Information Retrieval - Sommer 2006
Dipl.-Inform. Sascha Kriewel, LF 137
kriewel@is.informatik.uni-duisburg.de

Übungsblatt 5 — Freitext-Indexierung

Aufgabe 12: Informatischer Ansatz

(a) Welche spezifischen Probleme gibt es bei der Freitext-Suche im Gegensatz zur Benutzung von Dokumentationssprachen? Wie versucht der informatische Ansatz diesen zu begegnen, und welche Probleme löst dieser Ansatz nicht?

(b) Die INSPEC-Datenbank ist eine bibliographische Datenbank zu internationaler Fachliteratur der Bereiche Physik, Elektrotechnik, Elektronik, Computer- und Informationstechnik. Von Rechnern im Universitätsnetz (oder über den Proxyserver des HRZs) kann über die Universitätsbibliothek auf die INSPEC-Datenbank zugegriffen werden:

`http://fizweb.fiz-technik.de/cgi-bin/websuche?APPL=uni-duisb-essen&SPRACHE=de`

Finde heraus, welche Möglichkeitten zur Suche die INSPEC-Datenbank zulässt, insbesondere welche Operatoren zur Suche in den Feldern erlaubt sind.

Aufgabe 13: Computerlinguistischer Ansatz

Textdokumente werden üblicherweise zunächst aufbereitet, um eine Verbesserung der IR-Qualität zu erreichen. Überlege, wie sich die folgenden Techniken auf Precision und Recall eines Information-Retrieval-Systems auswirken, unter der Annahme, dass jeweils alle anderen Faktoren unverändert bleiben.

Wirkt sich die Sprache des Textes dabei aus?

(a) Stoppworteliminierung

(b) Stamm- und/oder Grundformreduktion

(c) Mehrwortgruppenerkennung

Aufgabe 14: Computerlinguistischer Ansatz - Praktische Aufgabe

Abgabe bis 18. Mai 2006, 12 Uhr:
`iruebg-abgaben@is.informatik.uni-duisburg.de`

Apache Lucene ist eine Programmbibliothek für Java zur Indexierung von und zur Suche in Texten. Unter anderen stellt sie auch Stemming-Algorithmen in

verschiedenen Sprachen zur Verfügung. Zur Benutzung benötigt man die beiden folgenden Jar-Dateien:

- `lucene-core-1.9.1.jar`
- `contrib/analyzers/lucene-analyzers-1.9.1.jar`
  (enthält u.a. sprachspezifische Stemmer)

Beide Jars sind in einem Zip-Archiv (`lucene-1.9.1.zip`) enthalten, das man von einer der unter dieser Adresse genannten Spiegelseiten beziehen kann:

`http://www.apache.org/dyn/closer.cgi/lucene/java/`

Beschaffe Dir die benötigten Dateien, finde heraus, wie die Bibliothek benutzt wird und schreibe ein kleines Java-Programm, das sich einen Text z.B. aus der deutschen Wikipedia (`http://de.wikipedia.org`) nimmt, zunächst Stoppworte eliminiert, dann ein Stemming durchführt und schliesslich das Vorkommen der gestemmten Formen bestimmt.

Definiere Dir dazu zunächst eine eigene Liste sinnvoller deutscher Stoppworte, statt die vorhandene Liste zu übernehmen. Setze dann einen `Tokenizer` auf den Text an, um ihn in einzelne Worte zu zerlegen. Anschliessend kannst Du den resultierenden `TokenStream` nacheinander durch `LowerCaseFilter`, `StopFilter` und `GermanStemFilter` schicken.

Welche Probleme des automatischen Stemming deutschsprachiger Texte treten dabei zutage? Wie könnte man Komposita besonders behandeln?