Suche in semi-strukturierten Dokumenten

Seminar Multimediale Informationssysteme

Christoph Thelen

Gliederung

- Volltextsuche
- Strukturieren bei Dokumenten
- Semi-strukturierte Daten
- Anfragesprachen
  - XPath
  - XQuery
  - Erweiterungen

Was ist ein Dokument?

- Zusammengehörig Daten und Informationen
- Keine fest definierte Form
- Festgelegte Reihenfolge (z.B. Buch)
- Implizite Verknüpfungen
  - Aufteilung in Abschnitte mit Überschrift
  - „Er heißt Meier und wohnt in der Marktstrasse 33“

Volltextsuche

- Suche nach Textfragmenten
- Einordnung und Verknüpfung mit anderen Daten maschinell sehr schwer
- Bedeutung oft nur aus Kontext erschließbar
- Aufgabe: Bedeutung der Daten analysieren
- 2 Arten der Suche:
  - Boolesche Suche
  - Unscharfe Suche (Fuzzy Retrieval)

Boolesches Retrieval

- Ergebnis: Menge von Dokumenten, die eine Anfrage erfüllen
- Logische Verknüpfungen
  - Dokument wird abgelehnt, wenn nur ein Teil einer UND-Verknüpfung nicht erfüllt ist
  - Exakte Anfrage nicht immer möglich – evtl. Ablehnen von relevanten Dokumenten
- Größe der Ergebnismenge vorher unbekannt
- Unnatürliche Frageformulierung

Unscharfe Suche

- Ranked Retrieval
  - Ergebnis nach der Relevanz geordnet
  - Größe der Ergebnismenge vorher bestimmbar
    - „Alle Dokumente mit Relevanz > 0,5“
    - „Die 10 besten Dokumente“

Volltextsuche

⇒Hilfestellung für den Menschen

⇒Nicht zur automatischen Informationsintegration

Strukturieren bei Dokumenten

- Menschliche Sicht - Ziele
  - Zweck des Dokumentes schnell erkennen
  - Informationen schnell finden
- Rechnersicht
  - Sammlung von Daten
  - Beispiel: Tabelle in DBS
    - Schema genau definiert ($\rightarrow$ festgelegte Semantik der Daten)
    - Schneller Zugriff

Semi-strukturierte Daten

- Datenaustausch
- Strukturtragende Daten
  - Daten
  - Struktur
- Schemainformationen
- Flexibler als Relationenmodell

Dokumenten- ↔ DB-Perspektive

- Reihenfolge wichtig (z.B Buch)
- Verknüpfungen durch Struktur
- Schema evtl. in separater Datei
- Ausnahmen/ Missachtung des Schemas möglich

- Daten im Vordergrund
- Reihenfolge unwichtig
- Festes Schema
  - Semantik klar
  - Explizite Verknüpfungen
  - Keine Ausnahmen
- Boolesches Retrieval
- Mächtige Anfragesprachen

eXtended Markup Language (XML)

- Untermenge von SGML (W3C)
- Erweiterbarkeit
- Tags
  - Paarweise
  - Korrekt verschachtelt
  - Attribute
- Struktur durch Verschachtelung der Tags
- Ein äußeres Tag-Paar $\rightarrow$ Wurzelelement

XML (2)

- Wohlgeformtes Dokument:
  - Befolgt die XML-Syntax
  - Korrekte Schachtelung
- Gültiges Dokument
  - Wohlgeformt
  - Folgt den Regeln einer DTD

DokumentType Definition (DTD)

- Grammatik/Schema von XML-Dokumenten
  - Erlaubte Strukturen
    - Art der Verschachtelung
    - Optionale Elemente

```
<!ELEMENT personen                (person*)>

<!ELEMENT person  (name, vorname?,  wohnort?)>
 <!ELEMENT name                     (#PCDATA)>
 <!ELEMENT vorname                  (#PCDATA)>
 <!ELEMENT wohnort                  (ort)>
```

Anfragesprachen

- Anforderungen:
  - Daten aus großen Dokumenten extrahieren
  - Flexible Anfragen
  - XML-Ausgabe ($\rightarrow$ Relationale Abgeschlossenheit)
  - Kein Schema benötigt
  - Vorhandenes Schema ausnutzen
  - Gegenseitige Einbindung mit XML
  - Reihenfolge und Verknüpfungen erhalten

XPath

- Grundlage der meisten „höheren“ Anfragesprachen für XML
- Ansprechen von Teilen eines XML-Dokuments
- Baum entsprechend der Struktur
- Knoten
  - Elemente
  - Attribute
  - Text

XPath – Knoten

- Knotentypen
  - Wurzelknoten
  - Elementknoten
  - Textknoten
  - Attributknoten
  - Namespace Knoten
  - Processing Instruction Knoten
  - Kommentarknoten

XPath - Ausdruck

- Wichtigstes Konstrukt in XPath
- Mögliche Rückgaben:
  - Knotenmenge
  - Boolescher Wert
  - Gleitkommazahl
  - String
- Auswertung erfolgt innerhalb eines bestimmten Kontextes

XPath – Location Path

Syntax

- `child::para` das Kind `para` des Kontextknotens
- `child::*` alle Kinder des Kontextknotens
- `self::para` Kontextknoten, wenn vom Typ `para`
- `/descendant::para` alle `para` Elemente im selben Dokument wie der Kontextknoten
- `child::para[position()=last()-1]` vorletztes `para` Element, das Kind des Kontextknotens ist
- `child::para[attribute::type="warning"]`

- Verkürzte Schreibweise

XPointer

- Neuer Ergebnistyp: „Location Set“
  - „vom dritten Wort im ersten bis zum fünften Wort im letzten Abschnitt“
- „Point“ („drittes Wort im ersten Abschnitt“)
  - Knoten und Index
- „Range“
  - Start- und Zielpunkt (Point)
  - Alle dazwischenliegenden Elemente oder Teile von Elementen

XPointer – Funktionen

- `location-set range-to (location-set)`
  - `//para[2]/range-to(following::chapter)` alles zwischen dem zweiten `para` Element und dem nächsten `chapter`
- `location-set string-range (location-set, string, number, number)`
  - `string-range(//para, "Ziel", 2, 0)` alle Positionen von „Ziel“ im String-Value aller `para` Elemente im Dokument (als Points)
  - Mixed-Content

XQuery

- Vorläufer: Quilt
- Basiert auf XPath
  - Knotentypen
  - Datenmodell
- FLWR-Ausdrücke
  - FOR $v IN pfad
  - LET $v IN pfad
  - WHERE bedingung
  - RETURN <element>$v</element>

XQuery - Blumenkind

```
<kunden>
 FOR $k IN dokument ("marktplatz.xml")//käufer
 LET $b =count ($k//bestellung[datum>2001-01-01])
 WHERE $b > 0
 RETURN
  <kunde>
    $k/name,
    $k/vorname,
    <anzahl>$b</anzahl>
  </kunde>
</kunden>
```

XQuery (3)

Rückgabe: Anzahl Bestellungen seit 1.1.2001

```
<kunden>
  <kunde>
    <name>Meier</name>
    <vorname>Klaus</vorname>
    <anzahl>1</anzahl>
  </kunde>
  <kunde>
    <name>Baumann</name>
    <vorname>Peter</vorname>
    <anzahl>3</anzahl>
  </kunde>
</kunden>
```

XQuery (4)

- Relational abgeschlossen
  - Anfrage liefert vollständiges Dokument zurück
  - Jeder Ausdruck der Relationenalgebra kann simuliert werden
    - Mengenoperationen, Joins, ...
- Dokumentspezifische Operatoren
  - BEFORE, AFTER
  - Reihenfolge im Dokument
- Sortierung mit SORT BY

XQuery (5)

- Arithmetisch Operatoren
  - Auf Werten, Knoten und Werten und Listen
- Vergleichsoperatoren
- ...

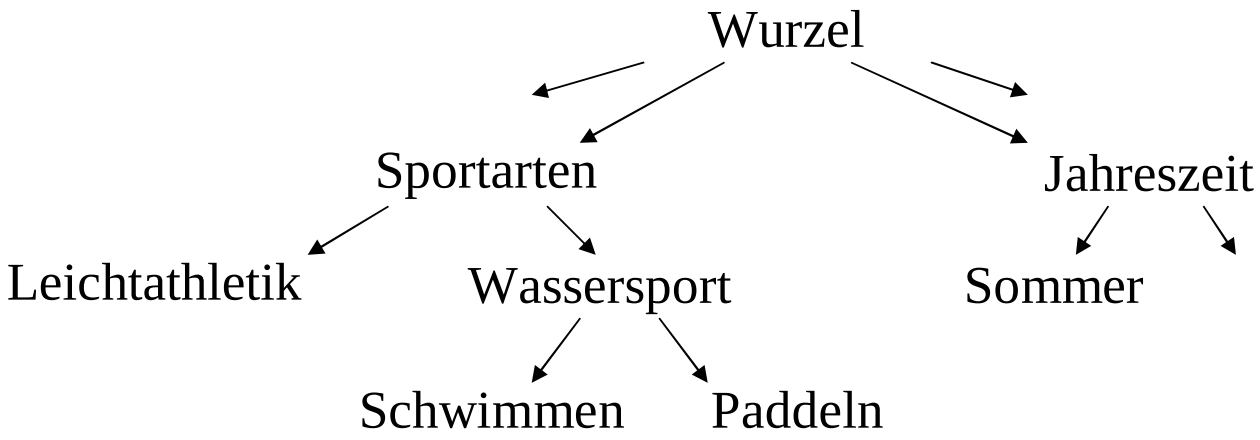
- Reines Boolesches Retrieval
- Keine Bewertung der Ergebnisse
- $\rightarrow$ Erweiterungen

XIRQL

- Erweitert XQL um
  - Wahrscheinlichkeitsgesteuerte Suche (gewichtet)
  - Relevanz-orientierte Suche
  - Erweiterbare Datentypen mit unscharfen Attributen
- Suche über Teile des Dokuments
  - Gewichtung der Zwischenergebnisse über ihren Kontext
  - Kein definierter Pfad nötig
  - Ähnlichkeitsvergleich über die Struktur

XXL

- Vergleich über Textähnlichkeit
- Benutzt eine Onthologie (Wissensdatenbank)
  - Informationen über die Miniwelt
  - Verknüpfung von Begriffen



Ausblick

- Datenaustausch zwischen verschiedenen Datenbanken
  - XML-DBMS
    - Freie Java-Bibliothek
    - Bidirektionale Abbildung zwischen XML-Dokumenten und relationalen Datenbanken
- XML in Middleware
  - Produktunabhängigkeit