The Team

Prof. Dr. Felix Naumann

Alexander Albrecht    Mohammed AbuJarour    Jens Bleiholder
Introduction

What is Information Retrieval?

- Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).
  - Also semi-structured data.
  - Web pages.

Contents of the Seminar

- Models for document & query representation.
- Indexing.
- XML Information Retrieval.
- Information Filtering / Recommendation.
- Search and Web Information Retrieval.
- Classification.
- Clustering.
Seminar’s web page: http://www.hpi.uni-potsdam.de/naumann/lehre/ws_0809/ir.html

ECTS credit points: 3.

Time: Tuesday 15:15 – 16:45.

Location: HPI A-2.1.

Registration: email with your favorite 3-topics to (Mohammed AbuJarour) before 23.10.2008.

Prerequisites:

- Database.
- Data Structures and Algorithms.
- Linear Algebra.
- Probability Theory.

A session on “Foundations of Information Retrieval” next week.

Papers will be available / accessed online.
Requirements to pass the seminar:

- Attendance:
  - Show up in all sessions.
  - If you cannot attend for some reason, let us know per email beforehand.

- Give a talk in English:
  - 30-45 minutes: talk.
  - 15 minutes: discussion and comments.

- Participation:
  - In all talks.
  - Discussion and challenging questions.

- Report
  - The report should discuss (not summarize) the assigned work/material.
  - Show strengths, weaknesses, suggestions and comments ...
  - Due in 2-3 weeks from the date of the talk.
Topics: Indexing
IO-Top-k: index-access optimized top-k query processing

- Ranked Retrieval & Top-k query processing.
- Index Lists & aggregate scores.
- Based on the family Threshold-Algorithms.
- Disk accesses:
  - Sorted Accesses & Random Accesses.
  - Scheduling problem!

- What’s new?
  - The integration view of both kinds of accesses.

Example query: //A ////a ////B ////b ////C ////c

Pre-computed index table:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Term</th>
<th>MaxScore</th>
<th>DocId</th>
<th>Score</th>
<th>ElemId</th>
<th>Pre</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>a</td>
<td>d3</td>
<td>1/4</td>
<td>e5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>a</td>
<td>d3</td>
<td>1/4</td>
<td>e10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>(A, a, d3,...)</td>
<td>(B, b, d1,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C, c, d2,...)</td>
<td>(A, a, d1,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(B, b, d3,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>b</td>
<td>d1</td>
<td>1/2</td>
<td>e8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>b</td>
<td>d1</td>
<td>1/2</td>
<td>e4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>(B, b, d1,...)</td>
<td>(B, b, d3,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C, c, d2,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>c</td>
<td>d2</td>
<td>1/3</td>
<td>e5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>c</td>
<td>d2</td>
<td>1/3</td>
<td>e7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>(A, a, d2,...)</td>
<td>(B, b, d2,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C, c, d3,...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sorted Block-Scans & Random Range-Scans
Topics: Indexing
IO-Top-k: index-access optimized top-k query processing

- Ranked Retrieval & Top-k query processing.
- Index Lists & aggregate scores.
- Based on the family Threshold-Algorithms.
- Disk accesses:
  - Sorted Accesses & Random Accesses.
  - Scheduling problem!

- What’s new?
  - The integration view of both kinds of accesses.

- Do you want to know more?

Example query: //A ["a", //B["b", //C["c"]]]

Pre-computed index table:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Term</th>
<th>MaxScore</th>
<th>DocId</th>
<th>Score</th>
<th>ElemenId</th>
<th>Pre</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a</td>
<td>1</td>
<td>d3</td>
<td>1/4</td>
<td>e5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>a</td>
<td>1/2</td>
<td>d1</td>
<td>1/2</td>
<td>e2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>a</td>
<td>2/9</td>
<td>d2</td>
<td>2/9</td>
<td>e1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>1</td>
<td>d1</td>
<td>1</td>
<td>e8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>1</td>
<td>d1</td>
<td>1/2</td>
<td>e4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>1</td>
<td>d1</td>
<td>3/7</td>
<td>e6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>1</td>
<td>d1</td>
<td>1/3</td>
<td>e2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>2/3</td>
<td>d2</td>
<td>2/3</td>
<td>e4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>2/3</td>
<td>d2</td>
<td>1/3</td>
<td>e3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>2/3</td>
<td>d2</td>
<td>1/3</td>
<td>e6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>1</td>
<td>d2</td>
<td>1</td>
<td>e5</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>2/3</td>
<td>d3</td>
<td>2/3</td>
<td>e7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>2/3</td>
<td>d3</td>
<td>1/5</td>
<td>e11</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>3/5</td>
<td>d1</td>
<td>3/5</td>
<td>e9</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>3/5</td>
<td>d1</td>
<td>1/2</td>
<td>e5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Sorted Block-Scans & Random Range-Scans
Topics: Indexing
IO-Top-k: index-access optimized top-k query processing

- Ranked Retrieval & Top-k query processing.
- Index Lists & aggregate scores.
- Based on the family Threshold-Algorithms.
- Disk accesses:
  - Sorted Accesses & Random Accesses.
  - Scheduling problem!
- What’s new?
  - The integration view of both kinds of accesses.
- Do you want to know more?

Example query: //A [//“a” //B//“b” //C//“c”]

Pre-computed index table:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Term</th>
<th>MaxScore</th>
<th>DocId</th>
<th>Score</th>
<th>ElemId</th>
<th>Pre</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>a</td>
<td>d3</td>
<td>1/4</td>
<td>e5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>(A, a, d3, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>a</td>
<td>d1</td>
<td>1/2</td>
<td>e2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>(B, b, d1, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>b</td>
<td>d1</td>
<td>1/2</td>
<td>e4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>(C, c, d2, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>a</td>
<td>d3</td>
<td>1/3</td>
<td>e2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>(B, b, d3, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>c</td>
<td>d2</td>
<td>1/3</td>
<td>e5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>(A, a, d1, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>c</td>
<td>d3</td>
<td>2/3</td>
<td>e7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>(B, b, d2, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A, a, d2, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>c</td>
<td>d1</td>
<td>3/5</td>
<td>e9</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>(B, b, d1, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C, c, d1, ...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sorted Block-Scans & Random Range-Scans
Topics: Indexing
A Time Machine for Text Search

- Search web archives.
- Too much data → Huge index size.
- Index compression and other techniques.

**FluxCapacitor**

```
Your query Iraq war@Jun 18, 2002 8:30 PM needed 266ms and has 50 results

Iran-Iraq War
The b Iran-Iraq War b was a border war between Iran and Iraq that took place between September 22, 1980 and August 20, 1988. It is also known as the b First Persian Gulf War b and the b Gulf War b.
Score: 13,385 | Created: May 27, 2002 5:59 PM

Gulf War
The b Gulf War b, also known as b Persian Gulf War b or b War in the Gulf b, was a conflict between Iraq and a coalition force led by the United States.
Score: 13,343 | Created: Jun 18, 2002 10:10 AM

History of Iraq
Ancient Times. For most of historic time the city and empire of Babylon occupied parts of the present-time region of Iraq. There were many dynasties and kingdoms which ruled Babylon and other.
Score: 12.76 | Created: Jun 10, 2002 3:01 AM
```
- Search web archives.
- Too much data → Huge index size.
- Index compression and other techniques. → 18.11.2008

FluxCapacitor

Your query Iraq war@Jun 18, 2002 8:30 PM needed 266ms and has 50 results

**Iraq War**
The b Iraq Iraq War b was a border war between Iran and Iraq that took place between September 22, 1980 and August 20 1988. It is also known as the b First Persian Gulf War b and the b Gulf War b.

**Score:** 13,385 | **Created:** May 27, 2002 5:59 PM

**Gulf War**
The b Gulf War b also known as b Persian Gulf War b b War in the Gulf b b Iraq Kuwait Conflict b b Second Gulf War b or b UN Iraq conflict b was a conflict between Iraq and a coalition force led.

**Score:** 13,343 | **Created:** Jun 18, 2002 10:10 AM

**History of Iraq**
Ancient Times: For most of historic time the city and empire of Babylon occupied parts of the present time region of Iraq. There were many dynasties and kingdoms which ruled Babylon and other.

**Score:** 12,76 | **Created:** Jun 10, 2002 3:01 AM
Growing volumes of XML data everywhere.

XML storing and searching approaches:

- Relational approaches: fast, redesign relational schema each time XML hierarchy is defined.
- Fixed-schema relational approach: potential longer runtimes, but still faster than tree-based.
Growing volumes of XML data everywhere.

XML storing and searching approaches:

- Relational approaches: fast, redesign relational schema each time XML hierarchy is defined.

Fixed-schema relational approach: potential longer runtimes, but still faster than tree-based.

- IR-style keyword-based search on XML document.
- How to return the result as a representation of meaningful related nodes to the user?
  - Store the relationship of each pair of nodes in each XML document to an index → overhead.
- Enhanced inverted index structure that stores path information in addition to node ID.
- IR-style keyword-based search on XML document.
- How to return the result as a representation of meaningful related nodes to the user?
  - Store the relationship of each pair of nodes in each XML document to an index → overhead.
- Enhanced inverted index structure that stores path information in addition to node ID.
A need to a high precision search strategy.
- E.g.: Nuclear analysts.

Semantic search using the XML Fragments query language.

Four query-time semantic needs:
- Specify target information type.
- Disambiguate keywords.
- Specify search term context.
- Relate select terms.

President Clinton was born William Jefferson Blythe IV on August 19, 1946, in Hope, Arkansas, three months after his father died in a traffic accident. When he was four years old, his mother wed Roger Clinton, of Hot Springs, Arkansas. In high school, he took the family name.

Clinton was graduated from Georgetown University and in 1968 won a Rhodes Scholarship to Oxford University. He received a law degree from Yale University in 1973, and entered politics in Arkansas.

Figure 2: Sample Excerpt From Clinton’s Biography

<BirthPlaceOf> <BirthDateOf> <Alias> <Person> President Clinton </Person> was born <Person> William Jefferson Blythe IV </Person> on <Date> August 19, 1946 </Date> </BirthDateOf>, in <City> Hope, Arkansas </City> </BirthPlaceOf>, three months after his father died in a traffic accident. When he was four years old, <SpouseOf> <Person> his mother </Person> wed <Person> Roger Clinton </Person> </SpouseOf>, of <City> Hot Springs, Arkansas </City>. In high school, he took the family name.

Clinton was graduated from <College> Georgetown University </College> </AlmaMater> and in <Date> 1968 </Date> won a Rhodes Scholarship to <College> Oxford University </College>. <AlmaMater> <Person ref="Clinton"> He </Person> received a law degree from <College> Yale University </College> </AlmaMater> in <Date> 1973 </Date>, and entered politics in <UsState> Arkansas </UsState>.

Figure 3: Sample Annotations of Text in Figure 2
Topics: XML Information Retrieval
Semantic search via XML fragments: a high-precision approach to IR

- A need to a high precision search strategy.
  - E.g.: Nuclear analysts.
- Semantic search using the XML Fragments query language.
- Four query-time semantic needs:
  - Specify target information type.
  - Disambiguate keywords.
  - Specify search term context.
  - Relate select terms.

President Clinton was born William Jefferson Blythe IV on August 19, 1946, in Hope, Arkansas, three months after his father died in a traffic accident. When he was four years old, his mother wed Roger Clinton, of Hot Springs, Arkansas. In high school, he took the family name.

Clinton was graduated from Georgetown University and in 1968 won a Rhodes Scholarship to Oxford University. He received a law degree from Yale University in 1973, and entered politics in Arkansas.

Figure 2: Sample Excerpt From Clinton’s Biography

Figure 3: Sample Annotations of Text in Figure 2
Challenges in building a web search engine:
- Web graph analysis.
- Statistical methods for inferring meanings in text.
- Retrieval in newsgroup postings, images, and sounds

Open research problems in the field!
Challenges in building a web search engine:

- Web graph analysis.
- Statistical methods for inferring meanings in text.
- Retrieval in newsgroup postings, images, and sounds

Open research problems in the field!

Looks amazing to know secrets about Google! → 06.01.2009
- A novel context-based technique for the ad-hoc retrieval of web documents.
- Dynamic generation of a measure of document term’s significance during retrieval.
- Document term significance based on term frequency. What about context?
- Term context in document reflects document’s significance.

Figure 1  Overview of context matching as part of the retrieval process.
■ A novel context-based technique for the ad-hoc retrieval of web documents.
■ Dynamic generation of a measure of document term’s significance during retrieval.
■ Document term significance based on term frequency. What about context?
■ Term context in document reflects document’s significance.
■ The whole story on 13.01.2009.

Figure 1 Overview of context matching as part of the retrieval process.
What are result snippets?
Are result snippets important?
What are the algorithms and data structures needed to generate these snippets efficiently in a search engine?
On 20.01.2009.
What are result snippets?

Are result snippets important?

What are the algorithms and data structures needed to generate these snippets efficiently in a search engine?

On 20.01.2009.
What are result snippets?

Are result snippets important?

What are the algorithms and data structures needed to generate these snippets efficiently in a search engine?

On 20.01.2009.