Indexierung von Datenbankinhalten durch Suchmaschinen Kurzfassung

Gunnar Weber
Lehrstuhl Datenbank- und Informationssysteme
Fachbereich Informatik, Universität Rostock

Zusammenfassung

In Datenbanken enthaltene Informationen machen einen Großteil des im WWW vorhandenen Wissens aus. In den meisten Fällen sind diese Informationen in dynamisch generierte WWW-Seiten eingebunden, so daß sie von den aktuellen Suchmaschinen nicht indexiert werden können. Im Projekt SWING wurden verschiedene Lösungen zur Integration von Datenbanken entwickelt. Ein bereits realisierter Ansatz nutzt lokal beim Datenbank-Anbieter vorhandene Summarizer, um Index-Fragmente, die aus den Datenbankinhalten generiert werden, an den Gatherer zu liefern. Eine weitere Möglichkeit ist die Beschreibung der Funktionalität des Anfrageformulars und des Aufbaus der Ergebnisseiten in Metadaten, die vom Gatherer zur automatischen Generierung von Anfragen und zur Extraktion der Informationen aus den Ergebnisseiten genutzt werden können.

1 Einleitung

Die Gatherer[1] der aktuell verfügbaren Suchmaschinen sammeln nur Daten ein, die im sogenannten öffentlich indexierbaren Web [6] vorhandenen sind. Informationen, die hinter Suchmasken verborgen sind, werden ignoriert. Dadurch wird bei der Recherche im Internet eine gewaltige Menge hochqualitativer Informationen aus großen, durchsuchbaren elektronischen Datenbanken nicht berücksichtigt. Solche Datenbanken haben typischerweise ein Web-Anfrageinterface, das ein HTML-Formular enthält. Der Zugriff auf die Datenbank ist dann nur über dynamisch generierte Seiten möglich, die als Antwort auf die Nutzeranfrage geliefert werden.
Es gibt verschiedene Varianten für die Integration von Datenbanken in Suchmaschinen. Diese unterscheiden sich in der Möglichkeit des Zugriffs auf die lokale Datenbank: *Kooperative Datenbankanbieter* erlauben den strukturierten Zugriff auf den Datenbankinhalt über eine spezielle Schnittstelle, bei *nicht-kooperativen Anbietern* können Datenbank-Anfragen nur über die Web-Schnittstelle gestellt werden, die mit Parametern versorgt wird. In diesem Artikel wird unter anderem ein Extraktionsmechanismus vorgestellt, bei dem vom Anbieter bereitgestellte Metadaten[2] zur Indexierung herangezogen werden. Da der Anbieter im Gegensatz zum nicht-kooperativen Fall aktiv werden muß, um die Indexierung zu ermöglichen, wird die Klasse der *eingeschränkt kooperativen Anbieter* eingeführt: Hier erfolgt der Zugriff auf die Datenbank zwar auch über das Web-Formular, das Schema der Datenbank und die Anfragefunktionalität des Formulars werden aber über Metadaten bekannt gegeben.
Für den nicht-kooperativen Fall sei auf zwei interessante Ansätze verwiesen. An der Stanford Universität wurde ein aufgabenspezifischer Web-Crawler mit der Bezeichnung *Hidden Web Exposer* [7] entwickelt, der beim Bestücken der Formulardaten auf eine endliche Menge von

[1] Gatherer sammeln Informationen über Dokumente, die im Internet verfügbar sind.
[2] Metadaten dienen zur Definition bestimmter Eigenschaften eines HTML-Dokuments, die vom WWW-Browser bei der Darstellung nicht anzeigt werden.

Konzepten bzw. Kategorien zugreift. Die Zuordnung eines Formularelements zu einem Konzept wird über Ähnlichkeiten zwischen der textuellen Beschreibung des Elements und der Konzeptbeschreibung vorgenommen. In [2] wird das Ausnutzen von kombinierten Analysetechniken des Text Mining vorgeschlagen, um Formularelemente zu Konzepten einer Ontologie zuzuordnen.
Im nächten Abschnitt werden die Probleme diskutiert, die bei der Indexierung von Web-Datenbanken auftreten. Anschließend wird in Abschnitt 3 ein in der Suchmaschine SWING realisierter Ansatz vorgestellt [5], der die Einbindung von Datenbanken kooperativer Anbieter erlaubt. Abschnitt 4 beschäftigt sich dann mit der Integration von Inhalten teilweise kooperativer Anbieter. Details zu den einzelnen Verfahren können in der Langfassung des Artikels [9] nachgelesen werden.

2 Probleme bei der Indexierung

Bei der Indexierung von Inhalten, die hinter Anfrageformularen in Datenbanken "versteckt" sind (oft auch als *Hidden Web* bezeichnet), treten drei grundlegende Probleme auf.
Das erste ist das Problem der Skalierung. Nach einer Studie [1] ist das Hidden Web bis zu 500 mal größer als das statische Web. Aus diesem Grund ist eine umfassende Indexierung unmöglich und häufig auch nicht sinnvoll. Die aktuell verfügbaren Suchmaschinen berücksichtigen keine Zusammenhänge zwischen den Informationen eines Dokuments[3]. Somit reicht es aus, nur unterschiedliche Attributwerte und keine kompletten Tupel in den Index aufzunehmen. Dies führt zu einer deutlichen Reduzierung der Größe des Indexfragmentes für eine Datenbank.
Das zweite Problem liegt in der automatischen Erfassung der Funktionalität des Anfrageformulars, das für den menschlichen Nutzer konzipiert ist. In diesem Artikel werden deshalb unter anderem Metadaten vorgeschlagen, mit denen Anbieter diese Funktionalität in maschinenverständlicher Art und Weise beschreiben können.
Die Aufgabe des Anfrageformulars besteht in der Filterung der für einen Nutzer relevanten Datenbankinhalte. Diese Filterung führt zum dritten Problem, da ein Gatherer alle wichtigen Inhalte einsammeln soll. Somit muß der Gatherer beim Zugang über eine Web-Schnittstelle mehrere Anfragen nacheinander an die Datenbank stellen. Mit zunehmender Komplexität des Anfrageformulars und zunehmender Anzahl möglicher Werte für die Elemente steigt der Zeit- und Ressourcenaufwand für das Abfragen der Datenquelle sehr stark an.
Für den kooperativen Fall spielen die letzten beiden Probleme keine Rolle, da direkt auf die Datenbank zugegriffen wird.

3 Einbindung von Datenbanken in SWING

Basisbestandteil der SWING-Suchmaschine ist das *Harvest*-System [4]. Zur Einbindung von Datenbanken kooperativer Anbieter wurde die Gatherer-Komponente des Systems modifiziert. Die Grundidee des Ansatzes besteht in der Verwendung lokaler Summarizer, die leicht konfigurierbar sind und vorhandene Standards für den Datenbankzugriff nutzen (DBPerl, JDBC).
Damit der Gatherer erkennen kann, daß es sich um eine einzubindende Datenbank handelt, wurde der neue Dokumenttyp `DBContent` eingeführt. Solch ein Dokument muß auf dem WWW-Server, der den Zugriff auf die einzubindende Datenbank ermöglicht, vorhanden sein und enthält folgende Informationen: die Adresse des lokalen Summarizers, die URL der Web-Anfrageschnitt-

[3] außer der Entfernung zwischen Wörtern

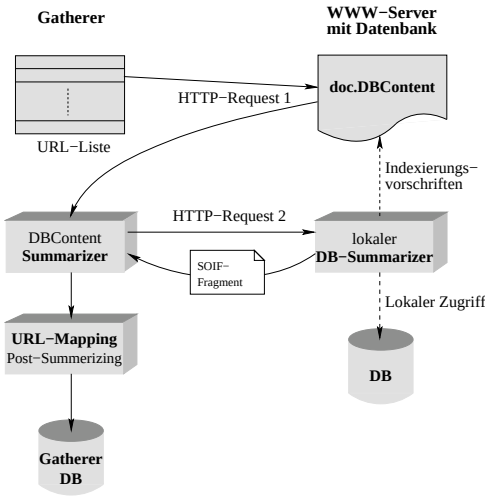

Abbildung 1: Kooperativer Ansatz

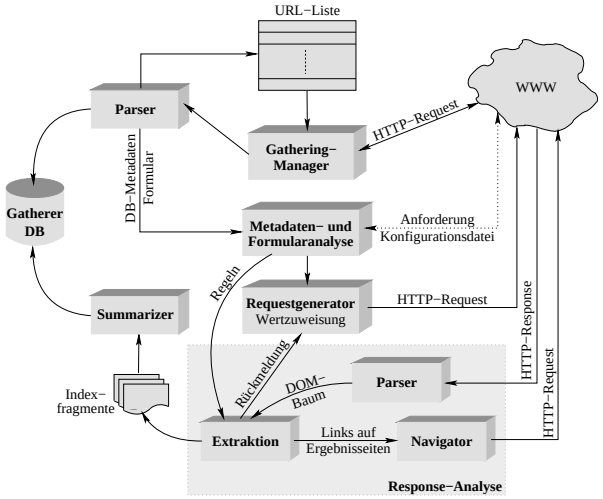

Abbildung 2: Eingeschränkt kooperativer Fall

stelle, Metadaten wie *Description* und *Keywords* sowie Angaben darüber, welche Attribute aus welcher Relation indexiert werden sollen.
In der Abbildung 1 sind die Schritte graphisch dargestellt, die vom Gatherer durchgeführt werden, um Datenbankinhalte in das Retrievalsystem der SWING-Suchmaschine einzuspeisen. Die URLs für die Konfigurationsdateien müssen dem Gatherer explizit bekannt gemacht werden. Stößt der Gatherer bei der Abarbeitung der *URL-Liste* auf solch eine Datei, wird der *DBContent-Summarizer* aufgerufen. Dieser parst die Konfigurationsdatei und ruft dann die URL auf, die für den *lokalen Summarizer* angegeben wurde. Als Parameter wird der Pfad der Konfigurationsdatei relativ zum lokalen Summarizer übergeben. Über diesen Pfad kann der lokale Summarizer auf die Datei zugreifen, um die datenbankspezifischen Informationen zu lesen (Name der Datenbank, zu indexierende Tabellen und Attribute). Anschließend werden SQL-Anfragen zum Ermitteln der Werte für die zu indexierenden Attribute an die Datenbank gestellt. Im letzten Schritt faßt der lokale Summarizer alle Ergebnisse in einem *SOIF-Fragment*[4] zusammen und schickt es an den Gatherer zurück. Dort wird es dann mit der URL für das Anfrageformular und weiteren Metainformationen (Größe des Fragmentes usw.) in der *Gatherer-Datenbank* gespeichert.

4 Datenbankeinbindung eingeschränkt kooperativer Anbieter

In vielen Fällen haben Anbieter von Web-Datenbanken Bedenken, fremde Programme auf ihren Rechnern zu installieren, die Inhalte nach außen geben. Deshalb soll im folgenden ein Mechanismus vorgestellt werden, der die normale Web-Schnittstelle für die Indexierung nutzt. Dieser Ansatz geht davon aus, daß der Anbieter mit Hilfe von Metadaten beschreibt, welche Funktionalität das Anfrageformular bietet und wie die Ergebnisseiten aufgebaut sind.
Einige Formularelemente haben *finite Domänen*, d.h. die gültigen Werte sind in der HTML-Seite eingebettet (Auswahllisten, Radiobuttons, Checkboxen). Andere Elemente wie Textfelder haben *infinite Domänen*, d.h. sie können beliebige Werte annehmen. Der Gatherer benötigt für die Generierung von Anfragen aber feste Werte. Die Verwendung von Ontologien zur Bestimmung möglicher Werte ist hier nicht möglich, da der Gatherer nicht auf eine Anwendungsdomäne eingeschränkt werden soll. Deshalb wird in diesem Ansatz versucht, infinite Domänen mit

[4]Das *Summary Object Interchange Format* (SOIF) besteht aus einer Liste von Attribut/Wert-Paaren.

regulären Ausdrücken[5] zu erfassen. Damit lassen sich zumindest solche Domänen beschreiben, die ein bestimmtes Format voraussetzen oder die Angabe einfacher Teilmuster zulassen.
Für die Beantwortung von Anfragen sind einige Elemente wesentlich, d.h. die Angabe eines Wertes ist unbedingt erforderlich. Es gibt aber auch Elemente, die nicht spezifiziert werden müssen und nur eine Reduzierung der Ergebnismenge bewirken. Damit der Gatherer die Datenbankinhalte mit einer minimalen Anzahl von Anfragen bestimmen kann, muß der Anbieter die wesentlichen Anfrageelemente in den Metadaten definieren.
Die Abbildung 2 zeigt die grundlegende Arbeitsweise des Gatherers. Der *Gathering-Manager* steuert den gesamten Prozeß der Datensammlung. Er entscheidet, welcher Link als nächstes bearbeitet werden soll und fordert dann das zugehörige Dokument an. Dieses Dokument wird an den *Parser* weitergegeben, der inhaltliche Informationen ermittelt und diese zusammen mit Informationen über das Dokument selbst (URL, Größe, usw.) in strukturierter Form (z.B. SOIF) in der *Gatherer-Datenbank* ablegt. Die extrahierten Links werden in die *URL-Liste* eingefügt und stehen somit dem Gathering-Manager zur Verfügung. Enthält das Dokument datenbankspezifische Metadaten, handelt es sich um ein Anfrageformular für eine Web-Datenbank. Der Gatherer unternimmt folgende Schritte, um die Inhalte der Datenbank zu bestimmen:

- *Metadaten- und Formularanalyse*: Die vom Parser übergebenen DB-Metadaten und das Anfrageformular werden zunächst zerlegt und analysiert. Wird in den Metadaten auf eine Konfigurationsdatei verwiesen, dann muß diese über einen zusätzlichen HTTP-Request angefordert werden. Für das Formular sind folgende Information zu bestimmen: die URL des aufzurufenden Skripts und die Methode für die Parameterübergabe, alle Elemente, deren Typen und interne Bezeichner sowie alle finiten Domänen.

- *Request-Generation*: Bei der Erzeugung der Anfragen werden nicht alle Formularelemente berücksichtigt, sondern nur diejenigen, die vom Datenbankanbieter als wesentlich spezifiziert wurden. Ist nur das Element E_i mit der Domäne D_i wesentlich, dann werden $|D_i|$ Anfragen erzeugt. Bei $n > 1$ Elementen wird das kartesische Produkt der Domänen $D_1 \times D_2 \times ... \times D_n$ gebildet, so daß insgesamt $|D_1| * |D_2| * ... * |D_n|$ Anfragen an die Datenbank gestellt werden müssen. Bei infiniten Domänen, die mit Hilfe regulärer Ausdrücke definiert wurden, ist zusätzlich noch eine Erzeugung der möglichen Werte notwendig.

- *Response-Analyse*: Die Bestimmung der Inhalte auf der Ergebnisseite erfolgt mit Hilfe der in den DB-Metadaten enthaltenen Extraktionsregeln. Zur Definition dieser Regeln wird die deklarative Sprache *HEL* (***H**TML **E**xtraction **L**anguage*) verwendet, die für das Wrapper-Toolkit W4F [8, 3] entwickelt wurde. Die Ergebnisseite dient als Eingabe für einen *HTML-Parser*, der daraus einen Analysebaum erzeugt. Innerhalb dieses Baums kann jedes Element im Dokument anhand von Pfadausdrücken eindeutig identifiziert und dessen Inhalt bestimmt werden. Für jede Antwortseite entsteht so ein Indexfragment, das die extrahierten Inhalte zusammenfaßt. Ist das Ergebnis über mehrere HTML-Seiten verteilt, müssen diese ebenfalls angefordert werden. Dafür ist der *Navigator* zuständig, der alle Links auf Ergebnisseiten sammelt und nacheinander abarbeitet.

- *Zusammenfassung*: Der *Summarizer* faßt die Indexfragmente, die pro Seite erzeugt wurden, zusammen. Doppelte Werte für ein Attribut werden dabei eleminiert.

Die extrahierten Datenbankinhalte werden zum Schluß gemeinsam mit dem Anfrageformular in der *Gatherer-Datenbank* abgelegt.

[5] Es sind nur reguläre Ausdrücke gestattet, die eine endliche Menge von Werten beschrieben.

5 Zusammenfassung und Ausblick

In diesem Artikel wurden zwei Ansätze zur Indexierung von Datenbankinhalten durch Suchmaschinen vorgestellt, die von einer unterschiedlichen Kooperationsbereitschaft des Datenbankanbieters ausgehen. Der erste in der Suchmaschine SWING realisierte Ansatz setzt die volle Kooperationsbereitschaft des Anbieters voraus, da ein lokaler Summerizer installiert werden muß. Dieser Mechanismus ist sehr effizient, da nur zwei HTTP-Requests für die Indexierung notwendig sind. Viele Anbieter scheuen sich aber aus unterschiedlichen Gründen davor, ein solches Programm auf ihrem Server zu installieren. Aus diesem Grund wurde ein zweiter Ansatz vorgeschlagen, der die Indexierung der Datenbankinhalte über die normale Anfrageschnittstelle erlaubt. Hier beschreibt der Anbieter über Metadaten, welche Funktionalität diese Schnittstelle bietet und wie die Ergebnisseiten aufgebaut sind. Der Gatherer nutzt die Metadaten für die Generierung von Anfragen und die Extraktion der Attribute, die indexiert werden sollen. Zwei Hauptprobleme treten bei diesem Mechanismus auf: zum einen die Beschreibung infiniter Domänen und zum anderen die Vielzahl der Anfragen, die an die Datenbank gestellt werden müssen. Zur Definition infiniter Domänen wird momentan auf reguläre Ausdrücke zurückgegriffen. Somit lassen sich nur Domänen beschreiben, die ein bestimmtes Format voraussetzen oder bei denen in der Anfrage einfache Teilmuster verwendet werden können. Weitere ungelöste Probleme sind Interaktionen im Anfrageprozeß bzw. abgeschnittene Ergebnislisten bei zu vielen Treffern.

Als nächster Schritt ist die prototypische Implementierung des Gatherers vorgesehen, um Aussagen über die Leistungsfähigkeit des Ansatzes geben zu können. Dabei ist vor allem interessant, wie sich das Zeitverhalten bei steigendem Datenvolumen und bei steigender Komplexität der Anfrageschnittstelle entwickelt. Dann kann auch die Frage geklärt werden, ob die Einbindung von Datenbanken überhaupt Sinn macht, wenn Formularelemente mit infiniten Domänen beim Abfragen der Datenbankinhalte berücksichtigt werden müssen.

Literatur

[1] M. K. Bergmann et al. The Deep Web: Surfacing Hidden Value. Whitepaper, BrightPlanet.com LCC, July 2000. Available at `http://www.completeplanet.com/tutorials/deepweb/`.

[2] I. Bruder. Zugriff auf dynamische Web-Dokumente mittels Web-Mining-Analysetechniken. Preprint CS-04-01, Fachbereich Informatik, Universität Rostock, Mar. 2001.

[3] R. Gohla. Integrierte WWW-Anfragesichten. Master's thesis, Fachbereich Informatik, Universität Rostock, 2000.

[4] D. R. Hardy, M. F. Schwartz, and D. Wessels. Harvest User's Manual. Technical Report CU-CS-743-94, University of Colorado, Boulder, Jan. 1996. Available at `http://www.tardis.ed.ac.uk/harvest/docs/`.

[5] A. Heuer and G. Weber. SWING: Eine Suchmaschine mit Datenbankanschluß. In *Workshop Internet-Datenbanken*, number 12 (Preprint), Magdeburg, Sept. 2000. Fakultät für Informatik, Otto-von-Guericke Universität.

[6] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[7] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. Technical report, Stanford University, Nov. 2000.

[8] A. Sahuguet and F. Azavant. WysiWyg Web Wrapper Factory (W4F). Technical report, University of Pennsylvania und Telecom Paris (E.N.S.T.), 1998. Available at `http://db.cis.upenn.edu/W4F/`.

[9] G. Weber. Indexierung von Datenbankinhalten durch Suchmaschinen. Preprint CS-03-01, Fachbereich Informatik, Universität Rostock, Mar. 2001.