

**Übungsblatt 11**

Kai Großjohann

Abgabe bis 22. Januar 2003

**Aufgabe 1: Verwendung eines IR-Systems**

[http://www.is.informatik.uni-duisburg.de/teaching/lectures/db\\_ws02/index.html](http://www.is.informatik.uni-duisburg.de/teaching/lectures/db_ws02/index.html)

[http://www.dcs.gla.ac.uk/idom/ir\\_resources/test\\_collections/](http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/)

Unter der ersten URL findet sich (unter Blatt 09) ein Link auf `ji`, ein einfaches in Java geschriebenes IR-System. Unter der zweiten URL findet man einen Link auf die CACM-Kollektion. Das `ji`-System ist geeignet, die CACM-Kollektion zu indexieren.

Unter Verwendung dieses Systems soll eine Evaluation durchgeführt werden. Die einzelnen Aufgabenteile sind bereits von Blatt 09 bekannt. Zunächst jedoch muss das System ein wenig erweitert werden, sodass man damit die Untersuchungen durchführen kann:

**Vervollständigt das IR-System** Das System `ji` ist nicht vollständig, d.h. man kann damit die Untersuchungen nicht ohne Weiteres durchführen. Mir fallen folgende Dinge ein, die fehlen:

1. Programm, das die Fragen einliest, jede Frage parst, die entsprechende Anfrage durchführt, die Ergebnisse geeignet abspeichert. (Es gibt bereits die Klasse `ji.Query`, mit der man die Fragen parsen kann, und es gibt die Klasse `ji.Search`, mit der man eine einzelne Anfrage stellen kann. Es bleibt also eine einfache Schleife und das Abspeichern der Ergebnisse.)
2. Programm, das die Datei mit den Relevanzurteilen einliest und die Relevanzinformation zu jedem Anfrageergebnis hinzufügt.
3. Programm, das die Recall-Precision-Kurven zu den Anfrageergebnissen ermittelt. (Siehe auch unten die Erwähnung des Perl-Moduls `RePrec`.)
4. Erweiterung der Gewichtungsfunktion, sodass verschiedene Gewichtungsfunktionen verwendet werden können (nicht nur `tf · idf`, wie in der jetzigen Implementierung).
5. Implementierung von Stoppwortelimination.

Falls noch weitere Dinge fehlen, schreibt sie bitte auf und setzt sie um.

Zeigt mir die Implementierung von allem, was ihr implementiert, zusammen mit einer Dokumentation, mit der ich das verstehen kann.

**Indexiert die Dokumente** Wie groß ist der Index? Gib das Verhältnis zur Größe von `cacm.all` an.

Vermutlich habt ihr nicht alle Abschnitte der Dokumente indexiert (ich würde `.X` und `.N` weglassen). Nehmt euch die Datei `cacm.all`, werft alle nicht indexierten

Abschnitte weg, und vergleicht die daraus resultierende Größe der Dokumente mit der Indexgröße.

Wie groß ist der Index, wenn man Stoppwörter eliminiert? Wie groß ist er, wenn man sie nicht eliminiert?

Wie lange dauert die Indexierung? (Beachte mögliche Einflüsse des Buffer-Cache des Betriebssystems und so weiter. Vielleicht sollte man die Indexierung direkt nach einem Reboot durchführen.)

**Lasst die Anfragen durchlaufen** Lasst alle Anfragen durchlaufen und erstellt Ranglisten. Eventuell ist es günstig, die Ranglisten nach 100 oder 500 oder 1000 Dokumenten abzuschneiden. Oder vielleicht ein Schwellenwert für den RSV?

Wie lange dauert die Anfrageprozessierung? (Schnellste Anfrage, langsamste Anfrage, arithmetisches Mittel.)

**Evaluiert die Retrievalqualität** Ermittelt Recall/Precision-Kurven für die verschiedenen Anfragen. (Also Recall/Precision-Punkt nach jedem Rang und die Punkte durch Geradenstücke verbinden.) Zeigt ein paar typische und ein paar ungewöhnliche Recall/Precision-Kurven. (Vielleicht noch gute und schlechte.)

Erstellt Mittelwerte über alle Fragen, nach dem im Skript genannten Verfahren. D.h. bei bestimmten Recall-Punkten (vielleicht 0.25, 0.5, 0.75) berechnet ihr die Precision (interpoliert gemäß PRR). Dann mittelt ihr alle Precision-Werte für den Recall-Punkt 0.25, entsprechend für 0.5 und 0.75. Anschließend kann man eine schöne Kurve malen.

Unter folgender URL findet ihr das Perl-Modul RePrec, das euch vielleicht bei den Recall/Precision-Kurven helfen kann.

`ftp://patty.is.informatik.uni-duisburg.de/pub/src/Perl/`

**Evaluiert verschiedene Varianten** Führt die Evaluation für verschiedene Gewichtungsfunktionen durch. Diskutiert die Ergebnisse.

Vielleicht sind einige Funktionen bei niedrigen Recall-Niveaus besser, andere bei hohen?

Probiert auch aus, welchen Einfluss Stemming hat. Bringt das was, oder ist es eher schädlich?

Untersucht den Einfluss von Stoppwortelimination auf die Retrievalqualität.

10 Punkte