

3 Alternative IR-Modelle

- **Probabilistisches Retrieval:**

Ranking aufgrund von Relevanzwahrscheinlichkeiten, die aus - geschätzten - Basisparametern abgeleitet werden.

- **Fuzzy-Set-Modell:**

Queries (inkl. einzelner Terme) beschreiben Fuzzy-Mengen mit Dokumenten als Elementen vom Grad $\mu \in [0,1]$. Mengenoperationen verwenden Funktionen max, min, $1-\mu$.

- **Latent Semantic Indexing:**

Berücksichtigung von Termkorrelationen durch Transformation des Term-Vektorraums in einen Themen-Vektorraum niedrigerer Dimensionalität

- **Neuronale Netze** und andere Inferenznetze zum Lernen von Termgewichten

Exkurs: Wahrscheinlichkeitstheorie

Ein **Wahrscheinlichkeitsraum** ist ein Tripel (Ω, E, P) mit

- einer Menge Ω elementarer Ereignisse,
- einer Familie E von Teilmengen von Ω mit $\Omega \in E$, die unter \cap , \cup und $-$ mit abzählbar vielen Operanden abgeschlossen ist (bei endlichem Ω ist in der Regel $E=2^\Omega$), und
- einem W.maß $P: E \rightarrow [0,1]$ mit $P[\Omega]=1$ und $P[\cup_i A_i] = \sum_i P[A_i]$ für abzählbar viele, paarweise disjunkte A_i

Eigenschaften von P :

$$\begin{aligned} P[A] + P[\neg A] &= 1 & P[\emptyset] &= 0 \\ P[A \cup B] &= P[A] + P[B] - P[A \cap B] & P[\Omega] &= 1 \end{aligned}$$

Eine **Zufallsvariable** X über einem W.raum (Ω, E, P) ist eine Funktion $X: \Omega \rightarrow M$ mit $M \subseteq \mathbb{R}$, so daß $\{e \mid X(e) \leq x\} \in E$ für alle $x \in M$.

$F_X: M \rightarrow [0,1]$ mit $F_X(x) = P[X \leq x]$ heißt Verteilungsfunktion von X ; bei abzählbarer Menge M heißt $f_X: M \rightarrow [0,1]$ mit $f_X(x) = P[X = x]$ Dichtefunktion von X , ansonsten ist $f_X(x)$ durch $F'_X(x)$ gegeben

Exkurs: Satz von Bayes

Zwei Ereignisse A, B eines W.raums heißen **unabhängig**, wenn gilt $P[A \cap B] = P[A] P[B]$.

Die **bedingte Wahrscheinlichkeit** $P[A \mid B]$ von A unter der Bedingung (Hypothese) B ist definiert als: $P[A \mid B] = \frac{P[A \cap B]}{P[B]}$

Satz von der totalen Wahrscheinlichkeit:

Für eine Partitionierung von Ω in Ereignisse B_1, \dots, B_n gilt:

$$P[A] = \sum_{i=1}^n P[A \mid B_i] P[B_i]$$

Satz von Bayes: $P[A \mid B] = \frac{P[B \mid A] P[A]}{P[B]}$

A-Posteriori-W.
von A

Probabilistisches Retrieval: Grundsätze

Ziel:

Ranking aufgrund von $\text{sim}(d, q) = P[R \mid d]$ [Dokument d ist für Query q relevant | d hat den Termvektor X_1, \dots, X_m]

Annahmen:

- Relevante und nichtrelevante Dokumenten unterscheiden sich in ihren Termen
- Binary Independence Retrieval (BIR) Model:
 - Die Wahrscheinlichkeiten des Auftretens eines Terms sind paarweise unabhängig.
 - Termgewichte sind binär $\in \{0,1\}$.
- Für Terme, die nicht in q vorkommen, sind die Wahrscheinlichkeiten des Auftretens in relevanten und in irrelevanten Dokumenten identisch.

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-3

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-4

Probabilistisches Retrieval: Ranking proportional zur Relevanz-Quote

$$\text{sim}(d, q) = O(R \mid d) = \frac{P[R \mid d]}{P[\neg R \mid d]} \quad (\text{Quote („Odds“) für Relevanz})$$

$$= \frac{P[d \mid R] \times P[R]}{P[d \mid \neg R] \times P[\neg R]} \quad (\text{Satz von Bayes})$$

$$\sim \frac{P[d \mid R]}{P[d \mid \neg R]} = \prod_i \frac{P[X_i \mid R]}{P[X_i \mid \neg R]} \quad (\text{independence bzw. linked dependence})$$

$$\text{sim}(d, q) \sim \log \prod_{i \in q} \frac{P[X_i \mid R]}{P[X_i \mid \neg R]} \quad (X_i = 1, \text{ falls } d \text{ den } i\text{-ten Term enthält, 0 sonst})$$

$$= \sum_i \log P[X_i \mid R] - \log P[X_i \mid \neg R]$$

Probabilistisches Retrieval: Ranking proportional zur Relevanz-Quote (Forts.)

$$= \sum_i \log(p_i^{X_i} (1-p_i)^{1-X_i}) - \log(q_i^{X_i} (1-q_i)^{1-X_i}) \quad (\text{binary features})$$

mit $p_i = P[X_i=1 \mid R]$ und $q_i = P[X_i=1 \mid \neg R]$

$$= \sum_i \log\left(\frac{p_i^{X_i} (1-p_i)^{1-X_i}}{(1-p_i)^{X_i}}\right) - \log\left(\frac{q_i^{X_i} (1-q_i)^{1-X_i}}{(1-q_i)^{X_i}}\right)$$

$$= \sum_i X_i \log \frac{p_i}{1-p_i} + \sum_i X_i \log \frac{1-q_i}{q_i} + \sum_i \log \frac{1-p_i}{1-q_i}$$

$$\sim \sum_i X_i \log \frac{p_i}{1-p_i} + \sum_i X_i \log \frac{1-q_i}{q_i} = \text{sim}(d, q)''$$

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-5

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-6

Probabilistisches Retrieval: Formel von Robertson und Sparck Jones

Schätze p_i und q_i auf der Basis einer Trainingsstichprobe (Anfrage q auf kleinem Trainingskorpus) bzw. der intellektuellen Bewertung eines ersten (groben) Resultats (Relevanz-Feedback):

Sei V die Kardinalität einer Stichprobe relevanter Dokumente
 V_i die Anzahl der Dok. in V , die den i -ten Term enthalten
 n_i eine Schätzung für die Anz. der Dok, die den i -ten Term enthalten

$$\Rightarrow \text{Schätzung: } p_i = \frac{V_i}{V} \quad q_i = \frac{n_i - V_i}{N - V}$$

$$\text{bzw.: } p_i = \frac{V_i + 0.5}{V + 1} \quad q_i = \frac{n_i - V_i + 0.5}{N - V + 1}$$

$$\Rightarrow \text{sim}(d, q)' = \sum_i X_i \log \frac{V_i + 0.5}{V - V_i + 0.5} + \sum_i X_i \log \frac{N - n_i - V_i + 0.5}{n_i - V_i + 0.5}$$

$$\Rightarrow \text{Gewicht von Term } i \text{ in } d: \log \frac{(V_i + 0.5)(N - n_i - V_i + 0.5)}{(V - V_i + 0.5)(n_i - V_i + 0.5)}$$

Probabilistisches Retrieval: Bezug zur tf-idf-Formel

Annahmen (ohne Trainingsstichprobe oder Relevanz-Feedback):

- p_i ist identisch für alle i
- Die meisten Dokumenten sind irrelevant.
- Jeder einzelne Term kommt selten vor.

Daraus folgt:

$$\bullet \sum_i X_i \log \frac{p_i}{1 - p_i} = c \sum_i X_i \text{ mit Konstante } c$$

$$\bullet q_i = P[X_i = 1 | \neg R] \approx \frac{df_i}{N}$$

$$\bullet \frac{1 - q_i}{q_i} = \frac{N - df_i}{df_i} \approx \frac{N}{df_i}$$

$$\Rightarrow \text{sim}(d, q)' = \sum_i X_i \log \frac{p_i}{1 - p_i} + \sum_i X_i \log \frac{1 - q_i}{q_i}$$

$$\approx c \sum_i X_i + \sum_i X_i \log \frac{1}{df_i}$$

Skalarprodukt des Produkts von c und gedämpften idf-Werten über die Anfrageterme

Beispiel für Probabilistisches Retrieval

Dokumente mit Relevanz-Feedback:

| | t1 | t2 | t3 | t4 | t5 | t6 | R |
|-------|-----|-----|-----|-----|-----|-----|---|
| d1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| d2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| d3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| d4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| n_i | 2 | 1 | 2 | 3 | 2 | 0 | |
| V_i | 2 | 1 | 1 | 2 | 1 | 0 | |
| p_i | 5/6 | 1/2 | 1/2 | 5/6 | 1/2 | 1/6 | |
| q_i | 1/6 | 1/6 | 1/2 | 1/2 | 1/2 | 1/6 | |

q : t1 t2 t3 t4 t5 t6

$V=2, N=4$

Bewertung (Retrieval Status Value) eines neuen Dokuments:

$$d5 \cap q: 1 \ 1 \ 0 \ 0 \ 1 \ 1 \rightarrow \text{sim}(d5, q) = \log 5 + \log 1 + \log 0.2$$

$$+ \log 5 + \log 5 + \log 5$$

$$\text{sim}(d, q)' = \sum_i X_i \log \frac{p_i}{1 - p_i} + \sum_i X_i \log \frac{1 - q_i}{q_i}$$

Ansätze zur Erweiterung des Probabilistischen Retrievals

- 1) Berücksichtigung von Termhäufigkeiten in Dok. (nichtbinäre X_i)
 → benötigt Annahmen über Termverteilung wie z.B.

- a) Gleichverteilung:

$P[\text{zufällig im Korpus gezogener Term} = t_i] = 1/m =: p_i$

$$\Rightarrow P[tf_{ij} = k | d_j \text{ hat } l \text{ Wörter}] = \binom{l}{k} p_i^k (1 - p_i)^{l-k} \quad \text{Binomialverteilung}$$

- b) Zipf-Verteilung:

$P[\text{zufällig im Korpus gezogener Term} = t_i] = (1/i) / H_m =: p_i$

$$\text{mit } H_m = \sum_{j=1}^m \frac{1}{j}$$

- c) 2-(Klassen-)Poisson-Verteilung:

$$P[tf_{ij} = k] = p_{i1} e^{-l_{i1}} \frac{l_{i1}^k}{k!} + p_{i2} e^{-l_{i2}} \frac{l_{i2}^k}{k!} \quad \text{mit Konstanten } \pi_{i1} \text{ und } \pi_{i2}$$

Ansätze zur Erweiterung des Probabilistischen Retrievals

- 2) Berücksichtigung von Termkorrelationen in Dok. (mit binären X_i)
 → Problem der Schätzung m -dimensionaler W-Verteilungen
 $P[X_1 = \dots \wedge X_2 = \dots \wedge \dots \wedge X_m = \dots] =: f_X(X_1, \dots, X_m)$

Ein Ansatz – Tree Dependence Model:

- a) Betrachte nur 2-dimensionale Wahrscheinlichkeiten (für Termpaare)
 $f_{ij}(X_i, X_j) = P[X_i = \dots \wedge X_j = \dots] = \sum_{X_1} \dots \sum_{X_{i-1}} \sum_{X_{i+1}} \dots \sum_{X_{j-1}} \sum_{X_{j+1}} \dots \sum_{X_m} P[X_1 = \dots \wedge \dots \wedge X_m = \dots]$
- b) Für jedes Term paar
 schätze den Fehler der Unabhängigkeitsannahme bzw. die Korrelation
- c) Konstruiere einen Baum mit Termen als Knoten und den $m-1$ höchsten Korrelationen als gewichteten Kanten

Bewertung zweidimensionaler Termkorrelationen (2b)

Variante 1:

Fehler der Approximation von f durch g
 (insbesondere g mit paarweiser Termunabhängigkeit):

$$e(f, g) := \sum_{\vec{X} \in \{0,1\}^m} f(\vec{X}) \log \frac{f(\vec{X})}{g(\vec{X})} = \sum_{\vec{X} \in \{0,1\}^m} f(\vec{X}) \log \frac{f(\vec{X})}{\prod_{i=1}^m g_i(X_i)}$$

Variante 2:

Korrelationskoeffizient von Term paaren

$$r(X_i, X_j) := \frac{\text{Cov}(X_i, X_j)}{\sqrt{\text{Var}(X_i)} \sqrt{\text{Var}(X_j)}}$$

mit Kovarianz bzw. Varianz:

$$\text{Cov}(X_i, X_j) := E[(X_i - E[X_i])(X_j - E[X_j])]$$

$$\text{Var}(X_i) := E[(X_i - E[X_i])^2] = \text{Cov}(X_i, X_i) = E[X_i^2] - E[X_i]^2$$

$$\text{und Erwartungswert } E[\dots]: E[X_i] := \sum_{k=0}^1 k f_{X_i}(k) = \sum_{k=0}^1 k P[X_i = k]$$

Beispiel für Approximationsfehler

$m=2$:

Gegeben seien Dokumente

$d1=(1,1)$, $d2=(0,0)$, $d3=(1,1)$, $d4=(0,1)$

Schätzung der 2-dimensionalen W. verteilung f:

$f(1,1) = P[X1=1 \wedge X2=1] = 2/4$

$f(0,0) = 1/4$, $f(0,1) = 1/4$, $f(1,0) = 0$

Schätzung der 1-dimensionalen Randverteilung $g1$ und $g2$:

$g1(1) = P[X1=1] = 2/4$, $g1(0) = 2/4$

$g2(1) = P[X2=1] = 3/4$, $g2(0) = 1/4$

Schätzung für die 2-dim. W. verteilung g mit unabhängigen X_i :

$g(1,1) = g1(1) \cdot g2(1) = 3/8$,

$g(0,0) = 1/8$, $g(0,1) = 3/8$, $g(1,0) = 1/8$

Approximationsfehler ϵ :

$\epsilon = 2/4 \log 4/3 + 1/4 \log 2 + 1/4 \log 2/3 + 0$

Bewertung zweidimensionaler Termkorrelationen (2c)

Gegeben:

Vollständiger Graph (V, E) mit m Knoten $X_i \in V$ und m^2 – mit ϵ bzw. p gewichteten – ungerichteten Kanten $\in E$

Gesucht:

Spannbaum (V, E') mit maximaler Gewichtssumme

Algorithmus:

Sortiere die m^2 Kanten von E nach absteigendem Gewicht

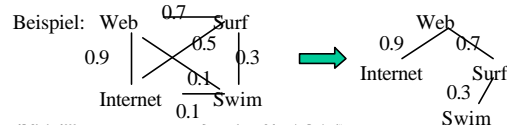
$E' := \emptyset$

Wiederhole bis $|E'| = m-1$

$E' := E' \cup \{(i,j) \in E \mid (i,j) \text{ hat max. Gewicht in } E\}$

sofern E' azyklisch bleibt;

$E := E - \{(i,j) \in E \mid (i,j) \text{ hat max. Gewicht in } E\}$



Schätzung mehrdimensionaler Wahrscheinlichkeiten mit dem Termpaarbaum

Gegeben sei der Termpaarbaum $(V = \{X1, \dots, Xm\}, E')$.

O.B.d.A. sei $X1$ die Wurzel, und sei angenommen, daß

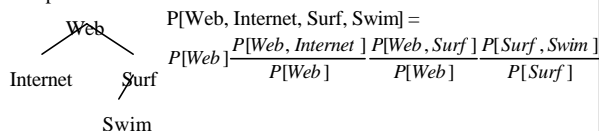
X_i und X_j für $(i,j) \notin E'$ unabhängig sind. Dann gilt:

$$P[X1 = \dots \wedge \dots \wedge Xm = \dots] = P[X1 = \dots] P[X2 = \dots \wedge \dots \wedge Xm = \dots \mid X1 = \dots]$$

$$= P[X1] \cdot \prod_{(1,i) \in E'} P[X_i \mid X1] \cdot P[\bar{X}_j \text{ mit } (1,j) \notin E' \mid X1, \bar{X}_i \text{ mit } (1,i) \in E']$$

$$= P[X1] \cdot \prod_{(i,j) \in E'} P[X_j \mid X_i] = P[X1] \cdot \prod_{(i,j) \in E'} \frac{P[X_i, X_j]}{P[X_i]}$$

Beispiel:



Latent Semantic Indexing (LSI): Grundsätze

Ziel:

Transformation der Dokumentvektoren vom hochdimensionalen Termvektorraum in einen

Themenvektorraum niedrigerer Dimensionalität unter

- Ausnutzung von Korrelationen zwischen Termen (z.B. „Web“ und „Internet“ häufig zusammen)
- implizite Differenzierung von Polysemen, die sich in ihren Korrelationen mit anderen Termen unterscheiden (z.B. „Java“ mit „Library“ vs. „Java“ mit „Kona Blend“ vs. „Java“ mit „Borneo“)

mathematisch:

gegeben: m Terme, n Dokumente (i.d.R. $n > m$) und eine

$m \times n$ -Term-Dokument-Ähnlichkeitsmatrix A ,

gesucht: möglichst gute – Ähnlichkeitsbewahrende –

Abbildung der Spaltenvektoren von A in einen k -dimensionalen Vektorraum ($k \ll m$) für gegebenes k

Exkurs: Singulärwertdekomposition (SVD)

Satz:

Jede reellwertige $m \times n$ -Matrix A mit Rang r kann zerlegt werden

in die Form $A = U \cdot D \cdot V^T$

mit einer $m \times r$ -Matrix U mit orthonormalen Spaltenvektoren,

einer $r \times r$ -Diagonalmatrix D und

einer $n \times r$ -Matrix V mit orthonormalen Spaltenvektoren.

Diese Zerlegung heißt Singulärwertdekomposition und ist eindeutig, wenn die Elemente von Δ der Größe nach geordnet werden.

Satz:

In der Singulärwertdekomposition $A = U \times \Delta \times V^T$ der Matrix A

sind U , Δ und V wie folgt bestimmt:

- Δ besteht aus den Singulärwerten von A , d.h. den positiven Wurzeln der Eigenwerte von $A^T \times A$,
- die Spaltenvektoren von U sind die Eigenvektoren von $A \times A^T$,
- die Spaltenvektoren von V sind die Eigenvektoren von $A^T \times A$.

Exkurs: SVD als Regressionsverfahren

Satz:

Sei A eine $m \times n$ -Matrix mit Rang r und sei $A_k = U_k \times \Delta_k \times V_k^T$, wobei die $k \times k$ -Diagonalmatrix Δ_k die k größten Singulärwerte von A enthält und die $m \times k$ -Matrix U_k sowie die $n \times k$ -Matrix V_k aus den zugehörigen Eigenvektoren der Singulärwertdekomposition von A bestehen.

Unter allen $m \times n$ -Matrizen C mit einem Rang, der nicht größer als k ist, ist A_k diejenige Matrix, die den Wert

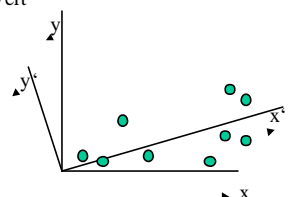
$$\|A - C\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (A_{ij} - C_{ij})^2$$

minimiert (die Frobenius-Norm).

Beispiel:

$m=2, n=8, k=1$

Projektion auf x' -Achse minimiert „Fehler“ bzw. maximiert Varianz im k -dimensionalen Raum



Anwendung der SVD auf das Vektorraummodell

A ist die $m \times n$ -Term-Dokument-Ähnlichkeitsmatrix. Dann sind:

- U bzw. U_k die $m \times r$ - bzw. $m \times k$ -Term-Themen-Ähnlichkeitsmatrix,
- V bzw. V_k die $n \times r$ - bzw. $n \times k$ -Dokument-Themen-Ähnlichkeitsmatrix
- $A \times A^T$ bzw. $A_k \times A_k^T$ die $m \times m$ -Term-Term-Ähnlichkeitsmatrix,
- $A^T \times A$ bzw. $A_k^T \times A_k$ die $n \times n$ -Dokument-Dokument-Ähnlichkeitsmatrix

Indexierung und Anfrageauswertung:

- Die Matrix V_k^T entspricht einem „Themen-Index“ und ist in einer geeigneten Datenstruktur zu verwalten.
- Zusätzlich muß die Term-Themen-Abbildung U_k gespeichert werden.
- Eine Anfrage q (ein $1 \times m$ -Zeilenvektor) im Termvektorraum wird in die Anfrage $q' = q \times U_k$ (ein $1 \times k$ -Zeilenvektor) transformiert und dann im Themenvektorraum (also V_k) ausgewertet (z.B. mittels Skalarproduktmaß $q' \times V_k^T$ oder Cosinusmaß)
- Ein neues Dokument d (ein $m \times 1$ -Spaltenvektor) wird in $d' = U_k^T \times d$ (ein $k \times 1$ -Spaltenvektor) transformiert und als neue Spalte an den „Index“ V_k^T angefügt („folding-in“)

Beispiel 1 für Latent Semantic Indexing

$m=5$ (Lafontaine, Schröder, Euro, Uefa, Beckenbauer), $n=7$

$$A = \begin{pmatrix} 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 1 \\ 0 & 0 & 0 & 0 & 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 0.58 & 0.00 \\ 0.58 & 0.00 \\ 0.58 & 0.00 \\ 0.00 & 0.71 \\ 0.00 & 0.71 \end{pmatrix} \times \begin{pmatrix} 9.64 & 0.00 \\ 0.00 & 5.29 \end{pmatrix} \times \begin{pmatrix} 0.18 & 0.36 & 0.18 & 0.90 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.53 & 0.80 & 0.27 \end{pmatrix}$$

$U \quad \Delta \quad V^T$

Anfrage $q = (0 \ 0 \ 1 \ 0 \ 0)$ wird in

$q' = q \times U = (0.58 \ 0.00)$ transformiert und gegen V^T evaluiert.

Neues Dokument $d_8 = (1 \ 1 \ 0 \ 0 \ 0)^T$ wird in

$d_8' = U^T \times d_8 = (1.16 \ 0.00)^T$ transformiert und an V^T angefügt.

Beispiel 2 für Latent Semantic Indexing

$m=6$ terms

t1: bak(e,ing)
t2: recipe(s)
t3: bread
t4: cake
t5: pastr(y,ies)
t6: pie

$n=5$ documents

d1: How to bake bread without recipes
d2: The classic art of Viennese Pastry
d3: Numerical recipes: the art of scientific computing
d4: Breads, pastries, pies and cakes: quantity baking recipes
d5: Pastry: a book of best French recipes

$$A = \begin{pmatrix} 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.5774 & 0.0000 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.4082 & 0.7071 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \end{pmatrix}$$

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-21

Beispiel 2 für Latent Semantic Indexing (2)

$$A = \begin{pmatrix} 0.2670 & -0.2567 & 0.5308 & -0.2847 \\ 0.7479 & -0.3981 & -0.5249 & 0.0816 \\ 0.2670 & -0.2567 & 0.5308 & -0.2847 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 \\ 0.5198 & 0.8423 & 0.0838 & -0.1158 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 \end{pmatrix} \times \begin{pmatrix} 1.6950 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.1158 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.8403 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.4195 \end{pmatrix} \times \begin{pmatrix} 0.4366 & 0.3067 & 0.4412 & 0.4909 & 0.5288 \\ -0.4717 & 0.7549 & -0.3568 & -0.0346 & 0.2815 \\ 0.3688 & 0.0998 & -0.6247 & 0.5711 & -0.3712 \\ -0.6715 & -0.2760 & 0.1945 & 0.6571 & -0.0577 \end{pmatrix}$$

$U \quad \Delta \quad V^T$

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-22

Beispiel 2 für Latent Semantic Indexing (3)

$$A_3 = \begin{pmatrix} 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.6003 & 0.0094 & 0.9933 & 0.3858 & 0.7091 \\ 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \\ -0.0326 & 0.9866 & 0.0094 & 0.4402 & 0.7043 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \end{pmatrix} = U_3 \times \Delta_3 \times V_3^T$$

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-23

Beispiel 2 für Latent Semantic Indexing (4)

Anfrage q: baking bread

$q = (1 \ 0 \ 1 \ 0 \ 0 \ 0)$

Transformation in den Themenraum mit $k=3$

$q' = q \times U_k = (0.5340 \ -0.5134 \ 1.0616)$

Skalarprodukt-Ähnlichkeit im Themenraum mit $k=3$:

$\text{sim}(q, d1) = q' \times V_{k1}^T \approx 0.8$ $\text{sim}(q, d2) = q' \times V_{k2}^T \approx -0.2$

$\text{sim}(q, d3) = q' \times V_{k3}^T \approx -0.2$ usw.

Folding-in eines neuen Dokuments d6:

algorithmic recipes for the computation of pie

$d_6 = (0 \ 0.7071 \ 0 \ 0 \ 0 \ 0.7071)^T$

Transformation in den Themenraum mit $k=3$

$d_6' = U_k^T \times d_6 \approx (0.5 \ -0.28 \ -0.15)$

d_6' als neue Spalte an V_k^T anhängen

27. Oktober 2000

Stammvorlesung „Information Retrieval“

3-24

Mehrsprachiges Retrieval mit LSI

- Konstruiere LSI-Modell (U_k, Δ_k, V_k^T) anhand von Trainingsdokumenten, die mehrsprachig vorliegen:
 - Betrachte alle Sprachversionen eines Dokuments als ein einziges Dokument und
 - extrahiere zur Indexierung alle Terme oder Wörter unabhängig von der Sprache.
- Indexiere weitere Dokumente durch „folding-in“, also Abbildung in den Themen-Vektorraum und Anhängen an V_k^T .
- Anfragen können dann in beliebiger Sprache gestellt werden und liefern Antworten in allen Sprachen.

Beispiel:

d1: How to bake bread without recipes .

Wie man ohne Rezept Brot backen kann.

d2: Pastry : a book of best French recipes .

Gebäck: eine Sammlung der besten französischen Rezepte.

Terme sind dann z.B. bake, bread, recipe, backen, Brot, Rezept, usw.
Dokumente und Terme werden auf einen kompakten Themenraum abgebildet.

Zusammenfassung zu LSI

- + Elegantes, mathematisch wohlfundiertes Modell
- + „Automatisches Lernen“ von Termkorrelationen (inkl. morphologischer Wortvarianten, Mehrsprachigkeit)
- + Impliziter Thesaurus (durch Korrelation von Synonymen)
- + Implizite Diskriminierung der verschiedenen Bedeutungen von Polysemen (durch verschiedene Korrelationen)
- + Verbesserte Retrievalgüte auf „geschlossenen“ Korpora (z.B. TREC-Benchmark, Finanznachrichten, Patentkollektionen u.ä.) mit empirisch günstigstem k in der Größenordnung 100-200
- Schwierige Wahl von günstigem k
- Rechen- und Speicheraufwand für sehr große (z.T. aber dünn besetzte) Matrizen
- Keine überzeugenden Resultate für Web-Suchmaschinen