1. Übung zur Vorlesung "Internet-Suchmaschinen" im Wintersemester 2007/2008

Dr. Andreas Hotho, Prof. Dr. Gerd Stumme, M.Sc. Wi-Inf. Beate Krause

01. November 2007

1 Information Retrieval – Grundlagen

1. Was unterscheidet Information Retrieval von der Suche in Datenbanken?
2. Nennen Sie jeweils drei Beispiele für Web-IR Anwendungen und herkömmliche (ruhig auch digitale) IR Anwendungen.
3. Grenzen Sie Web-IR Anwendungen von den anderen Anwendungen ab: Welche Besonderheiten weist die Web-Suche auf?
4. Verschiedene Benutzer suchen via Google nach "Titanic" und erhalten folgende Seite (`www.titanic-online.com`):

Werden die Benutzer diese Seite relevant finden? Diskutieren Sie drei verschiedene Suchziele und mögliche Beurteilungen durch die Benutzer.

2 Boolesches Retrieval

Betrachten Sie folgende Dokumente. Jede Zeile stelle ein Dokument dar.

D_1 pickled peppers hot
D_2 pickled peppers mild
D_3 a peck of pickled peppers
D_4 nine days old
D_5 some like it hot
D_6 some like it mild
D_7 some like pickled peppers
D_8 nine days old

1. Können Sie für jedes Dokument eine boolesche Query angeben, die genau dieses Dokument zurückliefert? Unter welchen Bedingungen gelingt dies?
2. Welche Wörter in den vorliegenden Dokumenten sind besonders ungeeignet, um bestimmte Dokumente auszuwählen?
3. Wie geht man in der Regel mit solchen Wörtern um?
4. Können Sie sich denken, warum man den Hamlet-Monolog *to be or not to be* mit dieser Anfrage in einfachen Retrieval-Systemen nicht gut findet?
5. Können Sie sich Abwandlungen des booleschen Modells überlegen, die dessen Ausdrucksmächtigkeit erhöhen oder z. B. ein Ranking von Ergebnissen ermöglichen?

3 Vektorraum-Modell

In der Vorlesung wurde ein Maß $cosSim(a, b)$ für die Ähnlichkeit zweier Dokumente a und b eingeführt. Dementsprechend sei $d_c(a, b) := 1 - cosSim(a, b)$ als Abstandsmaß definiert.

Weiterhin kann man die euklidische Distanz $d_e(a, b) := \|a - b\|_2 := \sqrt{\sum_i (a_i - b_i)^2}$ definieren.

1. Betrachten Sie die folgenden beiden Dokumente:

   D_1 max sagt fischers fritz fischt frische fische
   D_2 moritz sagt fischers fritz fischt frische fische frische fische fischt fischers fritz

   - Stellen Sie die Term-Dokument-Matrix auf. Das Gewicht sei die Termfrequenz ohne Normierung oder TF/IDF-Gewichtung.

- Berechnen Sie den euklidischen und den Kosinusabstand von D_1 und D_2! (Sie brauchen nicht die numerischen Werte auszurechnen, Ausdrücke der Art $3 + \sqrt{19}$ reichen.) Was beobachten Sie?

2. Rechnen Sie nach, in welchem Zusammenhang die beiden Maße stehen, wenn man mit normierten Dokumenten ($\|a\| = \|b\| = 1$) arbeitet!

4 Grundlegendes zu den Praxisübungen

1. Die Webseite zur Übung befindet sich unter `http://www.kde.cs.uni-kassel.de/lehre/ws2007-08/IR/uebungen`. Dort liegt der Programmcode und ein Textkorpus `texte.zip`, der in den Übungen zu Grunde gelegt wird.

2. Machen Sie sich – soweit nicht schon geschehen – mit der Java-API-Dokumentation und einer Java-Entwicklungsumgebung vertraut. Wir empfehlen die Benutzung von Eclipse 3.2 (`http://www.eclipse.org`).

3. In den folgenden Aufgaben werden gelegentlich Features von Java 1.5 benutzt. Vollziehen Sie sie am Beispielprogramm `Jdk15Beispiel` auf der Webseite zur Übung nach.

4. Zur Orientierung, ob Ihr Programm auch die Anforderungen der Aufgabe erfüllt, wird zu jeder Aufgabe eine Testklasse des Java-Frameworks jUnit mitgeliefert. Um diese auszuführen, müssen Sie das JAR-Archiv junit.jar in den CLASSPATH mit aufnehmen. Das Framework ist unter `http://sourceforge.net/projects/junit/` erhältlich.

5 Praxisübung – Bag-of-Words-Modell und boolesches Retrieval (Abgabe: 14.11.2006)

Erstellen Sie Klassen, um Texte in ein Bag-of-Words-Modell einzulesen und darauf (einfache) boolesche Anfragen nach Termen zu ermöglichen.

1. Implementieren Sie das Interface `Document`, welches ein Dokument repräsentiert. Ein `Document` zählt, welcher Term wie oft vorkommt und kann seinen Inhalt aus einem `InputStream` einlesen. Sie können davon ausgehen, daß der Text schon vorverarbeitet vorliegt, also ohne Groß-/Kleinschreibung, Satzzeichen usw.:

```
implementieren sie das interface document welches ein
dokument repräsentiert ein document zählt welcher term
wie oft vorkommt und kann seinen inhalt aus einem
inputstream einlesen
```

2. Implementieren Sie ebenso das Interface `Corpus`, das eine Sammlung von `Document` repräsentiert.

3. Überprüfen Sie Ihre Implementierung anhand des Programms `BooleanTest`.

   - Die Anfrage `corpus.getDocumentsContainingAll("cocoa", "shipment")` sollte die Nummern der Dokumente 1, 5258, 8961 und 13462 liefern.
   - Die Anfrage `corpus.getDocumentsContainingAny("alternative", "daily")` sollte die Nummern der Dokumente 49, 2310, 5258, 6657, 12179, 12772, 12924, 13462 liefern.