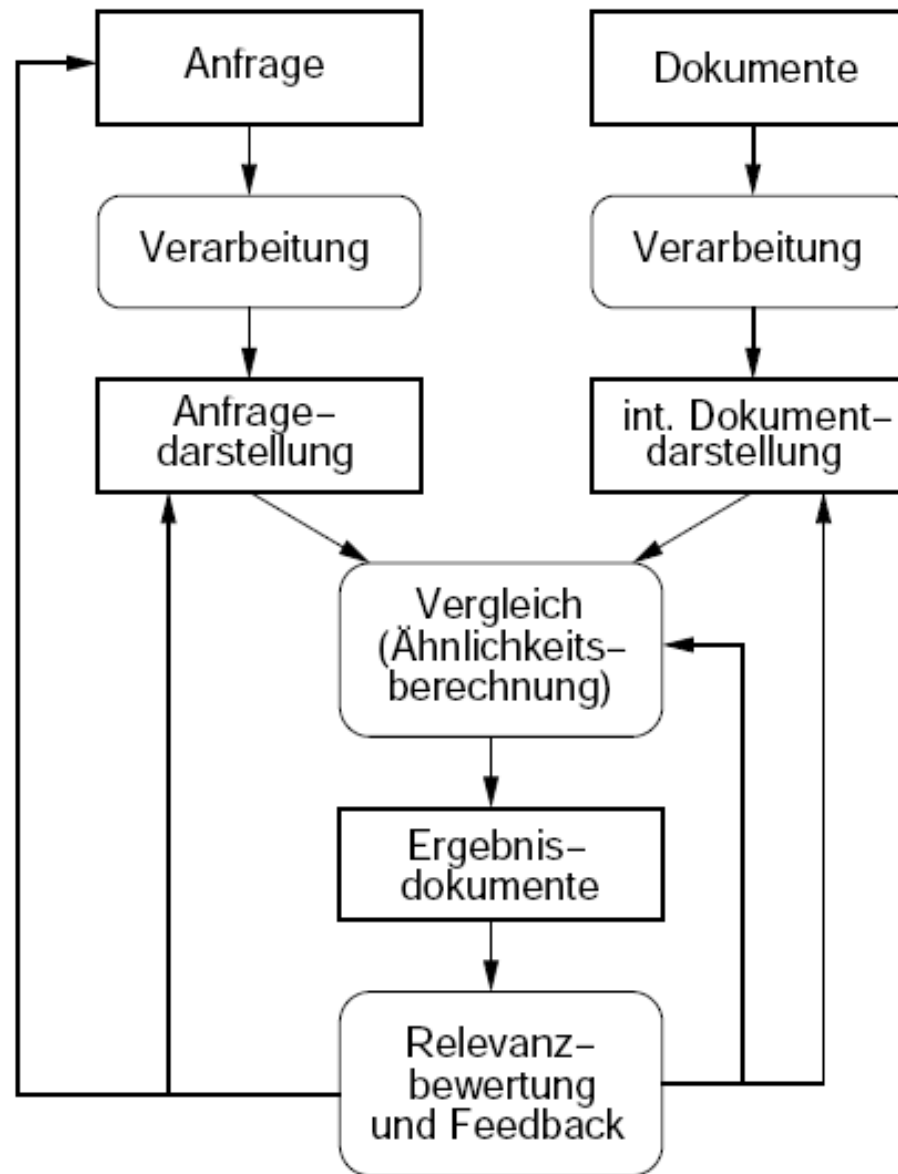


3.3 Relevance Feedback

Bewertung von Ergebnisdokumenten

Auswertung von Bewertungen



erstes Ergebnis oft nicht zufriedenstellend

Gründe:

vage Vorstellung des Nutzers über Suchergebnisse

schlechte Anfrageformulierung

unbekannte Dokumentenkollektion

keine relevanten Dokumente verfügbar

Reaktion auf nicht zufriedenstellendes Ergebnis <is web>

Abbruch

Browsing

manuelle Abfragemodifikation

Relevance Feedback

Anwender braucht keine Anfragemodifikation durchzuführen
→ muss Systeminterna nicht kennen

Anwender **bewertet nur Ergebnisse**

Annähern an Ergebnis iterativ oft einfacher als mit einer einzigen Anfrage

Relevanzwerte vom Nutzer zum System

→ **Relevance Feedback**

Anfrage	Ergebnisdokumente			
	1	2	3	...
q	d_0	d_1	d_2	...
q_0	d_4	$d_1 (+)$	$d_5 (-)$...
q_1	$d_1 (+)$	$d_3 (+)$	$d_4 (-)$...
q_2	d_3	d_1	d_0	...

q korrekte aber unbekannte Anfrage

q_0 initiale Anfrage

q_1 Anfrage nach 1. Iteration

q_2 Anfrage nach 2. Iteration

zu berücksichtigende Aspekte:

Anzahl der zu bewertenden Dokumente soviel wie möglich
versus Aufwand: ≤ 10

reduzierte Darstellung der Ergebnisdokumente
z.B. Thumbnails, Zusammenfassungen

Art der Bewertung

- ◆ relevant und keine Bewertung
- ◆ relevant, irrelevant und keine Bewertung
- ◆ gestufte Relevanzwerte

Bewertungsgranulat:

Dokument oder Dokumenteneigenschaften

- ◆ bzgl. mehreren Anfrageobjekten (Suchtermen) einer zusammengesetzten Anfrage, z.B. „*Korsika **and** Strand*“
- ◆ bzgl. verschiedener Eigenschaftswerte z.B.: Bildersuche anhand Form und Größe, aber getrennte Bewertung

sehr irrelevant	irrelevant	neutral	relevant	sehr relevant
-3	-1	0	1	3

automatische Bewertung, also Entlastung des Nutzers
erste relevante Dokumente gelten automatisch als relevant

→ Anfragemodifikation

schlechtere Ergebnisse als bei manueller Relevanzbewertung
oft bessere Ergebnisse als
ohne Anfrage-Iteration

Finden eines besseren Ergebnisses

Anfragemodifikation:

- ◆ Modifikation von Nutzerprofilen
z.B. Text-Retrieval:
Profil enthält nutzerspezifische Suchterme (etwa „Ferienwohnung“ für Reisebüro)
- ◆ Modifikation der Dokumentenbeschreibungen
z.B. Modifikation des Indexvokabulars
→ erfordert hohen Aufwand
- ◆ Modifikation des Suchalgorithmus
z.B. andere Distanzfunktion
- ◆ Modifikation von Anfragetermgewichten in zusammengesetzten, gewichteten Anfragen

Anfragemodifikation im Vektorraummodell: Rocchio

Verschiebung des Anfragevektors

- ◆ in Richtung der als relevant bewerteten Dokumente
- ◆ weg von als irrelevant bewerteten Dokumenten

Anfrage:

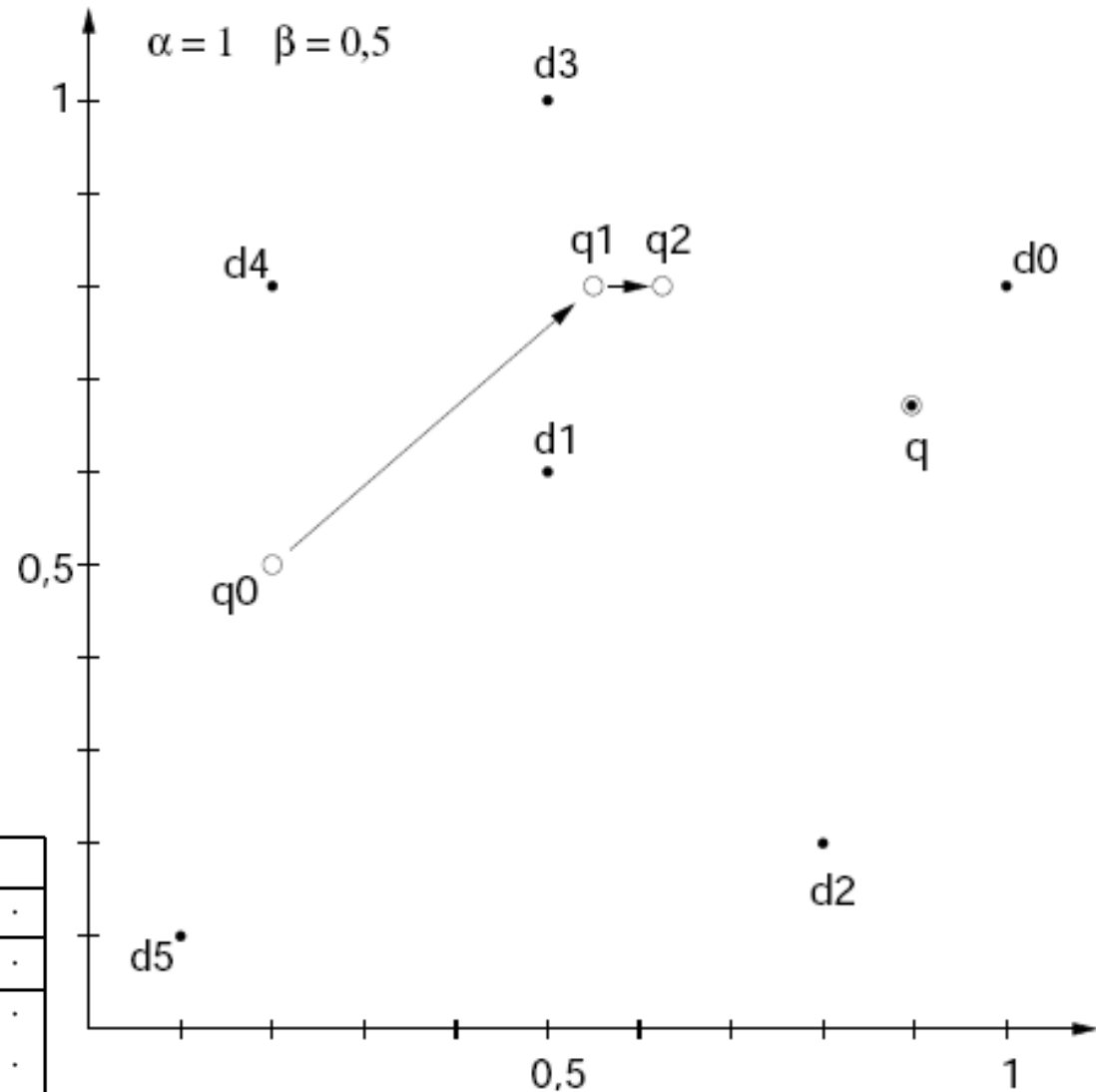
q_{alt}

relevant (irrelevant) bewertete Dokumente: $D_r (D_i)$

Anfragemodifikation im Vektorraummodell: Rocchio (2)

α und β wichten Einfluss relevanter und irrelevanter Dokumente

$$q_{neu} = q_{alt} + \frac{\alpha}{|D_r|} \sum_{d_r \in D_r} (d_r - q_{alt}) - \frac{\beta}{|D_i|} \sum_{d_i \in D_i} (d_i - q_{alt})$$



Anfrage	Ergebnisdokumente			
	1	2	3	...
q	d_0	d_1	d_2	...
q_0	d_4	$d_1 (+)$	$d_5 (-)$...
q_1	$d_1 (+)$	$d_3 (+)$	$d_4 (-)$...
q_2	d_3	d_1	d_0	...

Einführung

Precision, Recall und Fallout

Kombinierte Precision- und Recall-Werte

Precision- und Recall-Werte abhängig von der
Ergebnisgröße

Bewertung (Qualitätsvergleich) verschiedener Retrieval-Systeme

- ◆ quantitative Maße vonnöten

Ergebnisbewertung bzgl. versch. Formulierungsebenen

inwieweit ist Ergebnis relevant zu

- nicht formuliertem Informationsbedarf
- Informationsbedarf als natürlichsprachliche Frage
- Informationsbedarf als Anfrage

Ergebnisbewertung bzgl. versch. Formulierungsebenen (2)

Bewertungsunterschiede bzgl.

- ◆ Informationsbedarf und Frage:
 - ◆ mangelnde Fähigkeit des Nutzers, Informationsbedarf adäquat als Frage zu formulieren (Bewertung des Nutzers)
- ◆ Frage und Anfrage:
 - ◆ mangelnde Formulierung einer Frage als Anfrage (Bewertung des Nutzers und der Abfragesprache)

Relevanz: inwieweit befriedigt Ergebnis den Informationsbedarf

Nützlichkeit: inwieweit ist das Ergebnis hilfreich

Unabhängigkeit zwischen Nützlichkeit und Relevanz

Beispiel: irrelevant aber (vielleicht?) nützlich:

Suche nach Urlaubsmöglichkeiten in Korsika liefert Flugmöglichkeiten

Beispiel: relevant aber nutzlos:

Suche nach Urlaubsmöglichkeiten in Korsika liefert veraltete Dokumente zu Ferienwohnungen

im Folgenden Konzentration auf Relevanz und Irrelevanz von Ergebnissen bzgl. Anfrage

zunächst Ergebnis als Menge

zwei verschiedene Fehlentscheidungen

- 1. false alarms (fa) bezeichnet diejenigen Dokumente, die vom Retrieval-System irrtümlicherweise als relevant zurückgeliefert werden**
- 2. false dismissals (fd) sind Dokumente, die fälschlicherweise vom Retrieval-System als irrelevant eingestuft wurden**

zwei korrekte Entscheidungen:

1. correct alarms (ca)

2. correct dismissals (cd)

fa, fd, ca, cd stehen für entsprechende Dokumentanzahlen
bzgl. einer Anfrage

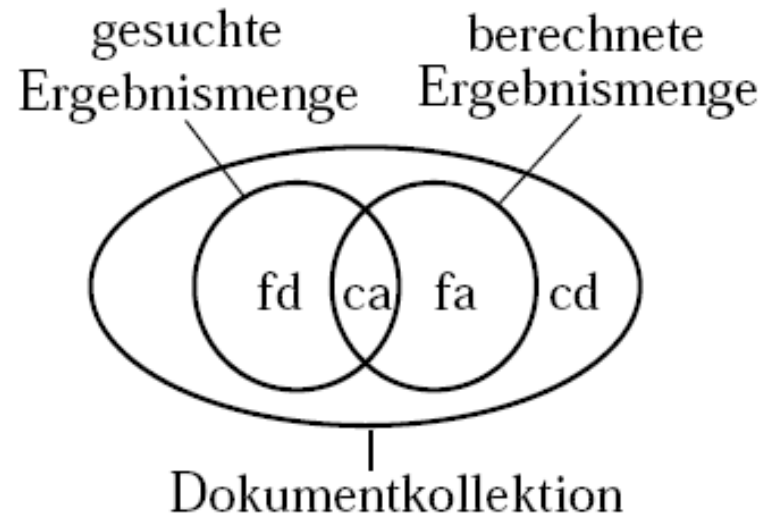
Nutzer- bewertung	Systembewertung	
	relevant	irrelevant
relevant	ca	fd
irrelevant	fa	cd

fa, fd, ca, cd im Euler-Venn-Diagramm

$$|\text{gesuchte Ergebnismenge}| = fd + ca$$

$$|\text{berechnete Ergebnismenge}| = ca + fa$$

$$|\text{Dokumentkollection}| = fd + ca + fa + cd$$



Precision: Wieviele (als Verhältnis) Ergebnisdokumente sind tatsächlich relevant?

$$P = \frac{ca}{ca + fa} \quad P \in [0, 1]$$

Recall: Wieviele (als Verhältnis) tatsächlich relevante Dokumente erscheinen im Ergebnis

$$R = \frac{ca}{ca + fd} \quad R \in [0, 1]$$

Verhältnis falsch gefundener zur Gesamtzahl irrelevanter Dokumente

$$F_q = \frac{fa}{fa + cd} \quad F \in [0, 1]$$

normalerweise nur $fa+ca+fd+cd$ und $ca+fa$ bekannt
für fd und ca Nutzerbewertung notwendig
 ca durch Relevanzbewertung der Ergebnisdokumente
 fd durch Relevanzbewertung der zurückgehaltenen
Dokumente

Problem: meist hoher Aufwand aufgrund hoher Dokumentenanzahl, insbesondere für fd

Lösungsansatz: Verwendung von kleiner, repräsentativer Testdatenbank

Precision, Recall, Fallout definiert bzgl. einer Anfrage
besser: mehrere Anfragen und entsprechende
Durchschnittswerte

20 Dokumente, 2 Anfragen, jeweils 10 Ergebnisdokumente

Anfrage	fa	ca	fd	cd	Precision	Recall	Fallout
<i>q</i> ₁	8	2	6	4	20%	25%	66%
<i>q</i> ₂	2	8	2	8	80%	80%	20%
Durchschnitt	–	–	–	–	50%	52,5%	43%

beide Werte gleich wichtig, da

gute Precision auf Kosten Recall:

möglichst kleine Ergebnismenge z.B. Ergebnismenge enthält nur ein relevantes Dokument:

$$P = 100\% \text{ und } R \rightarrow 0$$

gutes Recall auf Kosten Precision: möglichst große Ergebnismenge z.B. Ergebnismenge enthält alle Dokumente:

$$R = 100\% \text{ und } P \rightarrow 0$$

Ergebnis als Liste

*ca, cd, fa, fd, Precision und Recall sind abhängig von
Ergebnisgröße r*

*daher $ca(r)$, $cd(r)$, $fa(r)$, $fd(r)$, $P_q(r)$ und $R_q(r)$ bzgl. der r
ersten Ergebnisdokumente*

beim Inkrementieren von r , *pro neues Dokument* zwei Varianten

- relevantes Dokument: $P_q(r + 1) > P_q(r)$ und $R_q(r + 1) > R_q(r)$

$$P_q(r + 1) = \frac{ca(r) + 1}{ca(r) + fa(r) + 1} \quad R_q(r + 1) = \frac{ca(r) + 1}{ca(r) + fd(r)}$$

- irrelevantes Dokument: $P_q(r + 1) < P_q(r)$

$$P_q(r + 1) = \frac{ca(r)}{ca(r) + fa(r) + 1} \quad R_q(r + 1) = R_q(r)$$

20 Dokumente, 1 Anfrage, 2 Retrieval-Systeme

die selben 20 Dokumente, neue Anfrage, 3. Retrieval-System

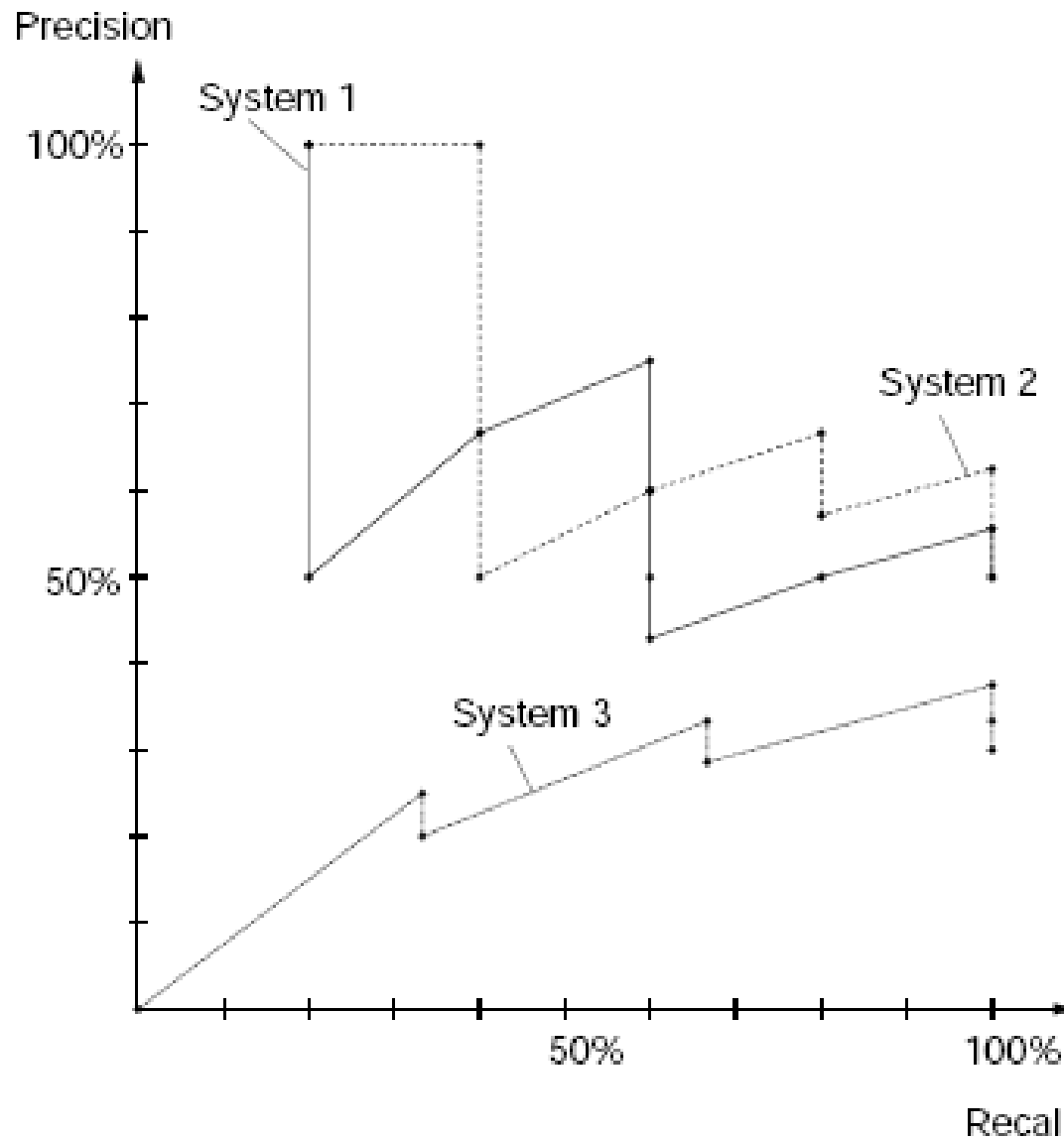
- tatsächlich relevante Dokumente: $\{d_2, d_3, d_5, d_8, d_{13}\}$
- Ergebnisliste 1. Retrieval-System: $\langle d_3, d_7, d_5, d_{13}, d_1, d_9, d_{12}, d_8, d_2, d_{16} \rangle$
- Ergebnisliste 2. Retrieval-System: $\langle d_{13}, d_2, d_7, d_6, d_8, d_5, d_1, d_3, d_{12}, d_{16} \rangle$

- tatsächlich relevante Dokumente: $\{d_1, d_3, d_8\}$
- Ergebnisliste 3. Retrieval-System: $\langle d_2, d_4, d_9, d_8, d_{10}, d_1, d_6, d_3, d_{11}, d_0 \rangle$

Beispielwerte für Precision und Recall tabellarisch

Anzahl	1	2	3	4	5	6	7	8	9	10
P_1	1/1	1/2	2/3	3/4	3/5	3/6	3/7	4/8	5/9	5/10
R_1	1/5	1/5	2/5	3/5	3/5	3/5	3/5	4/5	5/5	5/5
P_2	1/1	2/2	2/3	2/4	3/5	4/6	4/7	5/8	5/9	5/10
R_2	1/5	2/5	2/5	2/5	3/5	4/5	4/5	5/5	5/5	5/5
P_3	0/1	0/2	0/3	1/4	1/5	2/6	2/7	3/8	3/9	3/10
R_3	0/3	0/3	0/3	1/3	1/3	2/3	2/3	3/3	3/3	3/3

Beispielwerte für Precision und Recall graphisch **isweb**



Linie ist Sägezahnlinie

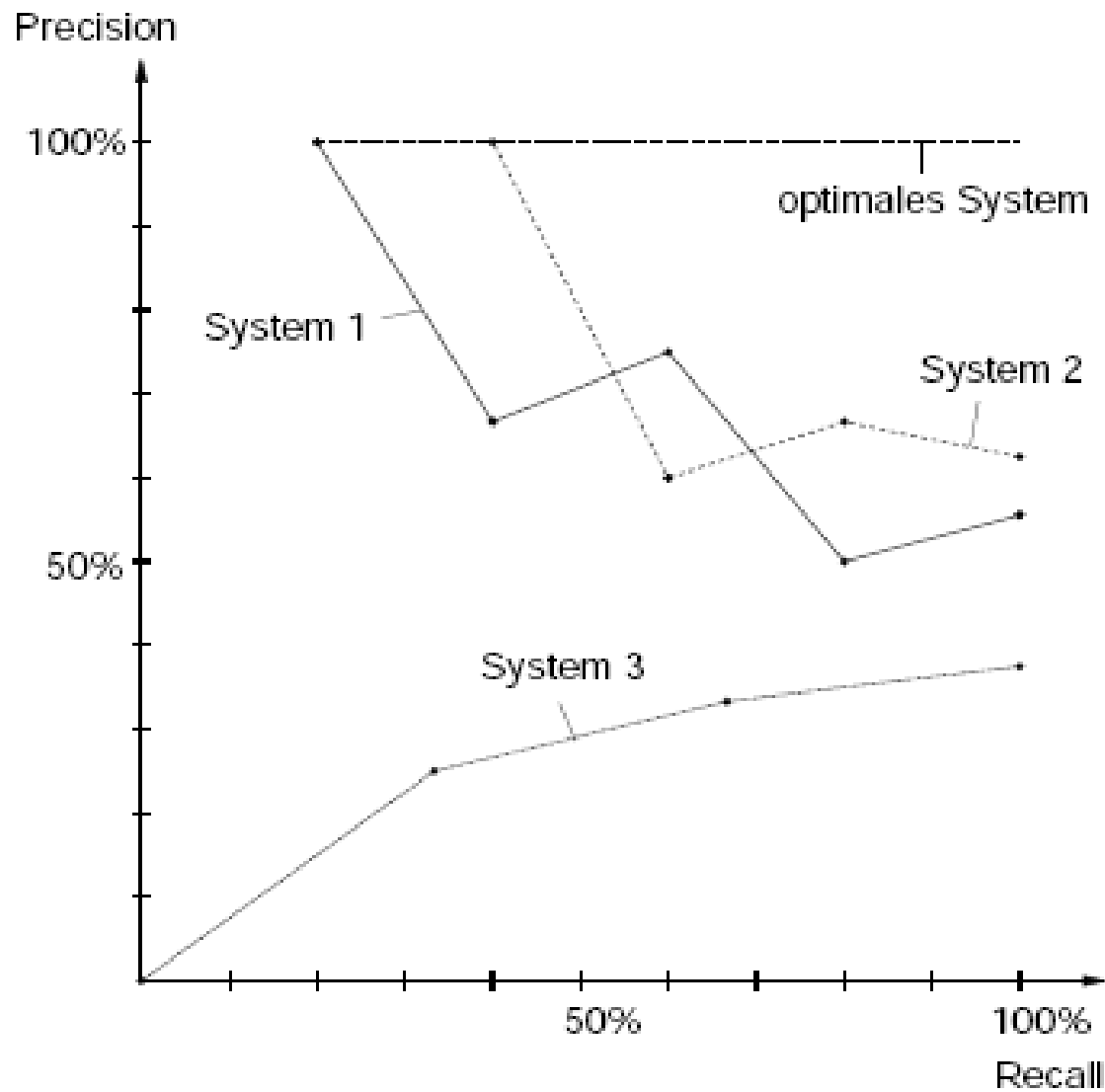
→ keine Funktion

Ziel: Precision als Funktion über Recall

Lösung: pro Recall-Wert maximalen Precision-Wert verwenden

Optimum: 100%-Linie

System besser, je näher zum Optimum



unterschiedliche Anfragen erzeugen oft unterschiedliche Recall-Werte

z.B. Anfrage 1,2 versus 3

Ziel: Vergleich Retrieval-System immer bzgl. 11 Standard-Recall-Stufen

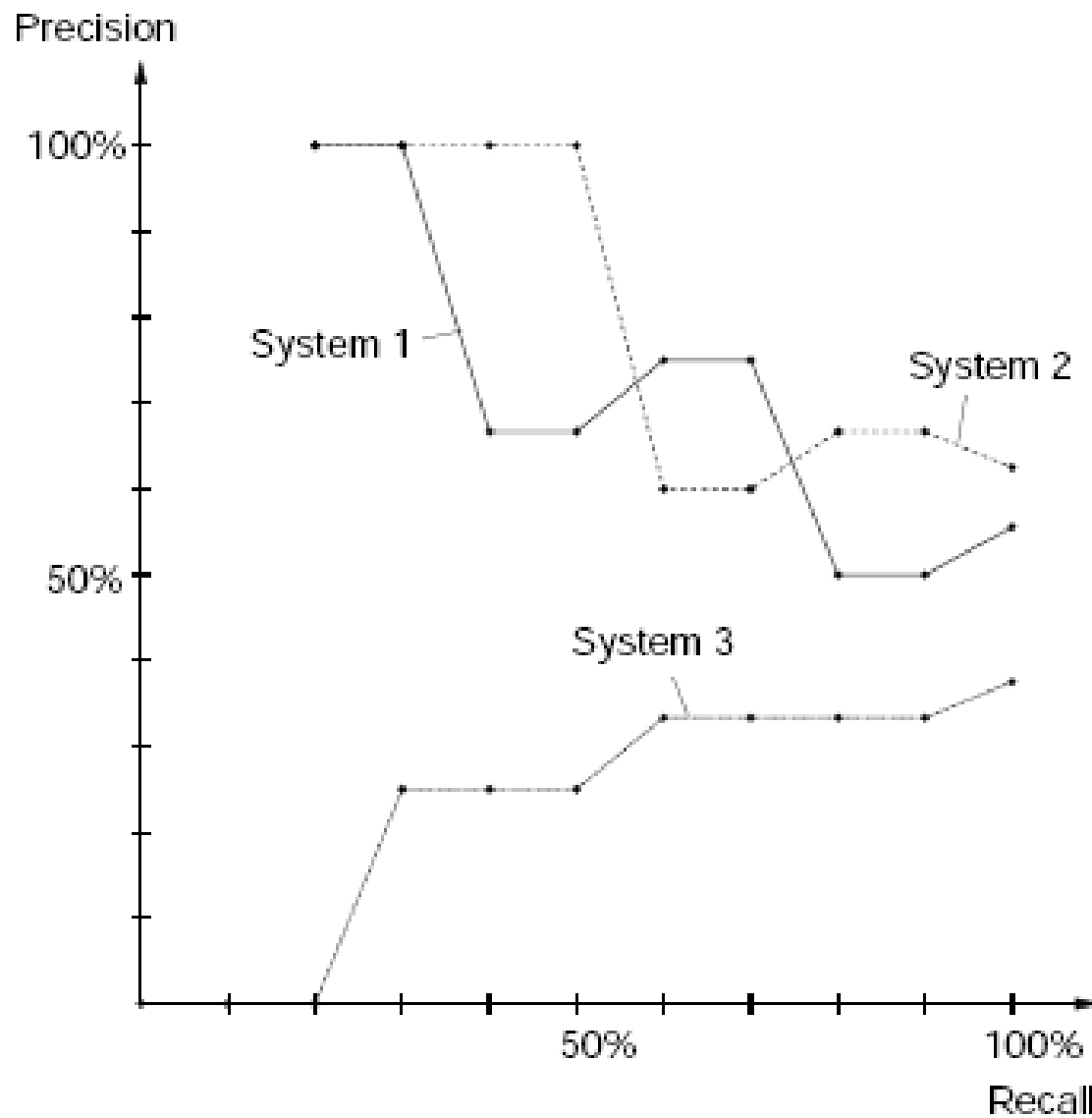
11 Standard-Recall-Stufen:

0%, 10%, ..., 100%

Lösung: Precision-Werte aus Intervallen berechnen:
 r_j^S ist j-te Stufe der 11 Stufen und r_i sind ursprüngliche Recall-Stufen

$$\mathcal{P}(r_j^S) = \begin{cases} \mathcal{P}(r_{j-1}^S) & : \forall r_i : r_i < r_j^S \vee r_i \geq r_{j+1}^S \\ \max\{P(r_i) \mid r_j^S \leq r_i < r_{j+1}^S\} & : \text{sonst} \end{cases}$$

11 Standard-Recall-Stufen grafisch

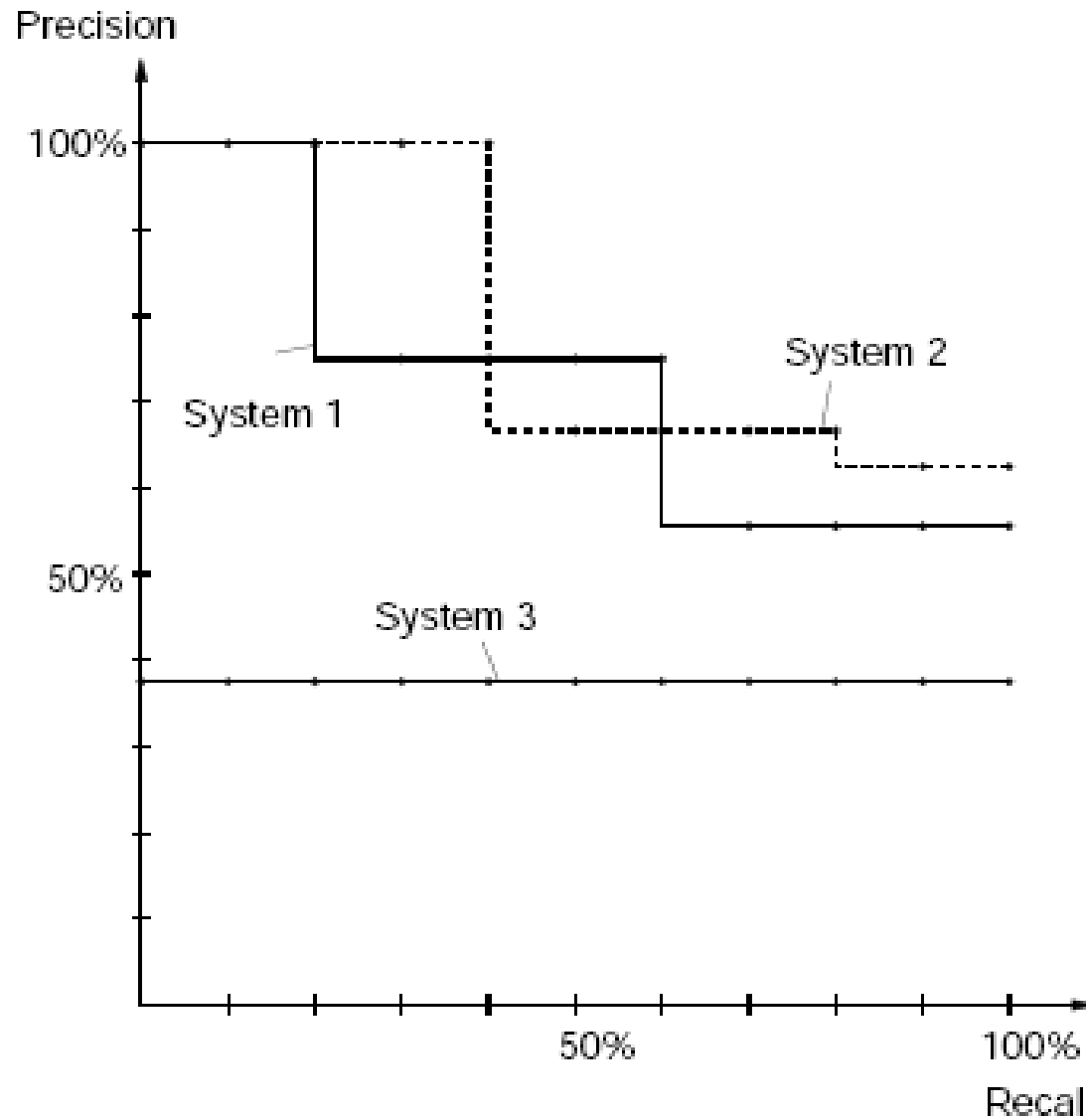


ursprüngliche Abbildung ist keine Funktion
(Sägezahnlinie)

alternative Lösung: pro Recall-Wert den größten
Precision-Wert des aktuellen und aller nachfolgenden
Recall-Werte übernehmen

$$\mathcal{P}(\textit{recall}) := \max\{P(r) \mid R(r) \geq \textit{recall}\}$$

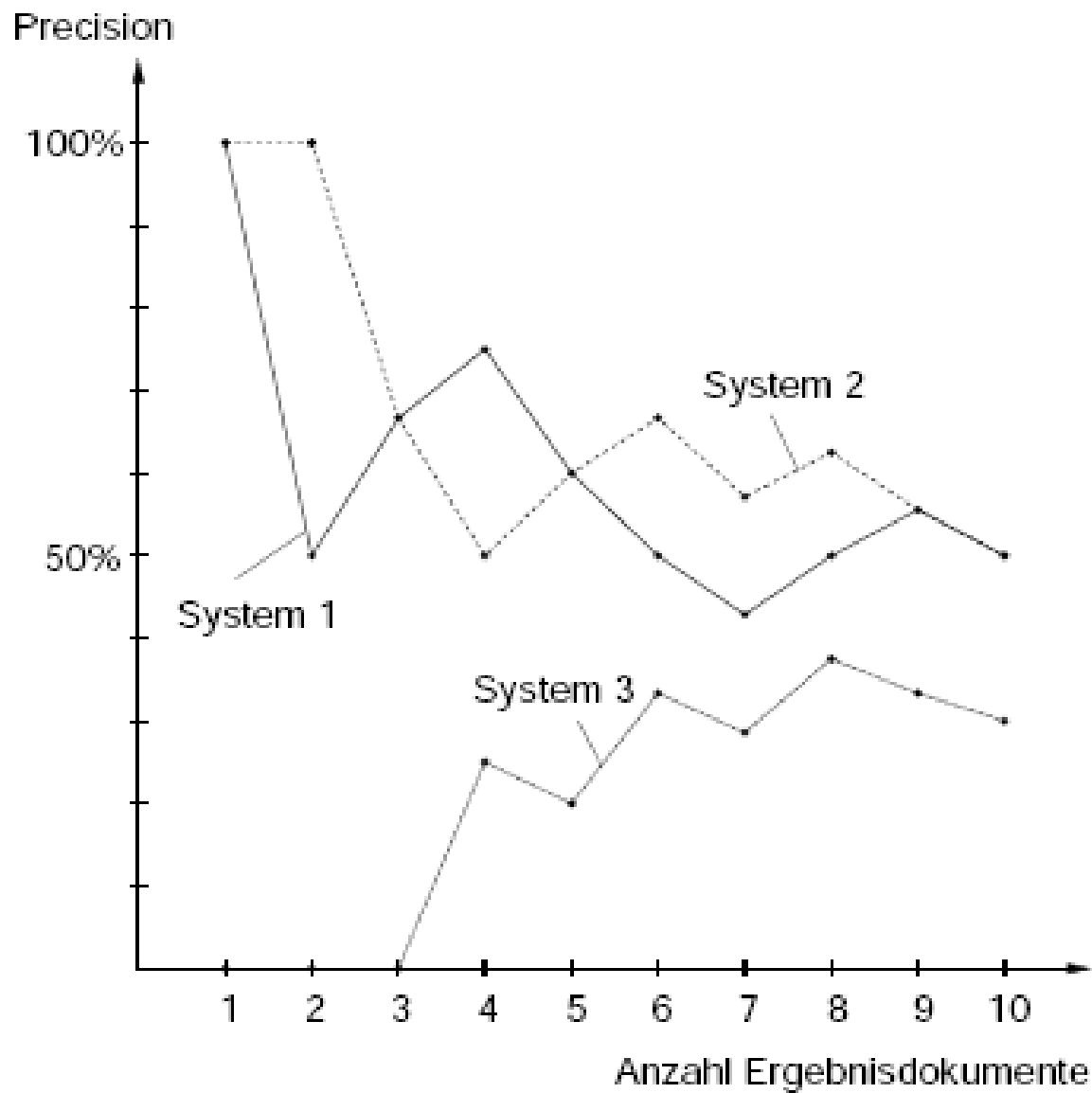
11 Standard-Recall-Stufen grafisch



Recall-unabhängiger Vergleich

Durchschnittswerte zum Vergleich nutzen

wenn gewünscht: zusätzlich Durchschnitt über mehrere
Anfragen

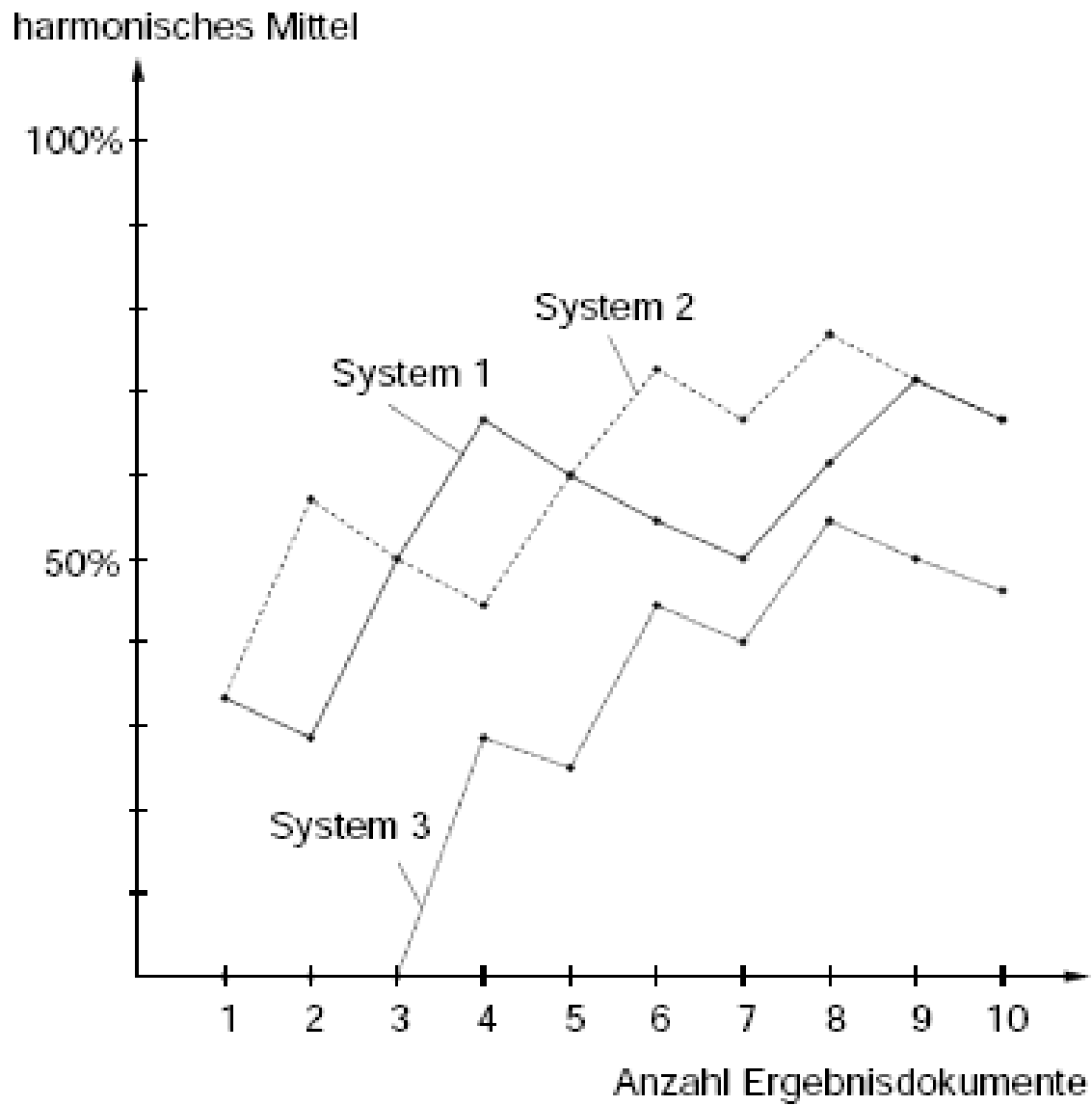


Kombination von Precision und Recall einer Ereignisgröße

$$HM(r) = \frac{2}{\frac{1}{R(r)} + \frac{1}{P(r)}}$$

Beispiel harmonisches Mittel

#	1	2	3	4	5	6	7	8	9	10
P_1	1/1	1/2	2/3	3/4	3/5	3/6	3/7	4/8	5/9	5/10
R_1	1/5	1/5	2/5	3/5	3/5	3/5	3/5	4/5	5/5	5/5
HM_1	2/6	2/7	4/8	6/9	6/10	6/11	6/12	8/13	10/14	10/15
P_2	1/1	2/2	2/3	2/4	3/5	4/6	4/7	5/8	5/9	5/10
R_2	1/5	2/5	2/5	2/5	3/5	4/5	4/5	5/5	5/5	5/5
HM_2	2/6	4/7	4/8	4/9	6/10	8/11	8/12	10/13	10/14	10/15
P_3	0/1	0/2	0/3	1/4	1/5	2/6	2/7	3/8	3/9	3/10
R_3	0/3	0/3	0/3	1/3	1/3	2/3	2/3	3/3	3/3	3/3
HM_3	0	0	0	2/7	2/8	4/9	4/10	6/11	6/12	6/13



Retrieval-Szenarien

Einfluss von Nutzerprofilen auf den Retrieval-Prozess

bis jetzt keine Unterscheidung von Anwender und Anwendergruppen

Verhalten bzw. Suchbedarf verschiedener Nutzer differiert oft

Idee: Subjektivität wird als Nutzerprofil modelliert und bei Suche berücksichtigt

nur sinnvoll bei häufigem Zugriff von einzelnen Nutzern mit Identifikation

Bibliothekar in einer Informatik-Bibliothek kennt Nutzer

- ◆ Professoren interessiert an spezieller Fachliteratur
- ◆ Studenten interessiert an Lehrbüchern

etwa eine feste Anfrage vorgegeben vom Nutzer oder aus Suchläufen ermittelt

zusätzlich möglich:

Bildungsstand: Vorkenntnisse

Vertrautheitsgrad mit Interessensgebiet: Anfänger versus Experte

Spachfähigkeit: Deutschkenntnisse

Retrieval-Historie:

Ergebnisse vorheriger Sitzungen

spezielle Präferenzen:

z.B. Bevorzugung eines bestimmten Journals

Szenario	Dokumentkolektion	Nutzerstamm
klassisches IR	statisch/dynamisch	dynamisch
Filtering mit Profilen	dynamisch	statisch
Retrieval mit Profilen	statisch/dynamisch	statisch

Vergleich Nutzerprofile mit neu eingefügten Dokumenten

Relevanz → Nutzer wird informiert

Dokument quasi als Anfrage und Profile als Datenkollektion

System ist aktiv → *Push-Dienst* oder *Current-Awareness-System* oder *Subscription*

zusätzliche Sortierung anhand Relevanz: *Routing*

Beispiel: Ärzte bleiben informiert anhand medizinischer Artikel,
die entsprechend ihren Profilen gefiltert werden

Nutzerstamm statisch

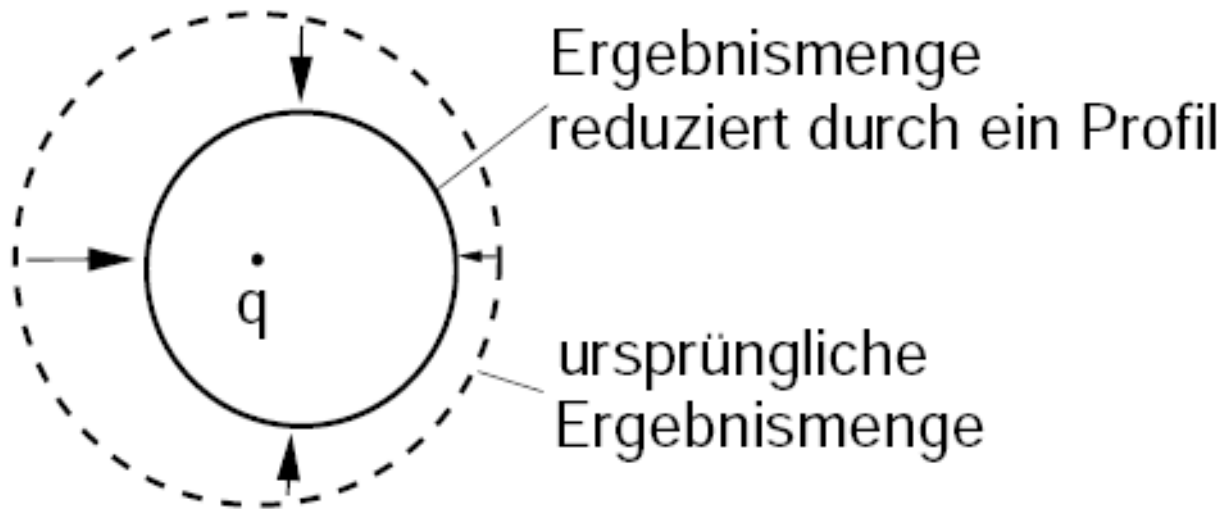
zwei Realisierungsmöglichkeiten

1. Nachfiltern:

- Filtering auf Anfrageergebnis
- hoher Berechnungsaufwand durch u.U. großem Zwischenergebnis
- reduzieren nur false alarms

2. Vorfiltern:

- Nutzerprofil beeinflusst Retrieval-Prozess direkt
- Reduzierung von false alarms und false dismissals

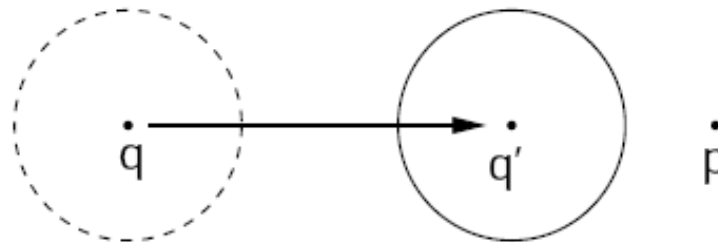


Einfluss von Nutzerprofilen auf den Retrieval-Prozess

Annahme: Anfrage als Profil

einfache Realisierung: Verschiebung Anfragepunkt q in
Richtung Profilabfragepunkt p

Einfluss von Nutzerprofilen auf den Retrieval-Prozess (2)



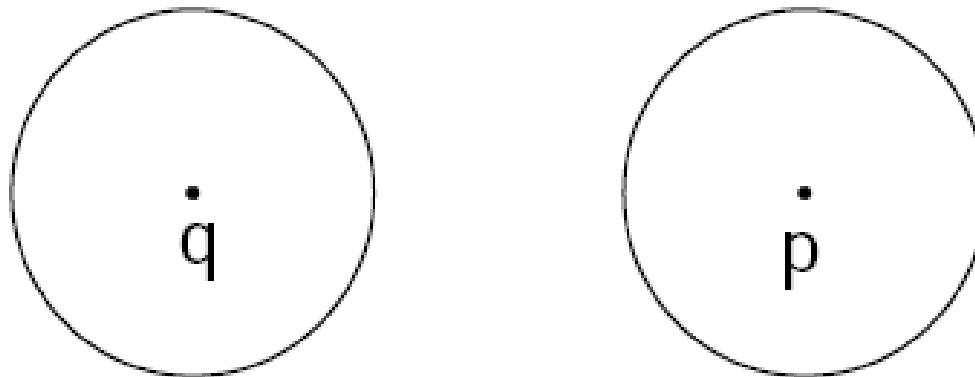
Problem: relevante Dokumente bzgl. q können irrelevant bzgl. q' werden, gewünscht: Reduzierung false dismissals statt Reduzierung false alarms

Minimumsfunktion

Erweiterung der Anfragesemantik statt -modifikation
naiver Ansatz: Min-Funktion

Minimumsfunktion

$$\text{Min}(\text{dissim}(d, q), \text{dissim}(d, p)) < r$$

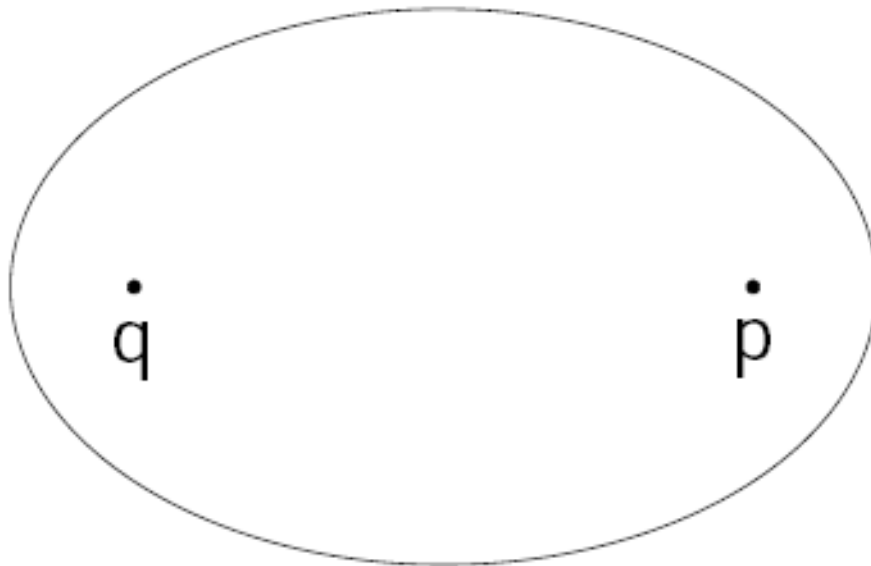


Problem 1: Profil liefert immer selbe Dokumentmenge

Problem 2: Dokumente zwischen *q* und *p* erscheinen nicht unbedingt im Ergebnis

Summenbildung

$$\text{dissim}(d, q) + \text{dissim}(d, p) < r$$



Problem: Suchbereich wird zu groß

Kompromiss zwischen Min-Funktion und Summenbildung

3 Varianten in Abhängigkeit

von r :

$$\text{dissim}(d, q) * \text{dissim}(d, p) < r$$

