**Max-Planck-Institut Saarbrücken**
**AG5: Datenbanken und Informationssysteme**
Dozenten: Prof. Dr.-Ing. G. Weikum, Dr. Ralf Schenkel
Betreuer: Dipl.-Inform. Sergej Sizov, Dipl.-Inform. Christian Zimmer

**Informationssysteme (SS 04)**
**Übungsblatt 1**

Ausgabe: 27.April 2004

**Abgabe: 4.Mai 2004 in der Vorlesung**

# Aufgabe 1.1: Relevanzbewertung

Für eine einzelne Anfrage $q$ ist die (unbeschränkte, also quasi top-unendlich) Präzision $r(q)/t(q)$, wobei $r(q)$ die Anzahl relevanter Treffer bezeichnet und $t(q)$ die Anzahl aller Treffer, die die Suchmaschine zur Anfrage $q$ liefert. Für mehrere Anfragen $q_1, \ldots, q_k$ (z.B. einen Benchmark) gilt:

- **Mikrodurchschnitt der Präzision** (micro-averaged precision) $\frac{\sum_{i=1}^{k} r(q_i)}{\sum_{i=1}^{k} t(q_i)}$
- **Makrodurchschnitt der Präzision** (macro-averaged precision) $\sum_{i=1}^{k} \frac{r(q_i)}{t(q_i)}$

a) Diskutieren Sie Vor- und Nachteil der Eignung dieser beiden Bewertungsmaße als globales Gütekriterium von Suchmaschinen.

b) Zeigen Sie, dass der Mikrodurchschnitt der Präzision im folgenden Sinne nichtmonoton ist. Gegeben sei ein Anfrage-Benchmark $q_1, ..., q_k$ und zwei Suchmaschinen $M_1$ und $M_2$, so dass $M_1$ bzgl. der mikrodurchschnittlichen Präzision über diese $k$ Anfragen besser ist als $M_2$. Dann gibt es eine Anfrage $q_{k+1}$, für die $M_1$ und $M_2$ identische Ergebnisse liefern, so dass $M_1$ bzgl. der mikrodurchschnittlichen Präzision über die $k+1$ Anfragen schlechter ist als $M_2$.

# Aufgabe 1.2: Ähnlichkeitsmaße

Zeigen Sie, dass für normalisierte Vektoren (der Länge 1) das Kosinus-Ähnlichkeitsmaß und die Euklidische Distanz im Vektorraummodell dasselbe Anfrageergebnis-Ranking produzieren.

# Aufgabe 1.3: Vektorraummodell

Betrachten Sie den folgenden Korpus, der aus 4 Dokumenten besteht.

- $d_1$: Marcus tried to assissinate Caesar.
- $d_2$: Marcus was a Roman.

- $d_3$: Caesar was a ruler. All Romans were either loyal to Caesar or hated him.
- $d_4$: Everyone is loyal to someone. People only try to assissinate rulers they are not loyal to.

Bei der Extraktion von Features dienen die folgenden Wörter als "*Stoppwörter*" (werden also nicht betrachtet!):

- *a, all, and, are, either, everyone, her, him, is, not, only, or, someone, they, to, was, were, who*

Ferner sollen alle restlichen Wörter nach der folgenden Abbildung auf ihre jeweilige Stammform reduziert werden, und Groß-/Kleinschreibung soll grundsätzlich unwesentlich sein:

- assissinate assassin
- assissinated assassin
- assassination assassin
- loyality loyal
- hated hate
- Roman Rome
- Romans Rome
- ruler rule
- rulers rule
- tried try

a) Bestimmen Sie die idf-Werte aller Terme (also der nach Stoppwortelimination und Stammformreduktion verbleibenden Wörter).

a) Bestimmen Sie für jedes der vier Dokumente den gewichteten Dokumentvektor aufgrund der $tf * idf$-Formel mit normalisierten tf-Werten und (mit Zweierlogarithmus) gedämpften idf-Werten.

a) Betrachten Sie die folgenden Anfragen:

- $q_1$: Who assissinated Caesar?
- $q_2$: Loyality and assassination.

Berechnen Sie die Resultatsranglisten für die beiden Anfragen gemäß Kosinus-Ähnlichkeit.

## Aufgabe 1.4: Threshold Algorithmus

Beweisen Sie die Korrektheit des Threshold-Algorithmus mit sortiertem Zugriff (TA-sorted). Zeigen Sie also, dass der Algorithmus das korrekte Top-$k$-Ergebnis berechnet und auf keinen Fall zu früh terminiert.

## Aufgabe 1.5: Erweiterter Threshold Algorithmus für Web-Portale

Betrachten Sie Top-$k$-Anfragen über mehrere Web-Portale. Jedes Web-Portal bewertet Objekte nach einem bestimmten Kriterium mit einem lokalen Score zwischen 0 und 1, und gesucht sind die besten Objekte bzgl. des über eine fest gegebene Menge von Portalen aufsummierten globalen Scores. Ein konkretes Beispiel wären etwas Restaurants als Objekte und Portale, die ein Restaurant nach Preis, Qualität des Essens und Entfernung vom Wohnort bewerten (also ein *Stadtführer-Portal* im Sinne von Yellow Pages, ein *Gourmet-Portal* und ein *Streetfinder-Portal*).

a) Die Web-Portale können bzgl. ihrer Anfragemöglichkeiten eingeschr"ankt sein. Es gibt Portale, auf denen man wahlfreie und sortierte Zugriffe machen kann (z.B. das Gourmet-Portal), und es gibt Portale, auf denen man nur wahlfreie Zugriffe machen kann (z.B. das Streetfinder-Portal). Die ersteren nennen wir SR-Portale (für *sorted and random access*), die letzteren R-Portale (für *random access*). Erweitern Sie den Threshold Algorithmus, so dass er mit einer Mischung von SR- und R-Portalen umgehen kann.

b) Die Web-Portale unterscheiden sich auch hinsichtlich ihrer Zugriffsgeschwindigkeit bzw. Zugriffskosten. Diskutieren Sie, in welcher Weise dies im Threshold-Algorithmus berücksichtigt werden kann.

c) Nehmen Sie nun an, dass es außer SR- und R-Portalen auch noch S-Portale gibt, die nur sortierten Zugriff unterstützen. Erweitern Sie Ihren Algorithmus aus Teil a), so dass er auch mit diesem Fall umgehen kann.

## Aufgabe 1.6: LSI

Betrachten Sie einen Dokumentenkorpus aus insgesamt 9 Dokumenten $d_1, \ldots, d_9$:

- $\boldsymbol{d}_1$: *Human machine interface for ABC computer applications*
- $\boldsymbol{d}_2$: *A survey of user opinion of computer system response time*
- $\boldsymbol{d}_3$: *The EPS user interface management system*
- $\boldsymbol{d}_4$: *System and human system engineering testing of EPS*
- $\boldsymbol{d}_5$: *Relation of user perceived response time to error measurement*
- $\boldsymbol{d}_6$: *The generation of random, binary, ordered trees*
- $\boldsymbol{d}_7$: *The intersection graph of paths in trees*
- $\boldsymbol{d}_8$: *Graph minors IV: Widths of trees and well-quasi-ordering*
- $\boldsymbol{d}_9$: *Graph minors: A survey*

Diese Dokumente definieren die folgende Term-Dokument-Matrix $A$ mit 12 Termen $t_1, \ldots, t_{12}$ (dargestellt in den Reihen) und den 9 Dokumenten $d_1, \ldots, d_9$ (dargetsellt durch die Spalten):

$A =$

| | $\boldsymbol{d}_1$ | $\boldsymbol{d}_2$ | $\boldsymbol{d}_3$ | $\boldsymbol{d}_4$ | $\boldsymbol{d}_5$ | $\boldsymbol{d}_6$ | $\boldsymbol{d}_7$ | $\boldsymbol{d}_8$ | $\boldsymbol{d}_9$ |
|---|---|---|---|---|---|---|---|---|---|
| $\boldsymbol{t}_1$= *human* | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_2$= *interface* | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_3$= *computer* | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_4$= *user* | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_5$= *system* | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_6$= *response* | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_7$= *time* | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_8$= *EPS* | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\boldsymbol{t}_9$= *survey* | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\boldsymbol{t}_{10}$= *trees* | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| $\boldsymbol{t}_{11}$= *graph* | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $\boldsymbol{t}_{12}$= *minors* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Die Singulärwertdekomposition of $A$ nach $U$, $\Delta$ and $V^T$ ($A = U \times \Delta \times V^T$) liefert das folgende Ergebnis:

$$U = \begin{pmatrix} 0,22 & -0,11 & 0,29 & -0,41 & -0,11 & -0,34 & 0,52 & -0,06 & -0,41 \\ 0,20 & -0,07 & 0,14 & -0,55 & 0,28 & 0,50 & -0,07 & -0,01 & -0,11 \\ 0,24 & 0,04 & -0,16 & -0,59 & -0,11 & -0,25 & -0,30 & 0,06 & 0,49 \\ 0,40 & 0,06 & -0,34 & 0,10 & 0,33 & 0,38 & 0,00 & 0,00 & 0,01 \\ 0,64 & -0,17 & 0,36 & 0,33 & -0,16 & -0,21 & -0,17 & 0,03 & 0,27 \\ 0,27 & 0,11 & -0,43 & 0,07 & 0,08 & -0,17 & 0,28 & -0,02 & -0,05 \\ 0,27 & 0,11 & -0,43 & 0,07 & 0,08 & -0,17 & 0,28 & -0,02 & -0,05 \\ 0,30 & -0,14 & 0,33 & 0,19 & 0,11 & 0,27 & 0,03 & -0,02 & -0,17 \\ 0,21 & 0,27 & -0,18 & -0,03 & -0,54 & 0,08 & -0,47 & -0,04 & -0,58 \\ 0,01 & 0,49 & 0,23 & 0,03 & 0,59 & -0,39 & -0,29 & 0,25 & -0,23 \\ 0,04 & 0,62 & 0,22 & 0,00 & -0,07 & 0,11 & 0,16 & -0,68 & 0,23 \\ 0,03 & 0,45 & 0,14 & -0,01 & -0,30 & 0,28 & 0,34 & 0,68 & 0,18 \end{pmatrix}$$

$$\Delta = \begin{pmatrix} 3,34 & & & & & & & \dots & 0 \\ & 2,54 & & & & & & & \vdots \\ & & 2,35 & & & & & & \\ & & & 1,64 & & & & & \\ & & & & 1,50 & & & & \\ & & & & & 1,31 & & & \\ & & & & & & 0,85 & & \\ \vdots & & & & & & & 0,56 & \\ 0 & \dots & & & & & & & 0,36 \end{pmatrix}$$

$$V^T = \begin{pmatrix} 0,20 & 0,61 & 0,46 & 0,54 & 0,28 & 0,00 & 0,01 & 0,02 & 0,08 \\ -0,06 & 0,17 & -0,13 & -0,23 & 0,11 & 0,19 & 0,44 & 0,62 & 0,53 \\ 0,11 & -0,50 & 0,21 & 0,57 & -0,51 & 0,10 & 0,19 & 0,25 & 0,08 \\ -0,95 & -0,03 & 0,04 & 0,27 & 0,15 & 0,02 & 0,02 & 0,01 & -0,03 \\ 0,05 & -0,21 & 0,38 & -0,21 & 0,33 & 0,39 & 0,35 & 0,15 & -0,60 \\ -0,08 & -0,26 & 0,72 & -0,37 & 0,03 & -0,30 & -0,21 & 0,00 & 0,36 \\ 0,18 & -0,43 & -0,24 & 0,26 & 0,67 & -0,34 & -0,15 & 0,25 & 0,04 \\ -0,01 & 0,05 & 0,01 & -0,02 & -0,06 & 0,45 & -0,76 & 0,45 & -0,07 \\ -0,06 & 0,24 & 0,02 & -0,08 & -0,26 & -0,62 & 0,02 & 0,52 & -0,45 \end{pmatrix}$$

Betrachten Sie nur die drei größten Singulärwerte, d.h. die approximierte SVD mit $k = 3$!

a) Vergleichen Sie Dokument $d_4$ mit allen anderen unter Verwendung der LSI Methode.

b) Bestimmen Sie die zwei *relevantesten* Dokumente unter Verwendung von LSI für die Anfrage *human interface user* mit den Termen $t_1, t_2, t_3$ (Verwenden Sie das einfachere Skalarprodukt als Ähnlichkeitsmaß).

## Aufgabe 1.7: LSI

Betrachten Sie das LSI Beispiel mit den Kochrezepten aus der Vorlesung. Berechnen Sie die approximierte SVD auf Basis der $k = 2$ größten Singulärwerten. Werten Sie die Anfragen *baking* and *baking bread* mit dieser Basis aus z.B. durch Verwendung der approximierten Term-Dokument-Matrix $A_2$.

Zur Vereinfachung verwenden Sie das Skalarprodukt als Ähnlichkeitsmaß zweier Vektoren (Kosinusmaß ist nicht notwendig). Vergleichen Sie Ihr Resultat mit dem der Vorlesung, bei dem die drei größten Singulärwerte benutzt wurden.