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

– mit Lösungsvorschlägen –

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?

   - Strukturierte Daten
   - Genau spezifizierte Anfrage
   - Exakt definierte Resultate
   - Deterministisches Modell
   - Komplexe Anfragesprache

2. Nennen Sie jeweils drei Beispiele für Web-IR Anwendungen und herkömmliche (ruhig auch digitale) IR Anwendungen.

   - Suchmaschinen (Google, Ask, MSN), Web-Kataloge (Yahoo, Web.de), Nachrichtendienste (BBC World Services, Yahoo Nachrichten)
   - Digitale Bibliotheken (CiteSeer, Uni-Bibliothek Kassel), Bildarchive, Desktopsuche

3. Grenzen Sie Web-IR Anwendungen von den anderen Anwendungen ab: Welche Besonderheiten weist die Web-Suche auf?

   - Größe des Korpus
   - Ständige Veränderung des Korpus, Update-Problematik
   - Semistrukturierte Daten, Layoutinformation (Überschriften, etc.)
   - Metadaten verfügbar
   - Ausnutzen der Linkstruktur

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.

Benutzer 1 wollte sich über die Geschichte und das Schicksal der Titanic informieren. Er findet die Seite relevant, weil sie genau diese Art von Inhalten bietet.

Benutzerin 2 interessiert sich für den Film von James Cameron und wollte Filmkritiken dazu lesen. Trotzdem findet sie die Seite relevant, da sie interessante Hintergrundinformationen zum Film bietet.

Benutzer 3 sieht das anders. Er wollte nur wissen, ob Victor Garber in dem Film mitspielte oder nicht. Er findet die Seite irrelevant.

Benutzerin 4 wollte sich über den Titanic-Algorithmus zur Berechnung von Iceberg-Begriffsverbänden informieren. Für sie ist das Resultat nicht relevant.

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? Dies gelingt,

wenn kein Dokument Teilmenge eines anderen ist. Dann kann man durch Aufzählung t_1 *AND* ... *AND* t_k der Terme $t_i, i = 1 \ldots k$ des Dokumentes das Dokument auswählen. Wenn jedoch ein Dokument Teilmenge eines anderen ist (hier etwa $D_4 = D_8$), geht das nicht.

2. Welche Wörter in den vorliegenden Dokumenten sind besonders ungeeignet, um bestimmte Dokumente auszuwählen? Einige Wörter, hier *pickled, peppers, some, like, it*, kommen in vielen Dokumenten vor, so daß sie nicht gut zur Unterscheidung einzelner Dokumente dienen können.

3. Wie geht man in der Regel mit solchen Wörtern um? *Stoppwörter* wie z. B. Artikel, Präpositionen, Personalpronomen usw. werden oft in der Vorverarbeitung entfernt.

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? Die Anfrage besteht nur aus Stoppwörtern. In frühen Suchmaschinen bekam man daher keine Ergebnisse. Heutige Suchmaschinen versuchen z. B. automatisch eine Suche nach der genauen Phrase und liefern relevante Ergebnisse.

5. Können Sie sich Abwandlungen des booleschen Modells überlegen, die dessen Ausdrucksmächtigkeit erhöhen oder z. B. ein Ranking von Ergebnissen ermöglichen?

   - Zählen von Termen anstatt binärer Unterscheidung Vorkommen/Nicht-Vorkommen
   - Fuzzy-Mengenoperatoren: *foo* kommt sehr oft vor, *bar* selten, also *foo AND bar* mit mittlerer Häufigkeit, usw.
   - Unterscheidung, ob Terme zusammen oder weit voneinander entfernt vorkommen
   - Berücksichtigung der Formatierung des Dokuments, z. B.
     - Gesonderte Behandlung von Text in Überschriften
     - Position des Textes im Dokument

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.

	max	moritz	fische	fischers	fischt	frische	fritz	sagt
D_1	1	0	1	1	1	1	1	1
D_2	0	1	2	2	2	2	2	1

- 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?

$$\begin{aligned} d_c(D_1, D_2) &= 1 - \frac{1+2+2+2+2+2}{\sqrt{7}\sqrt{22}} \\ &= 1 - \frac{11}{\sqrt{154}} \\ &\approx 0.11 \end{aligned}$$

$$\begin{aligned} d_e(D_1, D_2) &= \sqrt{1^2 + 1^2 + (5 \cdot 1^2) + 0^2} \\ &= \sqrt{7} \\ &\approx 2.6 \end{aligned}$$

Wie man sieht, liefert die Kosinusdistanz für Dokumente ähnlichen Inhalts, aber unterschiedlicher Länge einen kleinen Abstand, während die Eukliddistanz hier einen großen Abstand ergibt.

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

   Es gilt: $d_c(a, b) = 1 - \frac{\langle a,b \rangle}{\|a\| \, \|b\|} = 1 - \langle a, b \rangle = 1 - \sum_i a_i b_i$

   Für das Euklid-Maß gilt:

$$
\begin{aligned}
d_e(a,b) &= \|a-b\|_2 \\
&= \sqrt{\sum_i (a_i - b_i)^2} \\
&= \sqrt{\sum_i (a_i^2 + b_i^2 - 2a_i b_i)} \\
&= \sqrt{\sum_i a_i^2 + \sum_i b_i^2 - 2\sum_i a_i b_i} \\
&= \sqrt{1 + 1 - 2\ \langle a,b\rangle} \qquad \text{(wegen } \|a\| = \|b\| = 1\text{)} \\
&= \sqrt{2\, d_c(a,b)}
\end{aligned}
$$

Für normierte Dokumente gilt also der einfache Zusammenhang $d_e = \sqrt{2\, d_c}$.

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.