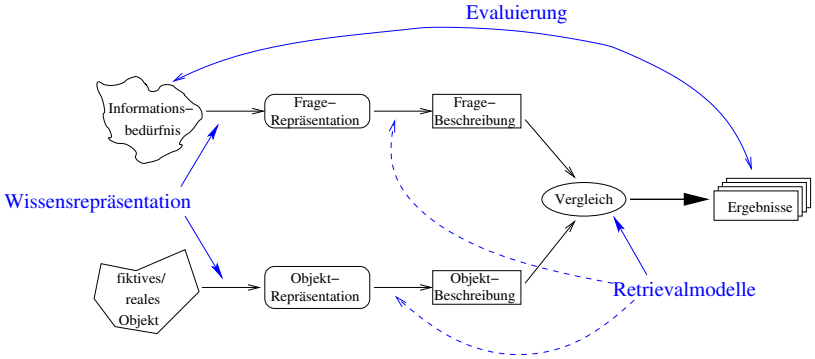
Internet-Suchmaschinen
Nicht-Probabilistische Retrievalmodelle

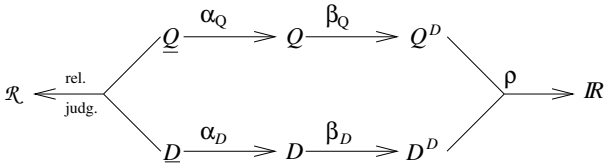
Norbert Fuhr

Rahmenarchitektur für IR-Systeme



Notationen

Notationen



$\underline{q_k} \in \underline{Q}$:	Anfrage/Info-bed.	$\underline{d_m} \in \underline{D}$:	Dokument
$q_k \in Q$	Anfragerepräs.	$d_m \in D$	Dokumentrepräs.
$q_k^D \in Q^D$:	Anfragebeschr.	$d_m^D \in D^D$:	Dokumentbeschr.
$\mathcal{R}$:	Relevanzskala	ϱ:	Retrievalfunktion
		R	Retrievalwert

$T = \{t_1, \ldots, t_n\}$: Indexierungsvokabular
d_m^D: $\vec{d}_m = (d_{m_1}, \ldots, d_{m_n})$: Dokument-Beschreibung als Menge von Indexierungsgewichten

Überblick über die Modelle

- Boolesches Retrieval
- Fuzzy-Retrieval
- Vektorraummodell
- Probabilistisches (Relevanz-orientiertes) Retrieval
- (Statistisches) Sprachmodell

	Bool.	Fuzzy	Vektor	Prob.	Sprachmod..
theoretische Basis	Boolesche Logik	Fuzzy-Logik	Vektorraum-Modell	Wahrsch.-Theorie	Statist. Sprachmod.
Bezug zur Retrievalqual.		(x)		x	(x)
gewichtete Indexierung		x	x	(x)	x
gewichtete Frageterme		(x)	x	x	x
Fragestruktur:					
– linear			x	x	x
– boolesch	x	x	(x)	(x)	

Boolesches Retrieval

Boolesches Retrieval

Historisch als erstes Retrievalmodell entwickelt und eingesetzt (Dokument-Beschreibungen auf Magnetbändern!)

Dokumenten-Beschreibungen D^D:
ungewichtete Indexierung, d.h. $d_m^D = \vec{d}_m$ mit $d_{m_i} \epsilon \{0,1\}$ für $i = 1, \ldots, n$

boolesches Retrieval liefert nur Zweiteilung der Dokumente in „gefundene“ ($\varrho = 1$) und „nicht gefundene“ ($\varrho = 0$) Dokumente

Frage-Beschreibungen Q^D:

1. $t_i \epsilon T \Rightarrow t_i \epsilon Q^D$
2. $q_1, q_2 \epsilon Q^D \Rightarrow q_1 \wedge q_2 \epsilon Q^D$
3. $q_1, q_2 \epsilon Q^D \Rightarrow q_1 \vee q_2 \epsilon Q^D$
4. $q \epsilon Q^D \Rightarrow \neg q \epsilon Q^D$

Retrievalfunktion $\varrho(q, d_m)$:

1. $t_1 \epsilon T \Rightarrow \varrho(t_i, \vec{d}_m) = d_{m_i}$
2. $\varrho(q_1 \wedge q_2, \vec{d}_m) = \min(\varrho(q_1, \vec{d}_m), \varrho(q_2, \vec{d}_m))$
3. $\varrho(q_1 \vee q_2, \vec{d}_m) = \max(\varrho(q_1, \vec{d}_m), \varrho(q_2, \vec{d}_m))$
4. $\varrho(\neg q, \vec{d}_m) = 1 - \varrho(q, \vec{d}_m)$

Mächtigkeit der booleschen Anfragesprache:

jede beliebige Dokumentenmenge kann selektiert werden

(Voraussetzung: alle Dokumente besitzen unterschiedliche Beschreibungen)

Konstruktion der booleschen Frageformulierung q zu einer vorgegebenen Dokumentenmenge D:

$$
\begin{aligned}
q_m &= x_{m_1} \wedge \ldots \wedge x_{m_n} \text{ mit} \\
x_{m_i} &= \begin{cases} t_i & \text{falls } d_{m_i} = 1 \\ \neg t_i & \text{sonst} \end{cases} \\
q &= \bigvee_{d_j \epsilon D} q_j
\end{aligned}
$$

Beispiel-Recherche

“The side effects of drugs on memory or cognitive abilities, not related to aging”

1.	19248	DRUGS
2.	2412	DRUGS in TI
3.	2560	AGING
4.	19119	DRUG not AGING
5.	2349	#2 and #4
6.	9305	MEMORY
7.	6	#5 and (DRUG near4 MEMORY)
8.	22091	COGNITIVE
9.	16	#5 and (DRUG near4 COGNITIVE)
10.	22	#7 or #9
11.	2023	SIDE-EFFECTS-DRUG in DE
12.	0	#11 and #10

Nachteile des booleschen Retrieval

1. Größe der Antwortmenge ist schwierig zu kontrollieren
2. Keine Ordung der Antwortmenge nach mehr oder weniger relevanten Dokumenten
3. Keine Möglichkeit zur Gewichtung von Fragetermen oder gewichteter Indexierung
4. Trennung gefunden / nicht gefunden zu streng:
   Zu $q = t_1 \wedge t_2 \wedge t_3$ werden Dokumente mit zwei gefundenen Termen genauso zurückgewiesen wie solche mit 0
   Analog für $q = t_1 \vee t_2 \vee t_3$ keine Unterteilung der gefundenen Dokumente
5. Erstellung der Frageformulierung sehr umständlich
6. schlechte Retrievalqualität

Trotzdem weiterhin Einsatz bei

- Patentretrieval (professionelle Rechercheure)
- Rechtsstreitigkeiten (Spezifik. offenzulegender Dokumente)

Fuzzy-Retrieval

Fuzzy-Retrieval

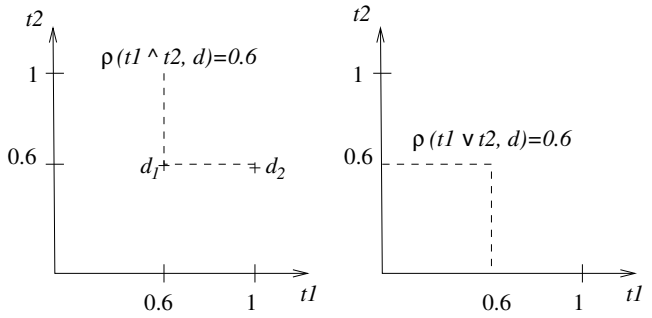
Teilweise Überwindung der Nachteile des booleschen Retrieval

Dokumenten-Beschreibungen:
Erweiterung auf gewichtete Indexierung, d.h. $d_{m_i} \epsilon [0,1]$

Frage-Beschreibungen, Retrievalfunktion:
wie beim booleschen Retrieval

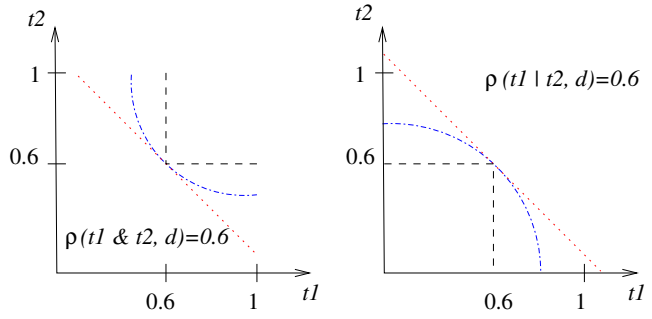
Retrievalfunktion liefert jetzt Werte $\varrho(q_k^D, \vec{d}_m) \epsilon [0,1]$
$\rightarrow$ Ranking der Antwortmenge

Problematische Definition der Retrievalfunktion



$$T = \{t_1, t_2\}$$
$$q = t_1 \wedge t_2$$
$$\vec{d}_1 = (0.6, 0.6) \quad , \quad \vec{d}_2 = (0.99, 0.59)$$
$$\varrho(q, \vec{d}_1) = 0.6 \quad , \quad \varrho(q, \vec{d}_2) = 0.59$$

Andere Definitionen der Fuzzy-Operatoren



überwinden Nachteile der Standard-Definition,

aber verletzen Gesetze der Booleschen Algebra:
(z.B. $\varrho(((t_1 \vee t_2) \wedge t_3), d) \neq \varrho(((t_1 \wedge t_3) \vee (t_2 \wedge t_3)), d))$

Kollektion	MEDLARS	ISI	INSPEC	CACM
#Dok.	1033	1460	12684	3204
#Fragen	30	35	77	52
Bool.	0.2065	–	0.1159	–
Fuzzy	0.2368	0.1000	0.1314	0.1551
Vektor	0.5473	0.1569	0.2325	0.3027

Experimenteller Vergleich von Booleschem Retrieval, Fuzzy-Retrieval und Vektorraummodel

Beurteilung des Fuzzy-Retrieval

+ Generalisierung des booleschen Retrieval für gewichtete Indexierung → Ranking
– keine Fragetermgewichtung
– schlechte Retrievalqualität
– Erstellung der Frageformulierung sehr umständlich

Das Vektorraummodell

Definition
Retrievalfunktion
Coordination Level Match
Dokumenten-Indexierung
Relevance Feedback

Das Vektorraummodell

Definition

zuerst entstanden im Rahmen der Arbeiten zu SMART (experimentelles Retrievalsystem von G. Salton und Mitarbeitern (Harvard/Cornell), seit 1961)

Dokumente und Fragen als Punkte in einem orthonormalen Vektorraum, der durch die Terme aufgespannt wird

orthonormaler Vektorraum:

- alle Term-Vektoren orthogonal (und damit auch linear unabhängig)
- alle Term-Vektoren normiert

Dokument-Beschreibung: ähnlich wie Fuzzy-Retrieval
$d_m^D = \vec{d}_m$ mit $d_{m_i} \epsilon \boldsymbol{R}$ für $i = 1, \ldots, n$

Frage-Beschreibung:
$q_k^Q = \vec{q}_k$ mit $q_{k_i} \epsilon \boldsymbol{R}$ für $i = 1, \ldots, n$

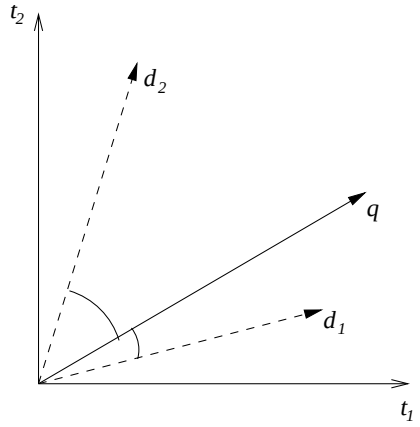
Retrievalfunktion

Vektor-Ähnlichkeitsmaße, z.B. Cosinus

$$\cos(\vec{q}_k \cdot \vec{d}_m) = \frac{\vec{q}_k \cdot \vec{d}_m}{|\vec{q}_k| \cdot |\vec{d}_m|}$$

Falls Vektoren normiert:
Betrachte nur Skalarprodukt

$$\begin{aligned} \varrho(\vec{q}_k, \vec{d}_m) &= \vec{q}_k \cdot \vec{d}_m \\ &= \sum_{t_i \in T} q_{k_i} \cdot d_{m_i} \end{aligned}$$



Beispiel-Frage:
"retrieval experiments with weighted indexing"

term	q_{k_i}	d_{1_i}	d_{2_i}	d_{3_i}	d_{4_i}
retrieval	1	0.33	0.33	0.25	0.25
experiment	1	0.33	0.33	0.25	0.25
weight	1				0.25
index	1			0.25	0.25
XML		0.33			
method			0.33		
binary				0.25	
RSV		0.66	0.66	0.75	1.00

Coordination Level Match

Vereinfachung des Vektorraummodells:
nur binäre Frage- und Dokumenttermgewichtung

Dokument-Beschreibung: wie Boolesches Retrieval
$d_m^D = \vec{d}_m$ mit $d_{m_i} \epsilon \{0, 1\}$ für $i = 1, \ldots, n$

Frage-Beschreibung:
$q_k^Q = \vec{q}_k$ mit $q_{k_i} \epsilon \{0, 1\}$ für $i = 1, \ldots, n$

Retrievalfunktion:
Skalarprodukt

$$\varrho(\vec{q}_k, \vec{d}_m) = \vec{q}_k \cdot \vec{d}_m = |q_k^T \cap d_m^T|$$

Dokumenten-Indexierung

Vektorraum-Modell liefert keine Aussagen darüber, wie die Dokumenten-Indexierung zu berechnen ist!

(Dokumenten-)Indexierung im Vektoraummodell:
heuristische Formeln zur Berechnung der Indexierungsgewichte

zugrundeliegende Dokumenten-Repräsentation: Multi-Menge (Bag) von Termen

Heuristiken:
Indexierungsgewicht umso höher, je ...

- häufiger der Term im Dokument
- seltener der Term in der Kollektion
- kürzer das Dokument

d_m^T	Menge der in d_m vorkommenden Terms
l_m	Dokumentlänge (# laufende Wörter in d_m)
al	durchschnittliche Dokumentlänge in $\underline{D}$
tf_{mi}:	Vorkommenshäufigkeit (Vkh) von t_i in d_m.
n_i:	# Dokumente, in denen t_i vorkommt.
N:	# Dokumente in der Kollektion

inverse Dokumenthäufigkeit (idf):

$$idf_i = \log \frac{N}{n_i}$$

normalisierte Vorkommenshäufigkeit:

$$ntf_i = \frac{tf_{mi}}{tf_{mi} + 0.5 + 1.5\frac{l_m}{al}}$$

Indexierungsgewicht tfidf:

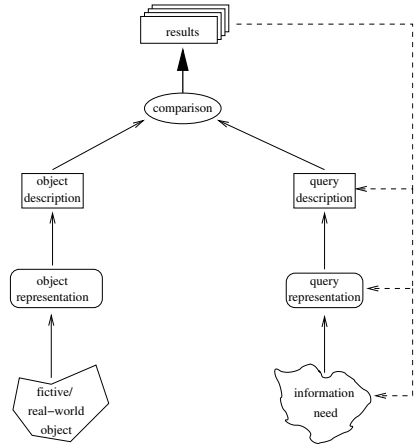
$$w_{mi} = ntf_i \cdot idf_i$$

Kollektion	CACM	CISI	CRAN	INSPEC	MED
Coord.	0.185	0.103	0.241	0.094	0.413
SMART	0.363	0.219	0.384	0.263	0.562

Binäre Gewichte (Coordination Level Match) vs.
SMART-Gewichtung von Fragen und Dokumenten
(aus Salton/Buckley 88)

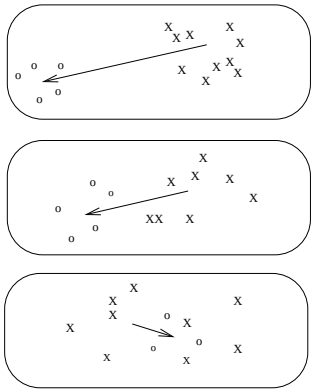
Relevance Feedback

iteratives Retrieval:



Relevance Feedback im VRM

Ziel: Modifikation des Fragevektors



Bestimmung des optimalen Fragevektors

D^R: relevante Dokumente
D^N: irrelevante Dokumente

Idee:
wähle Fragevektor $\vec{q}$ so, dass Differenz der RSVs zwischen relevanten und irrelevanten Dokumenten maximal wird:

$$\sum_{(d_k,d_l)\in D^R\times D^N} \vec{q}\vec{d}_k - \vec{q}\vec{d}_l \stackrel{!}{=} \max$$

mit der Nebenbedingung $\sum_{i=1}^{n} q_i^2 = c$

Extremwertproblem mit Randbedingung
$\rightarrow$ Lagrange-Multiplikator einsetzen

$$\begin{aligned}
F &= \lambda\left(\sum_{i=1}^{n} q_i^2 - c\right) + \sum_{(d_k,d_l)\in D^R\times D^N} \sum_{i=1}^{n} q_i d_{k_i} - q_i d_{l_i} \\
\frac{\partial F}{\partial q_i} &= 2\lambda q_i + \sum_{(d_k,d_l)\in D^R\times D^N} d_{k_i} - d_{l_i} \stackrel{!}{=} 0 \\
q_i &= -\frac{1}{2\lambda} \sum_{(d_k,d_l)\in D^R\times D^N} d_{k_i} - d_{l_i} \\
\vec{q} &= -\frac{1}{2\lambda} \sum_{(d_k,d_l)\in D^R\times D^N} \vec{d}_k - \vec{d}_l \\
&= -\frac{1}{2\lambda}\left(|D^N| \sum_{d_k\in D^R} \vec{d}_k - |D^R| \sum_{d_l\in D^N} \vec{d}_l\right) \\
&= -\frac{|D^N||D^R|}{2\lambda}\left(\frac{1}{|D^R|} \sum_{d_k\in D^R} \vec{d}_k - \frac{1}{|D^N|} \sum_{d_l\in D^N} \vec{d}_l\right)
\end{aligned}$$

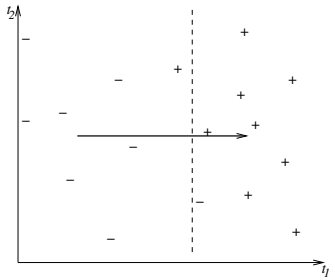
Optimaler Fragevektor

$$\vec{q} = -\frac{|D^N||D^R|}{2\lambda}\left(\frac{1}{|D^R|} \sum_{d_k\in D^R} \vec{d}_k - \frac{1}{|D^N|} \sum_{d_l\in D^N} \vec{d}_l\right)$$

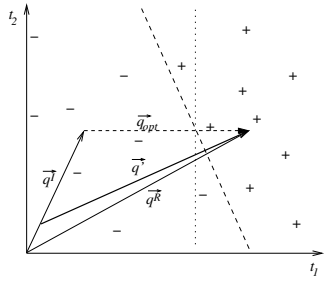
wähle c so, dass $|D^N||D^R|/2\lambda = -1$:

$$\vec{q} = \frac{1}{|D^R|} \sum_{d_k\in D^R} \vec{d}_k - \frac{1}{|D^N|} \sum_{d_l\in D^N} \vec{d}_l$$

$\hat{=}$ Verbindungsvektor der Zentroiden der relevanten / irrelevanten Dokumente



unterschiedliche Gewichtung positiver und negativer Beispiele:



Rocchio-Algorithmus

- unterschiedliche Gewichtung positiver und negativer Beispiele
- Berücksichtigung der ursprünglichen Anfrage

$$\vec{q}_k{}' = \vec{q}_k + \alpha \frac{1}{|D_k^R|} \sum_{d_j \epsilon D_k^R} \vec{d}_j - \beta \frac{1}{|D_k^N|} \sum_{d_j \epsilon D_k^N} \vec{d}_j$$

α, β — positive Konstanten, heuristisch festzulegen (z.B. $\alpha = 0.75$, $\beta = 0.25$)

Vorgehensweise:

1. Retrieval mit Fragevektor $\vec{q}_k$ vom Benutzer
2. Relevanzbeurteilung der obersten Dokumente der Rangordnung
3. Berechnung eines verbesserten Fragevektors $\vec{q}_k{}'$ aufgrund der Feedback-Daten
4. Retrieval mit dem verbesserten Vektor
5. Evtl. Wiederholung der Schritte 2-4

Beurteilung des Vektorraummodells

+ einfaches Modell, insbes. für den Benutzer
+ unmittelbar anwendbar auf neue Kollektionen
+ gute Retrievalqualität
– sehr viele heuristische Komponenten
– kein Bezug zur Retrievalqualität
(Optimalität von Relevance Feedback?)
– Dokumentrepräsentation kann schlecht erweitert werden