Lucene – eine Demo

Sebastian Marius Kirsch
skirsch@luusa.org

9. Februar 2006

Text Retrieval – wie funktioniert das?

- Aufgabe:
  - Finde zu Stichwörtern die passenden Dokumente
  - Sortiere sie nach Relevanz zur Suchanfrage.
- Probleme:
  - Suche auf Ursprungsdokumenten ist zu langsam.
    Vorverarbeitung ist notwendig!
  - Was ist Relevanz?
- Deshalb: Zweistufiges Verfahren
  - Erstelle aus den Dokumenten einen *Index*:
    Datenstruktur zum schnellen Auffinden von Stichwörtern
  - Suche auf dem Index.

Was ist Lucene?

- `http://lucene.apache.org/`
- Library für Text-Retrieval
- In Java geschrieben, unter Apache-Lizenz
- Keine Suchmaschine!
- Baukasten für Anwendungen, die schnelle Textsuche benötigen
- Robustes dateibasiertes Indexformat
- Schwesterprojekt Nutch implementiert Web-Suchmaschine
- Implementierung in C in den Kinderschuhen

Index-Erstellung

```
15      IndexWriter writer = new IndexWriter(indexdir,
16          new StandardAnalyzer(),
17          true);
18      writer.setInfoStream(System.err);
19
20      File source = new File(sourcedir);
21
22      File[] files = source.listFiles();
```

Index-Erstellung II

```
23      for (int i = 0; i < files.length; i++) {
24        System.err.println("Indexing␣"
25            + files[i].getCanonicalPath());
26        Document doc = new Document();
27        doc.add(new Field("content",
28            new BufferedReader(
29                new FileReader(files[i])),
30            Field.TermVector.YES));
31        doc.add(new Field("filename",
32            files[i].getCanonicalPath(),
33            Field.Store.YES,
34            Field.Index.UN_TOKENIZED));
35        writer.addDocument(doc);
36      }
37
38      writer.optimize();
39      writer.close();
```

Was geschieht bei der Indexierung?

- Lesen des Quelldokuments
- Aufteilen des Dokuments in Wörter („Token“)
- Weiterverarbeitung der Token:
  - Konvertieren in Kleinbuchstaben
  - Entfernen von inhaltsleeren Wörtern („Stoppwörter“)
  - Erzeugen von Grundformen („Stemming“)
  - etc.
- Speichern im Index von
  - Assoziation Token zu Dokument („inverted index“)
  - Häufigkeit der Token im Dokument
  - Position der Token

Suchen

```
14        IndexSearcher searcher =
15          new IndexSearcher(indexdir);
16        Query query = QueryParser.parse(querystring,
17            "content",
18            new StandardAnalyzer());
19
20        Hits hits = searcher.search(query);
21
22        for (int i = 0; i < hits.length() ; i++) {
23          Document doc = hits.doc(i);
24          System.out.println((i + 1) + ":␣"
25              + doc.getField("filename").stringValue()
26              + "␣(" + hits.score(i) + ")");
27        }
```

Was geschieht beim Suchen?

- Analyse der Anfrage
- Suchen nach Dokumenten, die die Anfrageterme enthalten
- Berechnen der Relevanz
- Sortieren
- Zurückgeben
- Grundprinzip: Nur was in den Index geschrieben wurde, kann gefunden werden.
  - Beim Indexieren alles zu Kleinbuchstaben konvertiert $\Rightarrow$ beim Suchen können keine Grossbuchstaben gefunden werden.
  - Beim Indexieren alle Stoppwörter weggeworfen $\Rightarrow$ können nicht gesucht werden.
  - Deshalb: Gleiche Verarbeitungsschritte für Dokumente und Anfrage.

Was kann Lucene noch?

- Viele verschiedene Anfragetypen: PhraseQuery, BooleanQuery, PrefixQuery, FuzzyQuery, WildcardQuery...
- „Baukasten“ für Token-Verarbeitung, mit diversen Filtern
- Verschiedene Feldtypen (indiziert, in Token zerlegt, nur gespeichert, mit Termvektoren, ...)
- Highlighting, Spellchecker, ähnliche Anfragen, Synonyme expandieren, ...
- Powered by Lucene: SourceForge, Wikipedia, CNet, mozDex, Simpy, Technorati, Eclipse, Zoë, (Beagle,) ...
- Bitte Version 1.9 aus Subversion benutzen!

Warum ist Lucene keine Suchmaschine?

- Lucene ist ein Baukasten
- Hat keine Dokumentbeschaffung (Crawler)
- Hat keine Dokumentkonvertierung (HTML, PDF . . . nach Text)
- Hat keine Zeichensatzkonvertierung
- Hat keine Datenbank über indexierte Dokumente
- Hat keine Benutzeroberfläche
- . . .
- Deshalb: Nutch implementiert Suchmaschine mit Lucene als Index-Komponente
- Nutch-Vortrag . . . irgendwann?

Fragen? Feedback?

Vielen Dank für's Zuhören!

Folien gibt es unter

`http://www.sebastian-kirsch.org/moebius/docs/lucenedemo.pdf`

Mehr zu Lucene unter

`http://lucene.apache.org/`

Sehr empfehlenswert: „Lucene in Action“

`http://www.lucenebook.com/`

Lucene – eine Demo

Sebastian Marius Kirsch
skirsch@luusa.org

9. Februar 2006