausgabe der Suchergebnisse wahlweise in alphabetischer Reihenfolge, nach Zeitpunkt der letzten Änderung oder nach berechneter Relevanz
- Betrachtung der Metatag Informationen für die Stichwortsuche und für die Relevanzbewertung.
- Individuell einstellbare Gewichtung der einzelnen Faktoren für die Relevanzbewertung über eine Konfigurationsdatei
- IR-basierte Verfahren zur Relevanzbewertung zur Berücksichtigung der relativen Häufigkeit des Stichworts, Überschriften-/Normaltext, Bestandteile der URL, Ankertexte eingehender Verweise, Abstand der Stichwörter bei Mehrwortanfragen, etc.
- Unterstützung der graphbasierten Relevanzbewertung durch mathematische Analyse der Verweisstruktur der indizierten Web Dokumente

Application Programming Interfaces

- C++ Programmierschnittstelle erlaubt die Integration in eigene C++ Anwendungen, etwa zur Bearbeitung von Suchanfragen oder Hinzufügen von Schlagworten in den Index
- Benutzerfreundliche, graphische Anwendung zur interaktiven Indexverwaltung, d.h. Modifikation, Erweiterung und Wartung des Index, demonstriert die Leistungsfähigkeit der C++ API.
- PHP Schnittstelle zur einfachen Integration der WebSearchBench Lösung in eine bestehende IT Landschaft

4 Vergleich von WebSearchBench mit ht://Dig

Das Open Source Projekt ht://Dig [6] bietet, ebenso wie WebSearchBench, eine Komplettlösung einer Volltextsuchmaschine, die aus folgenden Komponenten besteht: Dem Web Crawler *htdig*, den Indexmodulen *htmerge* und *htfuzzy*, sowie dem Suchmodul- und Interface *htsearch*. Dieses System wurde für den Einsatz auf verhältnismäßig kleinen Untermengen des World Wide Web entwickelt, d.h. Portalseiten von Firmen, Web Sites von Universitäten etc. WebSearchBench dagegen ist insbesondere für große Datenbestände konzipiert, worin einer der wesentlichen Unterschiede von WebSearchBench und ht://Dig liegt: Der in Abschnitt 9 beschriebene Web Search Showcase wäre mit ht://Dig kaum realisierbar. Die folgende Liste gibt eine Übersicht über die wesentlichen Leistungsmerkmale in denen sich WebSearchBench von ht://Dig unterscheidet.

Zusätzliche Leistungsmerkmale des WebSearchBench Web Crawlers

- leistungsstarker, robuster Web Crawler mit 15.000 Dokumente pro Stunde geeignet zum Einbringen von mehreren Millionen Webseiten im Gegensatz zum Suchroboter htdig
- konfigurierbare Netiquette, d.h. Anzahl und Intervalle der HTTP-Anfragen pro Web Server beliebig einstellbar
- Einbringen weiterer Dokumenttypen, wie z.B. Postscript und PDF möglich
- Vollständige Archivierung aller eingebrachten Webdokumente
- Erkennen und Eliminieren von Dubletten, d.h. identischer Dokumente mit unterschiedlichen URLs

Zusätzliche Leistungsmerkmale des WebSearchBench Indexmoduls

- Erstellen enorm großer Indizes mehrerer Millionen Webseiten ohne wesentliche Einbussen der Suchgeschwindigkeit

- Hinzufügen beliebiger Schlagworte zu einem bereits indizierten Dokument im Index
- automatisches Erstellen einer Verweisstruktur erlaubt die einfache Integration mathematischer Analyseverfahren zur graphbasierten Relevanzbewertung
- Verarbeitung von deutschen Umlauten und Sonderzeichen bei der Erstellung des Indexes
- Verwaltung eines Repositories erlaubt den lokalen Zugriff auf indizierte Dokumente

Zusätzliche Leistungsmerkmale des WebSearchBench Suchmoduls

- Ranking anhand der Wortabstände in Dokumenten bei Mehrwortanfragen, wie in Abschnitt 5 detailliert beschrieben
- Phrasensuche zum Beispiel für die Suche nach Eigennamen wie „Ad Hoc Netze“
- Anzeige des archivierten Dokumentes ermöglicht Ansicht des Dokumentes, selbst wenn der Web Server, von dem dieses Dokument stammt, nicht verfügbar ist
- Aufteilung in ein Suchmodul und einem Web Interface ermöglicht den Betrieb des Suchindex und des Web Interfaces auf verschiedenen Rechnern
- Unterstützung von deutschen Umlauten und Sonderzeichen
- PHP Schnittstelle mit umfassenden Konfigurationsmöglichkeiten
- PHP API zur Bearbeitung von Suchanfragen erlaubt einfache Integration in bestehende Portale, wobei aufgrund der Trennung von Suchmodul und Web Interface der eigentliche Suchindex „Off-House“ gehalten werden kann.
- PHP Interface benötigt keine Portierung auf andere Systeme, z.B. für den Einsatz auf Windows Servern (MS IIS), im Gegensatz zum CGI Suchinterface *htsearch*, welches derzeit nur auf einigen Unix-Systemen einsetzbar ist.

Application Programming Interfaces

- APIs für die Administration von WebSearchBench, die Verwaltung des Indexes und das Suchmodul
- C++ API ermöglicht die Entwicklung eigener Anwendungen, die beispielsweise Suchanfragen an die Suchkomponente von WebSearchBench stellen können; ht://Dig bietet keine API zur Anbindung an eigene Anwendungen
- PHP API ermöglicht schnelle Entwicklung eines PHP Web Interfaces

5 Das Rankingverfahren von WebSearchBench

5.1 Optimierung des Ranking eigener Webseiten in WebSearchBench

Die Suche und das Ranking von WebSearchBench orientiert sich an [1]. Nach diesem Vorbild gibt es 2 Formen invertierter Indizes. Der erste ist ein „kleiner“ Index (*inverted file index*) für Titel und Ankertexte, der zweite Index stellt einen *full inverted Index* dar, welcher alle Hits, d.h. Vorkommen von Stichworten und deren Position, enthält. Das Lexikon beinhaltet Zeiger auf den kleinen Index. Zeiger auf den jeweiligen Eintrag im großen Index werden dagegen im kleinen Index gespeichert.
Bei einer Suchanfrage wird zunächst der kleine Index befragt. Ist die Anzahl der erzielten Treffer nicht ausreichend, wird im großen Index gesucht. Schließlich wird die daraus resultierende Ergebnisliste nach dem Ranking-Verfahren geordnet. Die Bewertung eines Dokuments errechnet sich aus den sichtbaren Inhalten und nicht-sichtbaren Inhalten (Metatags) der Dokumente.
Dieser Abschnitt beschreibt, wie das Rankings in WebSearchBench für die eigenen Portalseiten optimiert werden kann. Im Normalfall möchte man, dass ein Dokument bei bestimmten Suchanfragen möglichst weit vorne gelistet wird. Daher müssen die Begriffe der Suchanfrage in bestimmten Positionen des Webdokuments als Stichworte eingefügt werden. Es ergibt sich folgende Vorgehensweise:

1. Stellen Sie sicher, dass die Indizierung der Seite nicht zufällig durch den Robots Metatag gesperrt ist. Dazu setzen sie das index-Attribut:
   `<meta name="robots" content="`**`index,follow`**`">`
   In diesem Beispiel ist das Folgen der in dieser Seite enthaltenen Links durch einen Web Crawler erlaubt. Falls dies nicht erwünscht ist, setzen Sie den Wert auf **`nofollow`**.

2. Fügen Sie die wichtigsten Suchbegriffe in den Titel des Dokuments ein:
   `<title>`**`TITEL MIT WICHTIGEN SUCHBEGRIFFEN`**`</title>`

3. Fügen Sie die Suchworte in die Meta-Tags zur Beschreibung, Schlüsselworte, sowie den Titel nach dem Dublin Core Metatags-Standard ein:
   `<meta name="description" content="`**`BESCHREIBUNG DER SEITE ENTHÄLT WICHTIGE SUCHBEGRIFFE`**`">`
   `<meta name="keywords" content="`**`LISTE DER SUCHBEGRIFFE`**`">`
   `<meta name="DC.Title" content="`**`TITEL WIE OBEN BEI 2.`**`">`
   Ist eines der Suchbegriffe zufällig der Autor der Webseite, setzen Sie zusätzlich den entsprechenden Metatag:
   `<meta name="author" content="`**`AUTORENNAME`**`">`
   Durch Mehrfachaufzählung bestimmter Stichworte im keywords-Tag lässt sich deren Bewertung noch erhöhen. Jeder Begriff sollte jedoch nicht mehr als 2 mal aufgelistet werden.

4. Fügen Sie die gewünschten Begriffe mind. einmal in einer Überschrift (Headline) ein. Headlines der Größe 1 (`<h1>...</h1>`) werden am höchsten bewertet.
5. Fügen Sie die gewünschten Begriffe im normalen Text ein. Beachten Sie, dass deren Schriftgröße zur normalen Schriftgröße nicht verkleinert ist (`<font size="-X">`).
6. Beachten Sie, dass die Vorkommenshäufigkeit eines Begriffs bei der Bewertung berücksichtigt wird. Um ein sogenanntes Index-Spamming zu vermeiden, wird die Bewertung von Stichworten bei steigender Anzahl gedämpft. Daher ist es nicht sinnvoll die einzelnen Begriffe mehr als 30 bis 40 mal in ein großes Dokument einzufügen.

5.2 Vergleich des Rankings von WebSearchBench mit ht://Dig

Die populäre Open Source Suchmaschine ht://Dig [6] arbeitet mit einem rein inhaltsbasierten Rankingverfahren. Das Indexmodul *htmerge* von ht://Dig erkennt verschiedene Strukturelemente eines HTML-Dokuments, wie Metatags, Überschriften usw. Der Anwender hat die Möglichkeit diese Elemente beim Ranking unterschiedlich zu gewichten. Das Ranking kann über die Konfigurationsdatei *htdig.conf* konfiguriert werden (www.htdig.org/confindex.html). Nach diesem Vorgehen wird auch WebSearchBench konfiguriert: alle Werte können über die Konfigurationsdatei *websearchbench.conf* eingestellt werden. Die Bedienungsanleitung zu WebSearchBench im Anhang beschreibt die verschiedenen Konfigurationsoptionen im Detail. Dieser Abschnitt beschreibt die Parallelen und wesentlichen Unterschiede des inhaltsbasierten Rankings von ht://Dig und WebSearchBench.

WebSearchBench erkennt eine Reihe bekannter Metatags und kann diese verschieden gewichten. Es gibt Direktiven zur Gewichtung von 4 Gruppen für insgesamt 7 bekannte Metatags. Die Gruppen unterscheiden dabei zwischen Autor (`author`), Beschreibung (`description`), Stichwörter (`keywords`) und Herausgeber (`publisher`). Ht://Dig dagegen bietet lediglich 2 Direktiven für Beschreibung und Stichwörter. Allerdings lässt sich konfigurieren, welche Metatags als `description`- bzw. `keywords`-Tags erkannt werden. So kann beispielsweise der Tag `author` als zu `description` gehörig gewertet werden.

Bei den sichtbaren Textelementen unterscheidet ht://Dig zwischen Titel, die 6 Überschriftsgrößen bei HTML, sowie normalen Text. All diese Elemente können verschieden gewichtet werden. WebSearchBench bietet genau die gleiche Unterscheidungsmöglichkeit und individuelle Gewichtung dieser Elemente.

Als *Ankertext* bezeichnet man den als Link markierten Text einer Webseite. Dieser kann zusätzlich dem Dokument zugeordnet werden, auf den der Link verweist. Das ist sinnvoll, da ein solcher Text oftmals eine bessere Beschreibung über den Inhalt eines Dokuments abgibt, als der Text des Dokuments selbst. Diese Zuordnung wird sowohl von ht://Dig als auch von WebSearchBench durchgeführt, wobei eine besondere Gewichtung dieser Stichworte

durchgeführt werden kann. Ein wesentlicher Unterschied ist jedoch, dass bei WebSearchBench −im Gegensatz zu htDig− keine Neuindizierung notwendig ist, nachdem das Gewicht geändert wurde.

6 Indizierung beliebiger Dokumenttypen

WebSearchBench ermöglicht das Einbringen beliebiger Dokumenttypen, d.h. es können neben HTML- und Text-Dokumenten auch PDF- Dokumente, Postscript- Dokumente, Word-Dokumente, etc. eingebracht und im Dokumenten Repository gespeichert werden. Allerdings erlaubt die derzeitige Implementierung der WebSearchBench Software ausschließlich die Indexierung von Text und HTML-Dateien. Der modulare Aufbau der WebSearchBench Software erlaubt jedoch die einfache Integration zusätzlicher Software-Module zur Indizierung und Weiterverarbeitung beliebiger Dokumenttypen.

Prinzipiell ergeben sich zwei Möglichkeiten für die Verarbeitung beliebiger Dokumenttypen. Die erste Möglichkeit besteht darin, den reinen Textanteil eines Dokuments zu extrahieren und der Weiterverarbeitung zuzuführen. Dabei werden allerdings eventuell bestehende Strukturinformationen (z.B. Überschriften, Schriftgrößen, etc.) des zu verarbeitenden Dokuments nicht weiter berücksichtigt. Die andere Möglichkeit besteht darin, den jeweiligen Dokumenttyp auf das HTML-Format abzubilden, d.h. das betrachtete Dokument in ein HTML-Dokument zu überführen (konvertieren). Dabei lassen sich Strukturinformationen des betrachteten Dokuments in äquivalente HTML-Strukturinformationen überführen. Zur Weiterverarbeitung dient aber in beiden Fallen das Text- bzw. HTML-Dokument.

Die Integration in die bestehende WebSearchBench Software kann sowohl im Repository-Modul als auch im Indexer-Modul stattfinden:

Repository-Modul

`(TRepository::insert(), src/repository/src/TRepository.cc)`

Vor dem Speichern des Dokuments im Repository wird das Dokument wie oben erläutert in das Text- bzw. HTML-Format konvertiert und in diesem Format im Repository abgelegt. Die Weiterverarbeitung geschieht dann ausschließlich auf Basis des konvertierten Dokuments. Der Vorteil dieser Methode besteht darin, dass sich ein derart konvertiertes Dokument einfach im Web Browser anzeigen lässt, was für die Funktion *„Im Archiv“* der PHP-Suchschnittstelle vorteilhaft ist.

Indexer-Modul

`(TIndexer::ParseDocument(), src/indexer/src/TIndexer.cc)`

Vor dem Aufruf des bestehenden Text/HTML Parsers wird das Dokument in das Text- bzw. HTML-Format konvertiert. Die Weiterverarbeitung geschieht dann ausschließlich auf Basis des konvertierten Dokuments. Der Vorteil dieser Methode besteht darin, dass das Dokument

in seiner ursprünglichen Form im Repository gespeichert ist, und bei Bedarf einfach auf dieses Dokument zugegriffen werden kann. Allerdings ist das Anzeigen des Dokuments im Web-Browser über die Funktion *„Im Archiv“* der PHP Suchschnittstelle nur dann möglich, wenn ein für diesen Dateityp passendes Web-Browser Plug-in installiert ist.

Im folgenden werden bereits bestehende Software-Module (Open-Source) für die Konvertierung in das Text bzw. HTML-Format aufgeführt, die wie oben beschrieben in die WebSearchBench Software integriert werden können:

Software-Modul	Ausgangsformat	Zielformat
wp2html	Wordperfect und MS Word	HTML
catdoc	MS Word	Text
catwpd	Wordperfect	Text
rtf2html	RTF	HTML
pdftotext	PDF	Text
ps2ascii	PostScript	Text
pptHtml	Powerpoint	HTML
xlHtml	MS Excel	HTML
xls2csv	Excel	Text
swfparse	Shockwave flash files	HTML

7 Suche nach ähnlichen Web Dokumenten

Die Suche nach ähnlichen Web Dokumenten verläuft andersartig als die weitläufig bekannte Stichwortsuche und ist nur für einen aus dem Web erzeugten Index sinnvoll. Die Suchanfrage besteht in diesem Fall nicht aus der Eingabe (der Booleschen Verknüpfung) einiger Suchbegriffe, sondern vielmehr aus der Angabe einer URL zu einem bestimmten Webdokument, zu dem der Benutzer thematisch vergleichbare Dokumente finden möchte. Diese Art von Suche tritt in der jüngsten Vergangenheit öfters auf. Beispielsweise gibt die populäre Suchmaschine Google [6] zu jedem Suchtreffer ein Link *„Ähnliche Seiten“* für die Ähnlichkeitssuche aus.

In den letzten Jahren sind in einschlägigen Publikationen [3], [4], [8] graphbasierte Verfahren erforscht worden, die eine effiziente Suche nach ähnlichen Dokumenten ermöglicht. Chakrabarti greift in [2] diese Ideen auf und gibt Vorschläge zur weiteren Verbesserung. Grundvoraussetzung all dieser Verfahren ist die Fähigkeit zu einem gegebenen Webdokument effizient die Listen aller eingehenden und ausgehenden Verweise erstellen zu können. Die WebSearchBench Software stellt Funktionen zur Verfügung, die genau diese Aufgabe erfüllen. Dabei wird der von der Indexer Komponente erstellte Index der Verweisstruktur verwendet. Im folgenden wird anhand des PHP Interfaces skizziert, wie solche oder ähnliche graphbasierte Verfahren in die WebSearchBench Software integriert werden können.

7.1 Integration graphbasierter Verfahren in WebSearchBench

Erweiterung des Web Interface

(`searchApi::sendQuery()`, `html-docs/searchApi.php`)
(`TCommands::ReturnCommand`, `src/searchapi/include/TCommands.h`)

Das Web Interface muss an das Suchmodul ein entsprechendes Kommando für eine Ähnlichkeitssuche senden. Das Kommunikationsformat ist dafür entsprechend anzupassen. In der PHP API muss die Sendefunktion `SendQuery` um das neue Kommando erweitert werden. Dafür kommen Kommandowerte ab 14 in Frage. Zusätzlich muss das neue Kommando zu `ReturnCommand` in `TCommands.h` hinzugefügt werden, wobei dieses Kommando natürlich denselben Wert haben muss wie in der PHP API. Die Zählung von `ReturnCommand` beginnt mit 0. Schließlich folgt die Anpassung eines PHP Skripts nach Bedarf, welches die PHP API als Kernelement verwendet. Bei WebSearchBench sind zwei Beispielinterfaces für Deutsch und Englisch im Verzeichnis *html-docs* gegeben: *index.php.de* und *index.php.en*.

Erweiterung der Kommunikation im Suchmodul

(`TCommSearch::SetReturnCommands`, `TCommSearch::getPHPSearchString`, `src/searcher/src/TCommSearch.cc`)

Als erstes muss das beim Web Interface neu hinzugefügte Kommando in `SetReturnCommands` integriert werden. Dazu verwendet man den in `TCommands.h` neu definierten Namen. Wird nun dieses neue Kommando an das Suchmodul gesendet, wird ein entsprechendes (neu definiertes) Klassenattribut durch diese Methode gesetzt, die wiederum von der Methode `getPHPSearchString` aufgerufen wird.

In Zeile 338 in der Methode `getPHPSearchString` wird das durch `SetReturnCommands` gesetzte Attribut `_cached` abgefragt, welches anzeigt, ob das Suchmodul lediglich ein Dokument aus dem Dokumenten Repository an das Web Interface senden soll. Da das Web Interface in diesem Fall außer dem Kommando nichts weiter sendet, erfolgt diese Abfrage an dieser Stelle. Möchte man also das Suchmodul durch solche einfachen Kommandos erweitern, müssen entsprechende Abfragen hier eingefügt werden. Bei allen weiteren Kommandos sendet das Web Interface einen Suchstring an das Suchmodul. In diesem Fall muss die Erweiterung ab Zeile 422 erfolgen. An dieser Stelle wurde der Suchstring vom Web Interface gelesen und zerlegt. Bei einer normalen Suchanfrage wäre die Variable `NumTerms` größer Null und daher würde die eigentliche Suchfunktion aufgerufen. Durch vorheriges Setzen dieser Variablen auf 0 kann man dies verhindern und schließlich an dieser Stelle die Ähnlichkeitssuche integrieren. Im folgenden wird beschrieben, auf welche Weise man hierzu Anfragen an den Index Verweisstruktur stellen kann.

Erweiterung des Suchmoduls durch Anfragen an den Index für Verweisstruktur

`(read_config(), src/searcher/src/main.cc)`

`(TCommSearch::SetPath, src/searcher/src/TCommSearch.cc)`

`(TCommSearch::getPHPSearchString, src/searcher/src/TCommSearch.cc)`

Der Index für Verweisstruktur besteht aus zwei Linkstrukturdatenbanken für ein- und ausgehende Links. Zunächst muss der Speicherpfad dieser Datenbanken konfiguriert werden. Dazu wird im Hauptprogramm *main.cc*, Zeile 184 der Aufruf `TCommSearch::SetPath` als zweiter String `WsbDirRoot + '/' + TempPath + '/'` übergeben und diese Methode in *TCommSearch.cc* entsprechend angepasst.

Die Schnittstelle zu Datenbanken ist in *TCommSearch.cc* bereits durch Einbinden der Headerdatei *LinkStructure.h* bekannt gemacht worden. Nun muss nur noch eine Instanz des Linkstruktur Interface mit dem durch `SetPath` gesetzten Pfad angelegt werden: angenommen die Variable für den Speicherpfad heißt `_LinksDBPath`:

```
_LinkStructure = new TLinkStructure(_LinksDBPath.c_str());
```

Der Konstruktor öffnet automatisch die Datenbanken und der Destruktor schließt diese wieder. Nun stehen dem Konstruktoraufruf alle benötigten Funktionen zur Verfügung:

Anfragen an den Index für Verweisstruktur

`(TLinkStructure, src/databases/include/LinkStructure.h)`

Das Linkstruktur Interface benötigen die Dokumenten ID zur Identifikation eines Links. Für eine detaillierte Liste aller Methoden schauen Sie bitte in die Datei *LinkStructure.h*. Zum Beispiel gibt es u.a. folgende Methoden zum Auslesen von Informationen:

- `UINT GetNumberNodes() const` **:** Liefert die Gesamtzahl aller Knoten
- `bool Find (UINT)` **:** Prüft, ob die URL in einer der Datenbanken vorhanden ist
- `bool Find (UINT, UINT)` **:** Prüft, ob eine Kante existiert
- `UINT GetBacklinkCount (UINT)` **:** Liefert die Anzahl aller eingehenden Kanten
- `UINT GetForwardCount (UINT)` **:** Liefert die Anzahl aller ausgehenden Kanten
- `UINT* GetNeighbors (UINT, UINT, UINT &, UINT=0)` **:** Liefert Anzahl aller Nachbarn und eine Liste dieser Nachbarn
- **...**

7.2 Auffinden ähnlicher Dokumente mit dem Cocitation Verfahren

Die Integration graphbasierter Verfahren wird nun anhand des Cocitation Verfahrens [4] verdeutlicht. Das Cocitation Verfahren dient zum Auffinden ähnlicher Web Dokumente

ausgehend von der URL eines Dokuments. Beispielsweise sollten das Cocitation Verfahren bei Angabe der URL www.rtl.de ähnliche Web Dokumente aus dem Bereich Fernsehen und Medien zurückliefern, z.B. www.ard.de, www.wdr.de, usw. Im folgenden wird der zugrundeliegende Algorithmus des Cocitation Verfahrens beschrieben und dessen Fähigkeiten anhand eines Beispiels demonstriert.

Algorithmus des Cocitation Verfahrens

Die URL u bezeichnet im folgenden die URL des Dokuments, für das ähnliche Dokumente gefunden werden sollen. Es wird dazu ein *Nachbarschafts-Graph* von u gebildet, in welchem Geschwister von u gesucht werden. Betrachtet man einen gerichteten Graphen G = (V, E) so sind zwei Knoten u, v ∈ V (u ≠ v) genau dann *Geschwister*, wenn ein Knoten w ∈ V existiert, so dass (w, u) und (w, v) Kanten aus E sind. Die Konstruktion des Nachbarschafts-Graphen von u ist in Abbildung 2 dargestellt.

Eingabe: G = (V, E); u ∈ V und BF, B ∈ N
Ausgabe: S = (V_u, E_u)

$V_u \leftarrow u$
P ← Eltern von u
if ||P|| ≤ B **then**
    $V_u \leftarrow V_u \cup P$
else
    $V_u \leftarrow V_u \cup$ {B beliebige Eltern aus P}
end if
for all p ∈ P ∧ p ∈ V_u **do**
    C ← Kinder von p
    if ||C|| ≤ BF **then**
        $V_u \leftarrow V_u \cup C$
    else
        $V_u \leftarrow V_u \cup$ {BF Kinder aus C, die um u liegen}
    end if
end for
S soll der knoteninduzierte Graph bezüglich V_u sein
Bilde E_u entsprechend V_u

Abbildung 2. Konstruktion des Nachbarschafts-Graphen

Der Algorithmus garantiert, dass der Nachbarschafts-Graph nicht zu groß wird. Damit soll die Online-Fähigkeit des Verfahrens erhalten bleiben. Es werden eine gewisse Anzahl von Eltern bezüglich u betrachtet. Ausgehend von diesen werden Geschwister von u berechnet. Hierbei wird berücksichtigt, dass die meisten Web-Seiten thematisch verwandte Links bündeln. Somit

werden die Geschwister von u betrachtet, die direkt unterhalb bzw. oberhalb des Links von u bezüglich einer Elter-Seite liegen. Man verwendet also die Umgebung des Links auf u in der Elter-Seite. Auf Basis dieses Nachbarschafts-Graphen werden nur diejenigen Geschwister gesucht, die am häufigsten mit u referenziert werden. Man betrachtet dazu die Anzahl gemeinsamer Eltern von u und einem Geschwist v. Dieser Wert wird als *degree of cocitation* bezeichnet. Es existieren Fälle, in denen die Grade der Referenzierung keine hinreichenden Rückschlüsse auf die Ähnlichkeit zu lassen. Wenn beispielweise nur 10 Knoten existieren, die zweimal mit u co-zitiert werden, kann keine angemessene Aussage getroffen werden. Daher wird das Verfahren neu angewandt, wobei ein Pfadelement aus der URL u entfernt wird. Wenn sich die Ausgangs URL u aus www.beispiel.de/X/Y/Z zusammensetzt, wird die URL www.beispiel.de/X/Y als neuer Startpunkt gewählt.

Anwendungsbeispiel des Cocitation Verfahrens

Die Fähigkeiten des Cocitation Verfahren zum Auffinden ähnlicher Seiten soll nun anhand eines Beispiels demonstriert werden. Dabei wurde der in Abschnitt 9 beschriebene Showcase verwendet. Sucht man beispielsweise nach dem Begriff "Ionenantrieb" so erhält man unter den Suchergebnissen *nur einen einzigen informativer* Link

http://nathanderweise.physik.uni-giessen.de/~dhasselk/literatur.html

Über die Schaltfläche "Ähnliche Seiten" wird nun das Cocitation Verfahren aufgerufen und nach ähnlichen Dokumenten gesucht. Betrachtet man die aufgefundenen Dokumente (Abbildung 3), so stellt man fest, dass es sich um *ähnliche* Dokumente, die sich mit dem Thema (Ionen-) Antrieb beschäftigen, oder um Seiten von Forschern aus diesem Themenbereich handelt.

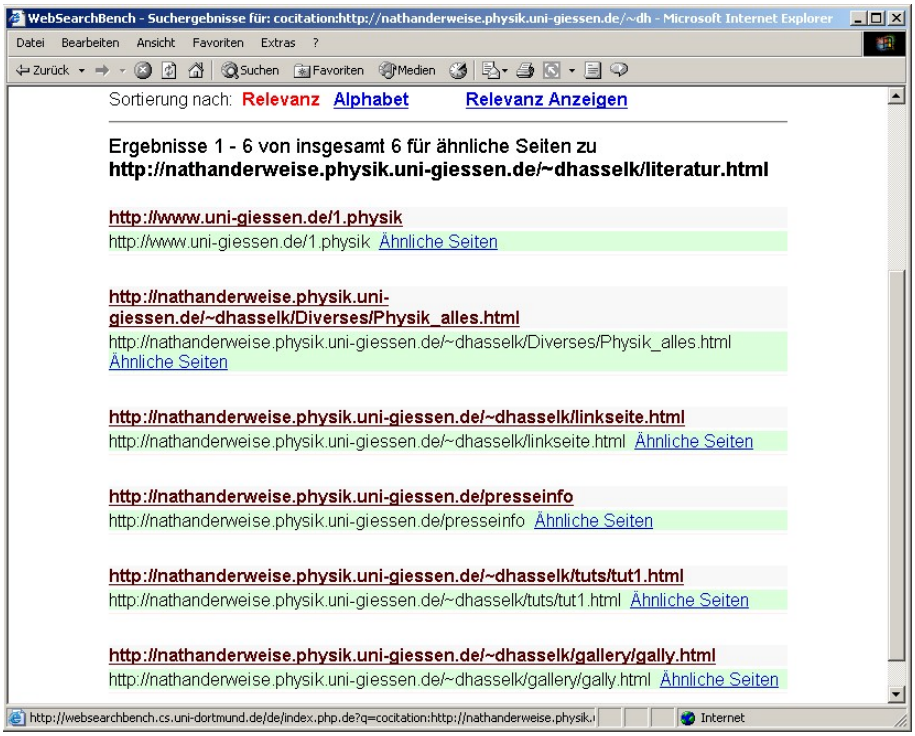

Abbildung 3. Anwendungsbeispiel zum Auffinden ähnlicher Dokumente

8 Application Programming Interfaces

WebSearchBench bietet die Möglichkeit eigene Anwendungen zu entwickeln, die Suchanfragen an das Suchmodul stellen können oder Eingriffe in den Indexer vornehmen. Hierzu wurde eine C++ Bibliothek mit einer entsprechenden API entworfen. Eine Anwendung kontaktiert mit Hilfe dieser Bibliothek einen entsprechenden Server, welcher Kommandos entgegennimmt und ausführt. Der Server für Suchanfragen ist das Suchmodul, welches auch für den normalen Betrieb der Suchmaschine eingesetzt wird. Das hat den Vorteil, dass die Anwendung auch Fragen an eine in Betrieb befindliche Suchmaschine stellen kann, ohne den normalen Betriebsablauf zu stören. Der Server für Indexkommandos wird durch die Indexer Komponente realisiert. Damit die selbsterstellte Anwendung mit der Bibliothek erfolgreich arbeiten kann, muss also der entsprechende Service gestartet worden sein. Details hierzu können der Bedienungsanleitung entnommen werden.

Für PHP gibt es ebenfalls eine API zur Bearbeitung von Suchanfragen. Diese API wird von WebSearchBench für das PHP Web Interface der Suchmaschine verwendet. Weitere Informationen zum PHP Interface finden Sie in der Bedienungsanleitung.

Leistungsmerkmale der APIs

Im folgenden werden die wichtigsten Merkmale der bereitgestellten APIs aufgeführt. Wie bereits oben erwähnt, werden die Admin-, Editor- und Search-API über statische und dynamische Bibliotheken (libAdminApi, libEditorApi, libSearchApi) bereitgestellt. Details dieser APIs sind der HTML Dokumentation der Software zu entnehmen.

AdminAPI

- Konfiguration verschiedener Features der Suchmaschinenkomponenten
- Admin Tool und Qt-GUI demonstrieren den praktischen Einsatz der AdminAPI (s.u.)

EditorAPI

- Suche nach einem Suchstring
- Festlegen des Bereichs der Rückgabeergebnisse
- Festlegen der Ausgabefelder der Suchergebnisse
- Hinzufügen von Metatags zu Dokumenten
- Löschen eines Dokuments
- Starten, Stoppen des Indexers
- Variieren der Threadanzahl des Indexers

SearchAPI

- Suche nach einem Suchstring
- Festlegen des Bereichs der Rückgabeergebnisse
- Festlegen der maximalen Anzahl der Suchergebnisse
- Festlegen der Ausgabefelder der Suchergebnisse
- Ausgabe Gesamtanzahl der Suchergebnisse
- Ausgabe von Suchergebnissen, Stop- und Ignorelisten
- Auswahl des Sortierkriteriums (alphabetisch, Datum oder Relevanz)

Anwendungsszenarien

Es folgen einige Beispielanwendungen für den Einsatz der APIs:

- Tool, welches mit Hilfe der EditorAPI den Index um Metainformationen erweitert, welche die Dokumente genauer beschreiben, etwa zur Klassifizierung der Dokumente eines Webkatalogs
- Einsatz der SearchAPI zur Erstellung eines für seinen Einsatzzweck maßgeschneidertes CGI Programm als Web Interface; z.B. kann das CGI Programm eine Lastbalancierung durchführen indem es Suchanfragen an verschiedene Suchmodule stellt
- Erstellen eines Lastgenerators für Performance Tests des Suchmoduls mit Hilfe der SearchAPI
- Mediator, der mit Hilfe der SearchAPI Suchergebnisse in andere Datenformate zwecks Datenaustausch umwandelt, etwa XML

Beispielanwendungen

Zur Demonstration der APIs gibt es zwei Anwendungen. Das Admin-Tool *admin* ist ein einfaches Programm, dass über Kommandozeile aufgerufen werden kann. Mit *admin* kann man das Indizieren von Dokumenten ferngesteuert anhalten und wieder aufnehmen, sowie den Indexer Server komplett herunterfahren. Weiterhin lassen sich durch *admin* Metainformationen zum Index hinzufügen oder ganze Dokumente entfernen. Die Anwendung von *admin* wird in der Bedienungsanleitung erklärt.

Die zweite Anwendung ist die einfache graphische Benutzeroberflache *qtsearch*, die mit Hilfe von Qt erstellt wurde. Abbildung 4 zeigt eine Beispielsitzung. *Qtsearch* wird in der Bedienungsanleitung beschrieben.

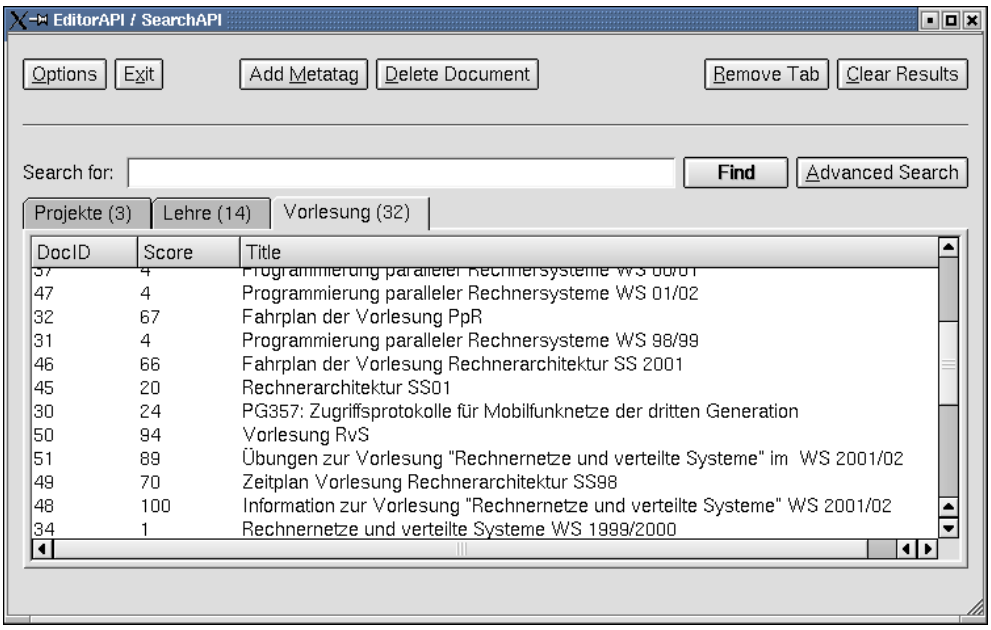

Abbildung 4. Graphische Benutzerfläche zur Search-/EditorAPI

9 Web Search Showcases

9.1 Beschreibung der Showcases

Zur Demonstration von WebSearchBench sind unter www.websearchbench.de mehrere laufende Installationen online verfügbar. Die in Abbildung 5 dargestellte Hauptseite zeigt eine Übersicht aller verfügbaren Suchdienste.

Unter *Showcases Portal Search* finden sich Installationen, die auf den Portalwebseiten deutscher Forschungs- und Förderinstitute suchen. Das Ranking erfolgt nach einer IR-basierten Bewertung unter Berücksichtigung der relativen Häufigkeit des Stichworts im Text und in Metatags, Überschriften-/Normaltext, Bestandteile der URL, Ankertexte eingehender Verweise und dem Abstand der Stichwörter bei Mehrwortanfragen. Eine Bewertung der Linkstruktur wird nicht vorgenommen, da die Verweise innerhalb eines Portals zur Bestimmung der Relevanz bedeutungslos sind. Abbildung 6 zeigt eine Beispielsuche auf dem Showcase für den DFN-Verein zum Suchbegriff *„Gerti Foest“*. Anstelle der Sortierung nach Relevanz erlaubt das PHP Interface zusätzlich die alphabetische Sortierung nach den Seitentiteln oder nach dem Erstellungsdatum der Seite. Weiterhin kann die Anzeige des Datums und der Relevanz beliebig ein- und ausgeschaltet werden. Mit einem Klick auf den Link *„Im Archiv“* kann sich der Benutzer genau die Seite aus dem Dokumenten Repository anzeigen lassen, aus welcher der Suchindex erstellt wurde. Daher ist eine Betrachtung des Suchtreffers auch dann möglich, wenn die Originalseite, beispielsweise aufgrund eines

Abbildung 5. Hauptseite der Online Showcases von WebSearchBench

temporären Ausfalls des Web Servers, nicht verfügbar ist. In allen Showcases werden pro Ergebnisseite jeweils 10 Treffer angezeigt. Über die PHP API kann diese Anzahl sehr einfach umkonfiguriert werden, sodass beispielsweise ein Auswahlmenü auf dem Web Interface eingebaut werden kann.

Der *Showcase Exploratory Web Search* unterscheidet sich von den Portal Showcases durch die Indexgröße und der Betrachtung der Linkstruktur. Diese Installation demonstriert mit einem Index von etwas mehr als 5 Millionen Webseiten, dass WebSearchBench durchaus in der Lage ist, große Suchindizes zu erstellen und zu verarbeiten. Der Index besteht aus einem Teil des Web, der eine ganze Reihe von Domains deutscher Universitäten und anderer Forschungseinrichtungen beinhaltet, mit direktem oder indirektem Bezug zum Thema *„Informatik“*, sowie einer Vielzahl von Webseiten, die auf diese Institutionen verweisen, oder auf die verwiesen wird. Somit enthält die Verweisstruktur Links, die zur Relevanzbewertung herangezogen werden können. Daher wurde im Explatory Showcase eine Relevanzbewertung der Graphstruktur integriert, der den von der WebSearchBench Software konstruierten Index für Verweisstruktur benutzt. Der so gewonnene Relevanzwert wird mit der normalen IR-basierten Bewertung derart kombiniert, dass das Ranking der Ergebnisse dem Ranking ohne Betrachtung der Verweisstruktur deutlich überlegen ist.

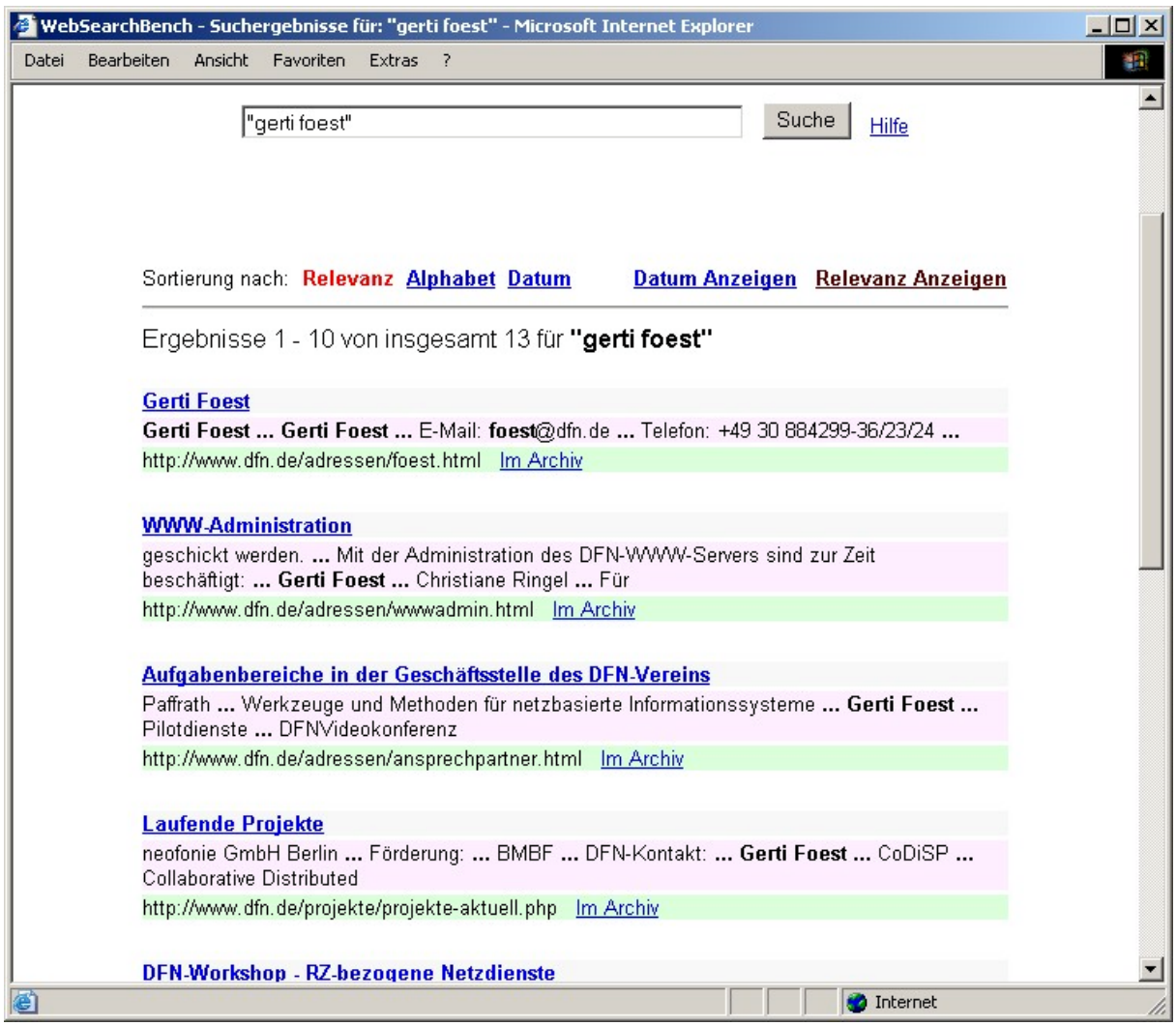

Abbildung 6. Beispielsuche auf Showcase DFN-Verein zum Suchbegriff „Gerti Foest“

9.2 URL Liste der Universitäten und Forschungsinstitute mit Informatik Bezug für Exploratory Web Search

Zur Erstellung des im letzten Abschnitt beschriebenen Showcases Explatory Web Search wurde eine Liste von URLs deutscher Universitäten und Forschungsinstitute ausgearbeitet, die Bezug zur Informatik haben. Diese Linksammlung diente als URL Startliste für den Web Crawler. Es folgt eine komplette Auflistung dieser URLs.

Allgemein

http://www.dfn.de/ (Deutsches Forschungsnetz (DFN) e. V.)
http://www.dfg.de/ (Deutsche Forschungsgemeinschaft)
http://www.bmbf.de/ (Bundesministerium für Bildung und Forschung)
http://www.bmwi.de/ (Bundesministerium für Wirtschaft und Arbeit)
http://www.pt-dlr.de/PT-DLR/ (Projektträger des DLR)
http://www.kowi.de/ (Koordinierungsstelle EG der Wissenschaftsorganisationen)
http://www.volkswagen-stiftung.de/ (Volkswagenstiftung)

http://www.ft-informatik.de/ (Fakultätentag Informatik)
http://www.gi-ev.de/ (Gesellschaft für Informatik e.V.)
http://www.dia-bonn.de/ (DIA Deutsche Informatik-Akademie)
http://www.vde.com/de/fg/itg/ (Informationstechnische Gesellschaft im VDE)

Universitäten

http://www.rwth-aachen.de/ (RWTH Aachen)
http://www.uni-augsburg.de/ (Universität Augsburg)
http://www.tu-berlin.de/ (Technische Universität Berlin)
http://www.fu-berlin.de/ (Freie Universität Berlin)
http://www.hu-berlin.de/ (Humboldt Universität Berlin)
http://www.uni-bielefeld.de/ (Universität Bielefeld)
http://www.uni-bonn.de/ (Rheinische Friedrich Wilhelms Universität Bonn)
http://www.tu-braunschweig.de/ (Technische Universität Braunschweig)
http://www.uni-bremen.de/ (Universität Bremen)
http://www.i-u.de/ (International University Bruchsal)
http://www.tu-chemnitz.de/ (Technische Universität Chemnitz-Zwickau)
http://www.tu-clausthal.de/ (Technische Universität Clausthal)
http://www.tu-cottbus.de/ (Brandenburgische Technische Universität Cottbus)
http://www.tu-darmstadt.de/ (Technische Hochschule Darmstadt)
http://www.uni-dortmund.de/ (Universität Dortmund)
http://www.tu-dresden.de/ (Technische Universität Dresden)
http://www.uni-duisburg.de/
(Gerhard-Mercator-Universität Gesamthochschule Duisburg)
http://www.ku-eichstaett.de/ (Katholische Universität Eichstätt-Ingolstadt)
http://www.uni-erlangen.de/ (Friedrich-Alexander-Universität Erlangen-Nürnberg)
http://www.uni-essen.de/ (Universität Gesamthochschule Essen)
http://www.uni-frankfurt.de/ (Johann Wolfgang-Goethe Universität Frankfurt)
http://www.tu-freiberg.de/ (Technische Universität Bergakademie Freiburg)
http://www.uni-freiburg.de/ (Albert-Ludwigs-Universität Freiburg)
http://www.uni-giessen.de/ (Justus-Liebig-Universität Gießen)
http://www.uni-greifswald.de/ (Ernst-Moritz-Arndt-Universität Greifswald)
http://www.fernuni-hagen.de/ (FernUniversität Gesamthochschule Hagen)
http://www.uni-halle.de/ (Martin-Luther-Universität Halle-Wittenberg)
http://www.uni-hamburg.de/ (Universität Hamburg)
http://www.tuhh.de/ (Technische Universität Hamburg-Harburg)
http://www.uni-hannover.de/ (Universität Hannover)
http://www.uni-hildesheim.de/ (Universität Hildesheim)
http://www.tu-ilmenau.de/ (Technische Universität Ilmenau)
http://www.uni-jena.de/ (Friedrich-Schiller-Universität Jena)
http://www.uni-kl.de/ (Universität Kaiserslautern)

http://www.uni-karlsruhe.de/Uni/ (Universität Fridericiana Karlsruhe)
http://www.uni-kassel.de/ (Universität-Gesamthochschule Kassel)
http://www.uni-kiel.de/ (Christian-Albrechts-Universität zu Kiel)
http://www.uni-koblenz-landau.de/ (Universität Koblenz-Landau)
http://www.uni-leipzig.de/ (Universität Leipzig)
http://www.mu-luebeck.de/ (Medizinische Universität zu Lübeck)
http://www.uni-magdeburg.de/ (Otto-von-Guericke-Universität Magdeburg)
http://www.uni-mainz.de/ (Johannes Gutenberg-Universität Mainz)
http://www.uni-mannheim.de/ (Universität Mannheim)
http://www.uni-marburg.de/ (Phillips Universität Marburg)
http://www.uni-muenchen.de/ (Ludwig-Maximilians-Universität München)
http://www.tu-muenchen.de/ (Technische Universität München)
http://www.unibw-muenchen.de/ (Universität der Bundeswehr München)
http://www.uni-muenster.de/ (Westfälische Wilhelms-Universität Münster)
http://www.uni-oldenburg.de/ (Carl von Ossietzky Universität Oldenburg)
http://www.uni-osnabrueck.de/ (Universität Osnabrück)
http://www.uni-paderborn.de/ (Universität-Gesamthochschule Paderborn)
http://www.uni-passau.de/ (Universität Passau)
http://www.uni-rostock.de/ (Universität Rostock)
http://www.uni-saarland.de/ (Universität des Saarlandes)
http://www.uni-siegen.de/ (Universität Gesamthochschule Siegen)
http://www.uni-stuttgart.de/ (Universität Stuttgart)
http://www.uni-trier.de/ (Universität Trier)
http://www.uni-tuebingen.de/ (Eberhard-Karls-Universität Tübingen)
http://www.uni-ulm.de/ (Universität Ulm)
http://www.uni-weimar.de/ (Hochschule für Architektur und Bauwesen Weimar)
http://www.uni-wuerzburg.de/ (Universität Würzburg)

Weitere Forschungsinstitutionen

http://www.helmholtz.de/
(Hermann von Helmholtz-Gemeinschaft Deutscher Forschungszentren)
http://www.hmi.de/ (Hahn-Meitner-Institut Berlin GmbH)
http://www.mdc-berlin.de/ (Max-Delbrück-Centrum für Molekulare Medizin)
http://www.awi-bremerhaven.de/
(Alfred-Wegener-Institut für Polar- und Meeresforschung)
http://www.ipp.mpg.de/ (Max-Planck-Institut für Plasmaphysik)
http://www.gkss.de/ (GKSS - Forschungszentrum Geesthacht GmbH)
http://www.desy.de/ (Deutsches Elektronensynchrotron)
http://www.dkfz-heidelberg.de/ (Deutsches Krebsforschungszentrum)
http://www.fz-juelich.de/ (Forschungszentrum Jülich GmbH)
http://www.fzk.de/ (Forschungszentrum Karlsruhe)

http://www.gsf.de/ (GSF - Forschungszentrum für Umwelt und Gesundheit GmbH)
http://www.fraunhofer.de/
(Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V.)
http://www.first.fhg.de/
(Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik)
http://www.fit.fhg.de/ (Fraunhofer Institut für angewandte Informationstechnik)
http://www.fokus.fgh.de/ (Fraunhofer Institut für offene Kommunikationssysteme)
http://www.isst.fhg.de/ (Fraunhofer Einrichtung für Software- und Systemtechnik)
http://www.igd.fhg.de/ (Fraunhofer Institut für Graphische Datenverarbeitung)
http://www.imk.fraunhofer.de/ (Fraunhofer Institut für Medienkommunikation)
http://www.zgdv.de/ (Zentrum für Graphische Datenverarbeitung e.V.)
http://www.iitb.fhg.de/ (Fraunhofer Institut Informations- und Datenverarbeitung)
http://www.iitb.fhg.de/ (Fraunhofer Institut für Arbeitswirtschaft und Organisation)
http://www.mpg.de/ (Max-Planck-Gesellschaft zur Förderung der Wissenschaften)
http://www.gwdg.de/
(Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen)
http://www.mpi-sb.mpg.de/ (Max-Planck Institut für Informatik Saarbrücken)
http://www.forwiss.tu-muenchen.de/
(Bayrisches Forschungszentrum für Wissensbasierte Systeme)
http://www.dfki.de/ (Deutsches Forschungszentrum für Künstliche Intelligenz)
http://www.faw.uni-ulm.de/
(Forschungsinstitut für anwendungsorientierte Wissensverarbeitung)
http://www.zib.de/ (Konrad-Zuse-Zentrum für Informationstechnik Berlin)
http://www.icd.de/ (Informatik Centrum Dortmund e.V.)
http://www.fzi.de/ (Forschungszentrum Informatik)

Referenzen

[1] BRIN, SERGEY und LAWRENCE PAGE: The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Proc 7th World Wide Web Conference*, 107-117, Elsevier Science Publishing 1997.

[2] Chakrabarti, Soumen: Mining the Web, Morgan Kaufmann, 2003.

[3] S. Chakrabarti, B.E. Dom, S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, David Gibson, and Jon Kleinberg, Mining the Web's Link Structure, *IEEE Computer* **32**, 60-67, 1999.

[4] J. Dean and M.R. Henzinger, Finding Related Pages in the World Wide Web, *Proc. 8th World Wide Web Conf.*, Toronto, Canada, 389-401, 1999.

[5] GNU wget, http://www.gnu.org/software/wget/.

[6] Google Inc., http://www.google.com/.

[7] The ht://Dig Group: ht://Dig, WWW Search Engine Software: http://www.htdig.org/.

[8] J.M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, *Journal of the ACM* **46**, 604-632, 1999.

[9] Text REtrieval Conference (TREC), http://www.trec.nist.gov/.

Kurzbericht des Projektpartners “Deutscher Bildungsserver”

1. Ziel der Beteiligung am Projekt

Praktische Erprobung der im Rahmen des Projektes WebSearchBench an der Universität Dortmund entwickelten Software. Über die Probleme, Ergebnisse und Erfolge bei der Installation der Software und ihrer Komponenten sowie bei der Nutzung des Systems auf dem Deutschen Bildungsserver sollte in 5 Etappen berichtete werden. Die Berichte dienten der schrittweisen Verbesserung der Software aus der Sicht der späteren Nutzungsmöglichkeiten.

2. Kurze Zusammenfassung der geleisteten Arbeiten mit Auswirkung auf die Softwareentwicklung an der Uni Do

a) Einrichtung eines Rechners und Integration in das Hochverfügbarkeitssystem des Deutschen Bildungsservers.

b) Beschreibung des Deutschen Bildungsservers und Spezifikation der Anforderung an die Suchmaschine aus Sicht des Deutschen Bildungsservers (26.10.01) und Übergabe einer fachspezifischen Wortliste (Thesaurusbegriffe Bildung) (25.01.02)

c) Die gelieferte Software der Universität Dortmund wurde von Seiten des Deutschen Bildungsservers installiert und nach vorgegebenen Erprobungsschemata geprüft und die Ergebnisse wurden protokolliert, siehe Testberichte vom 13.05.02, 21.10.02, 20.02.03, 08.04.03

d) Überprüfung der technischen Stabilität und der Funktionen der Komponenten Crawler, Indexer, Sorter und Searcher auf dem Deutschen Bildungsserver

e) Überprüfung der technischen Stimmigkeit und der inhaltlichen Relevanz der Suchergebnisse z.B. bei der Phrasensuche, bei der Abarbeitung der Logiken.

f) Hinweise für die Verbesserung des Suchinterfaces und der Hilfeangebote für die Benutzer

g) Entsprechend dem Projektantrag wurde auf der Grundlage der Prüfberichte in regelmäßigen Abständen der Stand der Arbeiten und die nächsten Aufgaben diskutiert und die Spezifikation der Anforderungen an die Systementwicklung fortgeschrieben.

h) Das Endprodukt der Software wurde auf dem Deutschen Bildungsserver installiert

3. Beurteilung und ggf. Einsatz der Software (oder von Teilen) im eigenen Portal

WebSearchBench ist auf dem Deutschen Bildungsserver installiert und unter der Adresse http://www.bildungsserver.de/websearch/ zugänglich.. Sobald wir die regelmäßige Erneuerung der Datenbasis durch geeignete Scripte vollständig automatisiert haben, werden wir den jetzt noch parallel zu WebSearchBench betriebenen Harvest abschalten