Vergleich/Matching

- Die Schwierigkeiten bei der Verbalisierung/Codierung vager Informationsbedürfnisse,
- Die Vielfalt/fehlende Präzision des Ausdrucks in natürlicher Sprache,
- Die Subjektivität der Relevanz
- ….

→ machen ein «Exact Matching» unpraktikabel

! Wir suchen eine Möglichkeit, die Wahrscheinlichkeit der Relevanz eines Dokumentes hinsichtlich des Informationsbedürfnisses bestmöglich zu schätzen → «Best Match»

Rangierungsprinzipien

Erster Ansatz:

- Versuch „common sense“ Regeln aufzustellen
- Es gibt **KEINE** konkreten Systeme, die diese Rangierungsprinzipien so implementieren!, aber:
- Hilft bei späterer konkreter Umsetzung
- Hilft dem Benutzer als mentales Model

Rangierungsprinzip 1

- Je mehr Suchbegriffe (Terme) in einem Dokument vorkommen, desto wahrscheinlicher ist das Dokument relevant.

Anfrage:

Erdbeben Epizentrum Richter Skala

Dokument 1:

Erdbeben in China

Gestern ereignete sich in China ein Erdbeben mit *Epizentrum* 200km nördlich der Hauptstadt. Es hatte auf der *Richter Skala* die Stärke 5.7. Das Erdbeben dauerte 10 Sekunden. ...

Dokument 2:

Wirtschaftlicher Aufschwung

Nach dem *Erdbeben* 1990 wurden die Schäden zügig repariert, und die Stadt erlebt seither einen wirtschaftlichen Aufschwung.

Rangierungsprinzip 2

- Je häufiger ein Suchbegriff in einem Dokument vorkommt, desto wahrscheinlicher ist das Dokument relevant.

- ➔ Merkmalshäufigkeit (Anzahl Vorkommen eines Merkmals in einem Dokument)

Anfrage:

Erdbeben

Dokument 1:

Erdbeben in China

Gestern ereignete sich in China ein *Erdbeben* mit Epizentrum 200km nördlich der Hauptstadt. Es hatte auf der Richter Skala die Stärke 5.7. Das *Erdbeben* dauerte 10 Sekunden. ...

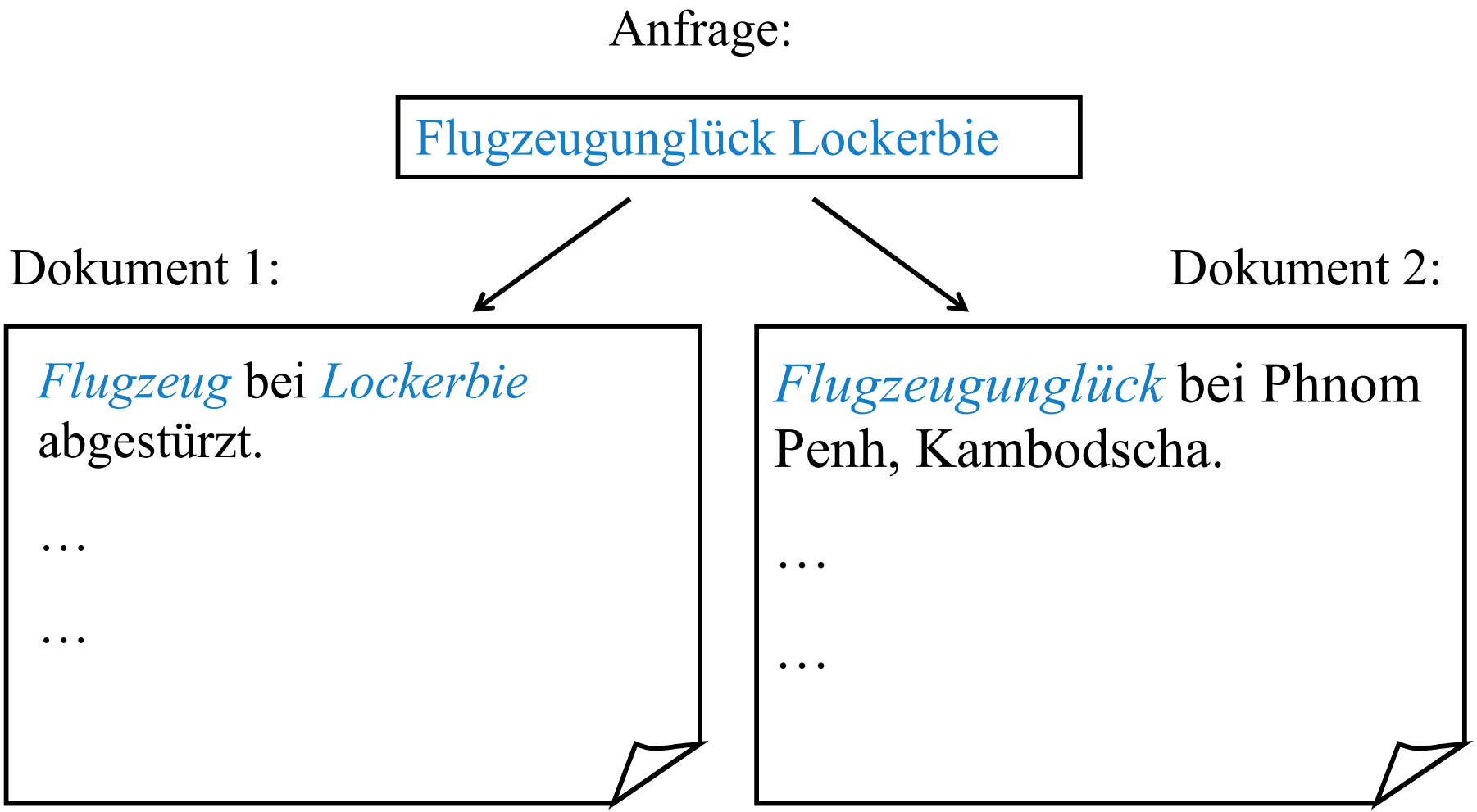
Dokument 2:

Wirtschaftlicher Aufschwung

Nach dem *Erdbeben* 1990 wurden die Schäden zügig repariert, und die Stadt erlebt seither einen wirtschaftlichen Aufschwung.

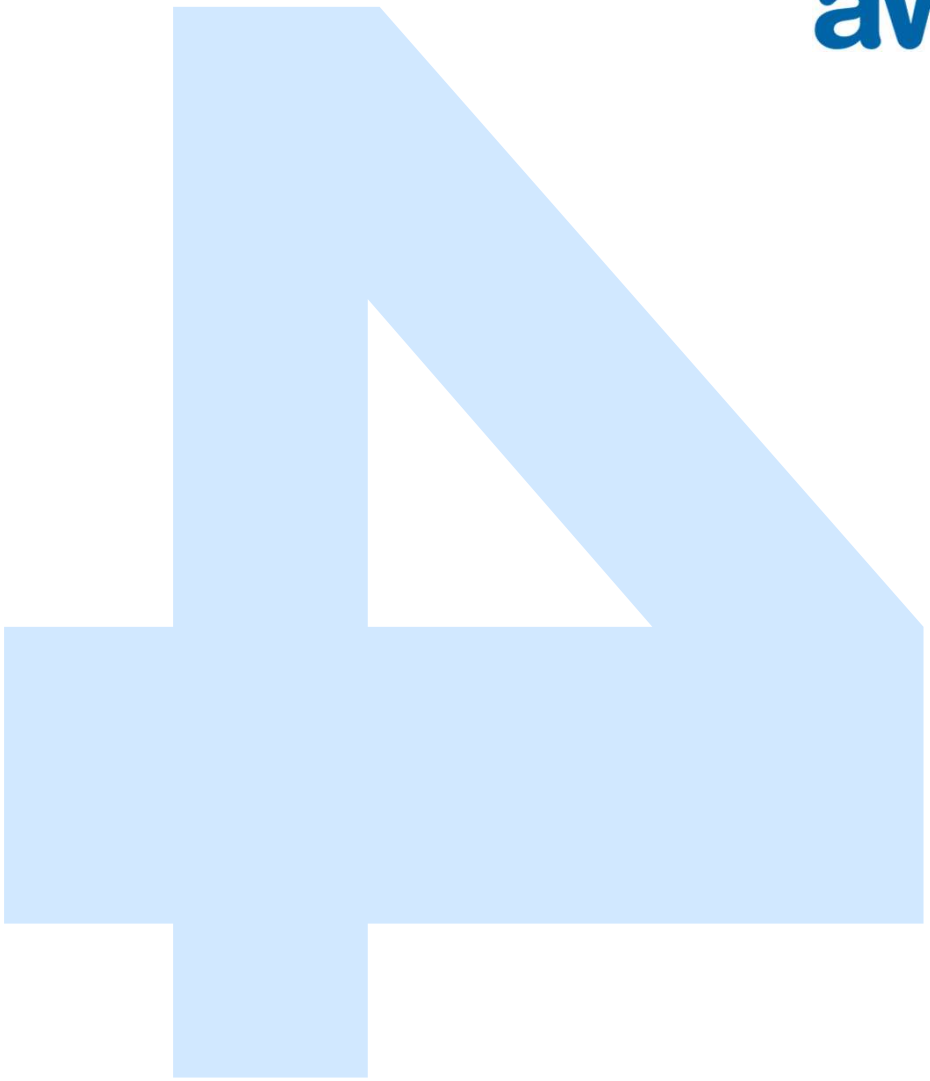
Rangierungsprinzip 3

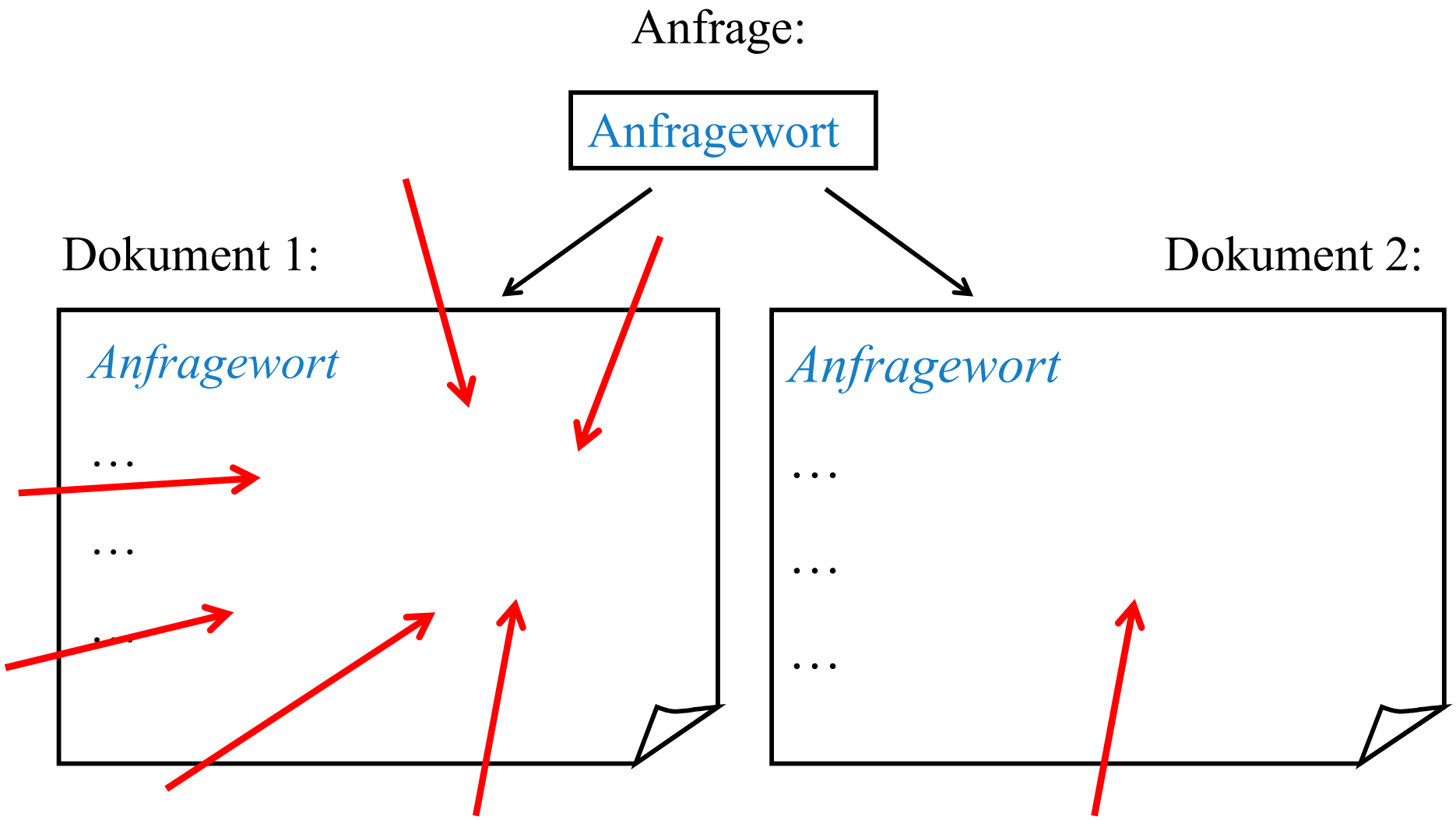
- Dokumente, die seltene Suchbegriffe enthalten, sind mit einer höheren Wahrscheinlichkeit relevant als Dokumente, die häufige Suchbegriffe enthalten.
- ➔ Dokumentenhäufigkeit (Anzahl Dokumente, die einen Begriff enthalten)


Anfrage:
Flugzeugunglück Lockerbie
Dokument 1:
Flugzeug bei Lockerbie abgestürzt.
…
…
Dokument 2:
Flugzeugunglück bei Phnom Penh, Kambodscha.
…
…

Rangierungsprinzip 4

- Je mehr Hyperlinks auf ein Dokument zeigen desto wahrscheinlicher ist es relevant.




Anfrage:
Anfragewort
Dokument 1:
Anfragewort
…
…
…
Dokument 2:
Anfragewort
…
…
…

Rangierungsprinzip 5

- Je näher die Suchbegriffe beieinander liegen, desto wahrscheinlicher ist das Dokument relevant.

Anfrage:

Hans Meyer

Dokument 1:

Hans Meyer neuer SNB-Präsident

Gestern wurde *Hans Meyer* zum neuen Präsidenten der Schweizer Nationalbank gewählt.

…

Dokument 2:

Die Liste der freigelassenen Schweizer:

…

Hans Gfeller

…

Markus *Meyer*

…

Rangierungsprinzip 6

- Je früher die Suchbegriffe in einem Dokument vorkommen, desto höher die Wahrscheinlich-keit für Relevanz.

Anfrage:

Hans Meyer

Dokument 1:

Hans Meyer neuer SNB-Präsident

…

…

Dokument 2:

Neue Banknoten

…

…

Gemäss *Hans Meyer* erfolgt die Ausgabe der neuen Banknoten aus Sicherheitsgründen.

Erkenntnisse Rangierungsregeln

- Wir können aus den Rangierungsregeln einige Erkenntnisse mitnehmen:

1. Keine der Regeln gilt universell. Es können immer Gegenbeispiele gefunden werden
2. Die Regeln widersprechen sich teilweise
3. Das Vorkommen von Wörtern und deren Häufigkeiten ist ein zentrales Konzept

Vektorraummodell

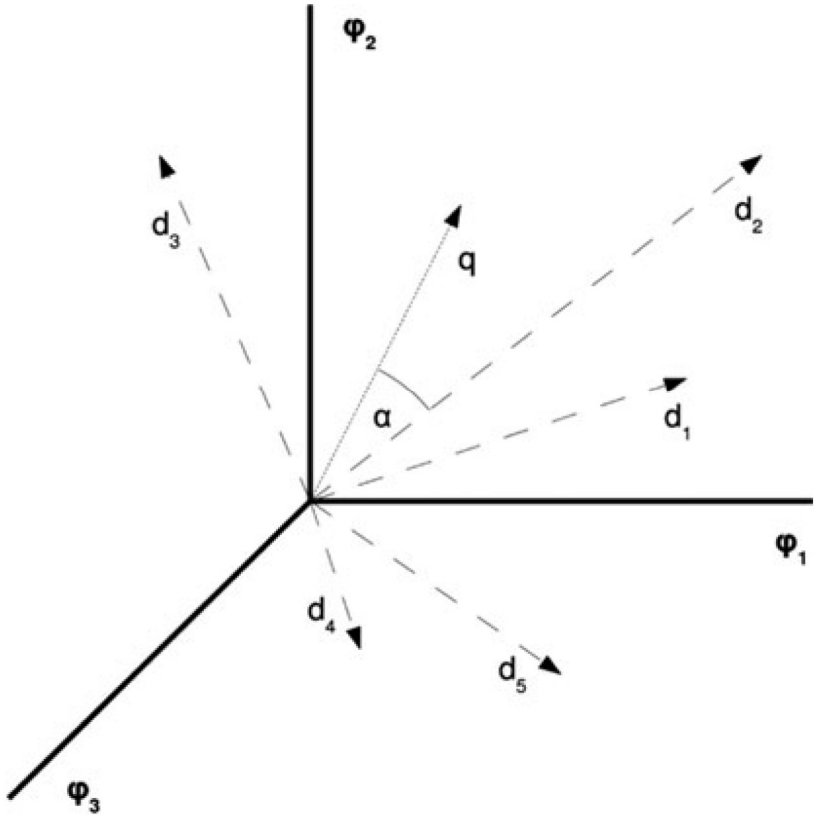
- Wir fassen die Dokumente und Anfragen als n-dimensionale Vektoren auf.
- Dieser Vektorraum ist orthogonal
- Die Basisvektoren müssen geeignet gewählt werden
  - «Kernkonzepte» der Kollektion: wäre relativ statisch, per Definition paarweise unabhängig, schwer zu bestimmen
  - Die unterschiedlichen Terme/Merkmale der Kollektion: einfach zu bestimmen, sicher nicht paarweise unabhängig, wächst beständig, extrem hochdimensional
- Ein Vergleichskriterium muss geeignet gewählt werden

Vektorraummodell

- Wir stellen Anfragen und Dokumente mittels deren «Bag of Words/Features», d.h., unseren Output des Indexierungsprozesses, als Vektoren dar.
- Die Basisvektoren sind dabei die Merkmale/Features.
- Dimensionalität dieser Vektoren n=Anzahl der verschiedenen Merkmale *in der Kollektion* (d.h., extrem hochdimensional!)
- Die Dokumentenvektoren sind dünn besiedelt (nur die Komponenten derjenigen Merkmale, die im Dokument vorkommen, sind ≠ 0)
- Vektor dj =(...,$w_{\phi i}$,...) (Gewicht $w_{\phi i}$, z.B. 1 wenn das entsprechende Merkmal im Dokument ist, 0 sonst)

Vergleichskriterium

- Dokumente als Vektoren (siehe oben)
- Anfrage als Vektor (analog zu Dokumenten)
- Sim(Dokument, Anfrage) = Winkel (je kleiner, desto besser)



Frage:

Warum nicht die euklidische Distanz?

Vektorraummodell

- Probleme:
  - Merkmale spannen einen **orthogonalen** Raum auf. Merkmalabhängigkeiten (Phrasen etc.) passen nicht in das Modell.
  - Das Modell liefert keine Antwort wie die einzelnen Merkmale zu gewichten sind (einfachste Lösung: binär)
  - Keine sauber fundierte Theorie, entspricht mehr einer «geometrischen Intuition»
- Vorteile:
  - Viele Gewichtungsschemata lassen sich über das Vektorraummodell darstellen und interpretieren
  - Einige dieser Gewichtungsschemata haben sehr gute Performance

→ In diesem Sinne ist das Vektorraummodell die Grundlage für klassisches Information Retrieval

Gewichtung: Wortstatistiken

- Die Gewichtung erfolgt aufgrund verschiedener wortbasierter Statistiken, welche aus dem Dokumentenkorpus direkt ermittelt werden:
  - ff (feature frequency; Merkmalshäufigkeit), auch tf (term frequency, Termhäufigkeit) – wie oft tritt ein Merkmal/Term in einem Dokument auf
  - df (document frequency, Dokumentenhäufigkeit) – in wievielen Dokumenten tritt ein Merkmal/Term auf
  - Dokumentlänge – ein Mass für die Länge des Dokuments:z.B. Anzahl Tokens, Anzahl Merkmale/Terme, Bytelänge
  - Position der Merkmale im Text, Inlinks/Outlinks (im folgenden nicht weiter betrachtet)
- Im folgenden wird oft von Termen statt Merkmalen gesprochen

Inverse document frequency

- Anstatt nur Worthäufigkeiten in einem Text zu zählen
  - ermittle wie viele Dokumente der Kollektion den Term beinhalten in Verhältnis zur totalen Anzahl von Dokumenten der Kollektion.
- Dies wird *Inverse Document Frequency* $idf(\varphi_k)$ von Term k genannt:

$$\mathbf{idf(\varphi_k) = log((1+N)/(1+df(\varphi_k)))}$$

- wobei
  - N: Anzahl der Dokumente in der Kollektion
  - $df(\varphi_k)$: Anzahl Dokumente, die Term k enthalten
- Mit $\mathbf{idf(\varphi_k)}$ kann die Ausprägung (Wichtigkeit; wie "charakteristisch") eines Terms für ein bestimmtes Dokument besser bestimmt werden als durch simples Zählen der Häufigkeiten.

Inverse document frequency

- Termgewichtung mit idf:
  - Die Wichtigkeit, oder das Gewicht eines Terms *k* in einem Dokument *i* nimmt zu, wenn die Häufigkeit tf(dj) steigt, nimmt aber ab, wenn die Dokumentenhäufigkeit df zunimmt.
- Dadurch wird die *Gewichtungsfunktion* wie folgt definiert:

$$w(\varphi_k,d_i) = tf(\varphi_k,d_i) \bullet idf(\varphi_k)$$

- wobei
  - $tf(\varphi_k,d_i)$: Anzahl der Vorkommen des Terms k in Dokument i
  - $idf(\varphi_k)$: inverse Dokumentenhäufigkeit des Terms k

- Eine hohes Gewicht haben dabei Terme, die in wenigen Dokumenten innerhalb der Kollektion häufig auftreten.

Umsetzung Gewichtung im VRM

- Wenn Dokumente und Anfrage als Vektoren in einem n-dimensionalen Vektorraum dargestellt werden, kann der Kosinus des Winkels wie folgt berechnet werden:

$$RSV = \frac{(q, d)}{\|q\|\|d\|} = \cos(\alpha)$$

- RSV = retrieval status value
- Dabei ist **α** der Winkel zwischen den zwei Vektoren q und d, (q,d) ist das skalare Produkt, und $\| \ \|$ kennzeichnet die Länge des Vektors.
- Der Kosinus ist = 1 für Winkel 0 Grad, und = 0 für Winkel 90 Grad
- Dieses Mass ist von der Länge der einzelnen Vektoren unabhängig – d.h., wir haben eine «Längennormaliserung»

Gewichtungsformel

- Es ergibt sich bei Kombination mit tf.idf-Gewichtung die folgende Gewichtungsformel ("tf-idf-Cosinus")

$$a_{i,j} := ff(\varphi_i, d_j) * idf(\varphi_i)$$

$$b_i := ff(\varphi_i, q) * idf(\varphi_i)$$

$$RSV(q, d_j) := \frac{\sum_{\varphi_i \in \Phi(q) \cap \Phi(d_j)} a_{i,j} * b_i}{\sqrt{\sum_{\varphi_i \in \Phi(d_j)} a_{i,j}^2} * \sqrt{\sum_{\varphi_i \in \Phi(q)} b_i^2}}$$

Vektorraummodell

$$d_1 = (\varphi_1, \varphi_2, \varphi_1, \varphi_1, \varphi_3, \varphi_2) = (3,2,1)$$
$$d_2 = (\varphi_3, \varphi_1, \varphi_1, \varphi_3) = (2,0,2)$$
$$q = (\varphi_2, \varphi_3) = (0,1,1)$$

$$P(R \mid q, d_1) := \cos(\alpha) = \frac{q \cdot d_1}{|q||d_1|} = \frac{0*3+1*2+1*1}{\sqrt{2}*\sqrt{14}} = 0{,}57$$

$$P(R \mid q, d_2) := \cos(\alpha) = \frac{q \cdot d_2}{|q||d_2|} = \frac{0*2+1*0+1*2}{\sqrt{2}*\sqrt{8}} = 0{,}50$$

Terme sind hier binär gewichtet. Die Gewichtung mit tf.idf sei als Hausaufgabe überlassen..

Aufgaben Vektorraummodell

- Was kann man über die Informationsobjekte sagen, wenn zwischen zwei Vektoren der Winkel Null ist, der eine Vektor aber doppelt so lang ist wie der zweite?
- Was bedeutet es, wenn für zwei Informationsobjekte die Vektoren gleich lang sind, der Winkel aber gross ist?

Term Discrimination Model

- Eine weitere Idee für die Termgewichtung folgt direkt aus der Idee des Vektorraummodells
- Wir berechnen, wie gut ein Term geeignet ist, Dokumente zu unterscheiden
- Abstrakt gesprochen: «gute» Terme expandieren den Vektorraum

- Die durchschnittliche Ähnlichkeit der Dokumente wird sowohl mit als auch ohne den Term berechnet

Term Discrimination Model

- Eine effiziente Berechnungsweise ist der Vergleich mit einem «virtuellen», «durchschnittlichen» Vektor («centroid vector»)

$$AVGSIM = K \cdot \sum_{i=1}^{n} sim(D^*, D_i)$$

- Wobei K der Normalisierung dient (z.B. K=1/n)
- Es sei nun $AVGSIM_k$ die «Dichte» ohne den Term k
- Dann gilt:

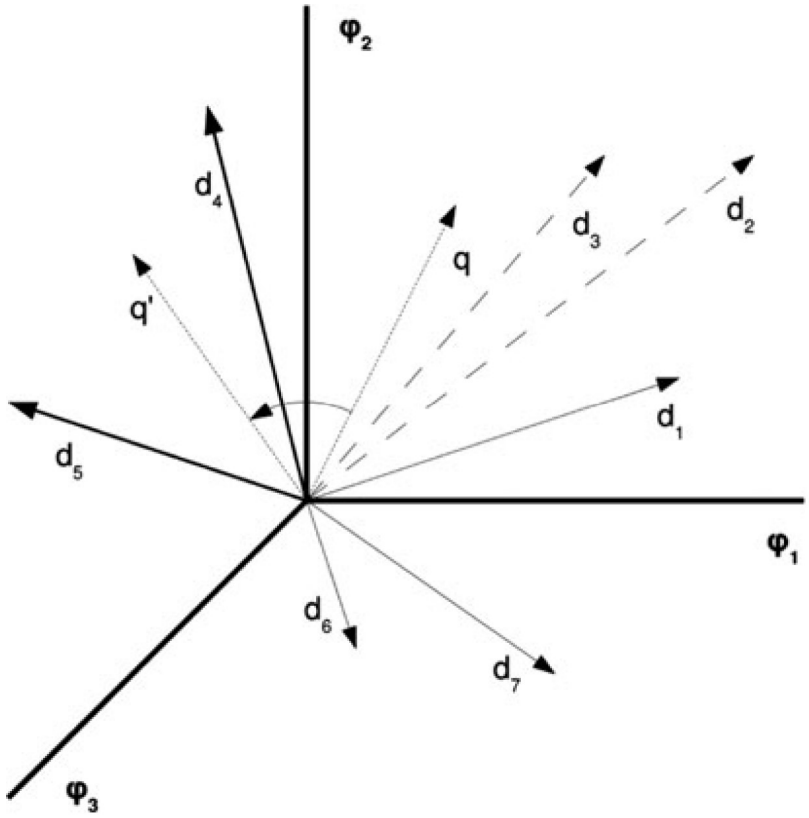
$$DISCVALUE_k = AVGSIM_k - AVGSIM$$

Term Discrimination Model

- Wir unterscheiden nun:
  - $DISCVALUE_k > 0$: (gut)
    - Term dünnt Vektorraum aus, d.h., dient der Unterscheidung von Dokumenten
    - Meistens Terme mit mittlerer Häufigkeit
  - $DISCVALUE_k \approx 0$: (neutral)
    - Hinzufügen des Terms hat kaum Effekt
    - Meistens Terme tiefer Häufigkeit (schaden Ausbeute)
  - $DISCVALUE_k < 0$: (schlecht)
    - Term verdichtet Vektorraum, d.h., erschwert die Unterscheidung von Dokumenten
    - Meist Terme mit hoher Häufigkeit (schaden Präzision)
- Hauptkritik: unterscheidet nicht relevante und irrelevante Dokumente

Vektormodell: Relevanzrückkoppelung

- = Eng. Relevance Feedback
- Idee: der/die BenutzerIn identifiziert im Resultat relevante und irrelevante Dokumente
- Es wird ein neuer Vektor konstruiert, der näher bei den relevanten Dokumenten, und weiter weg von den irrelevanten Dokumenten liegt

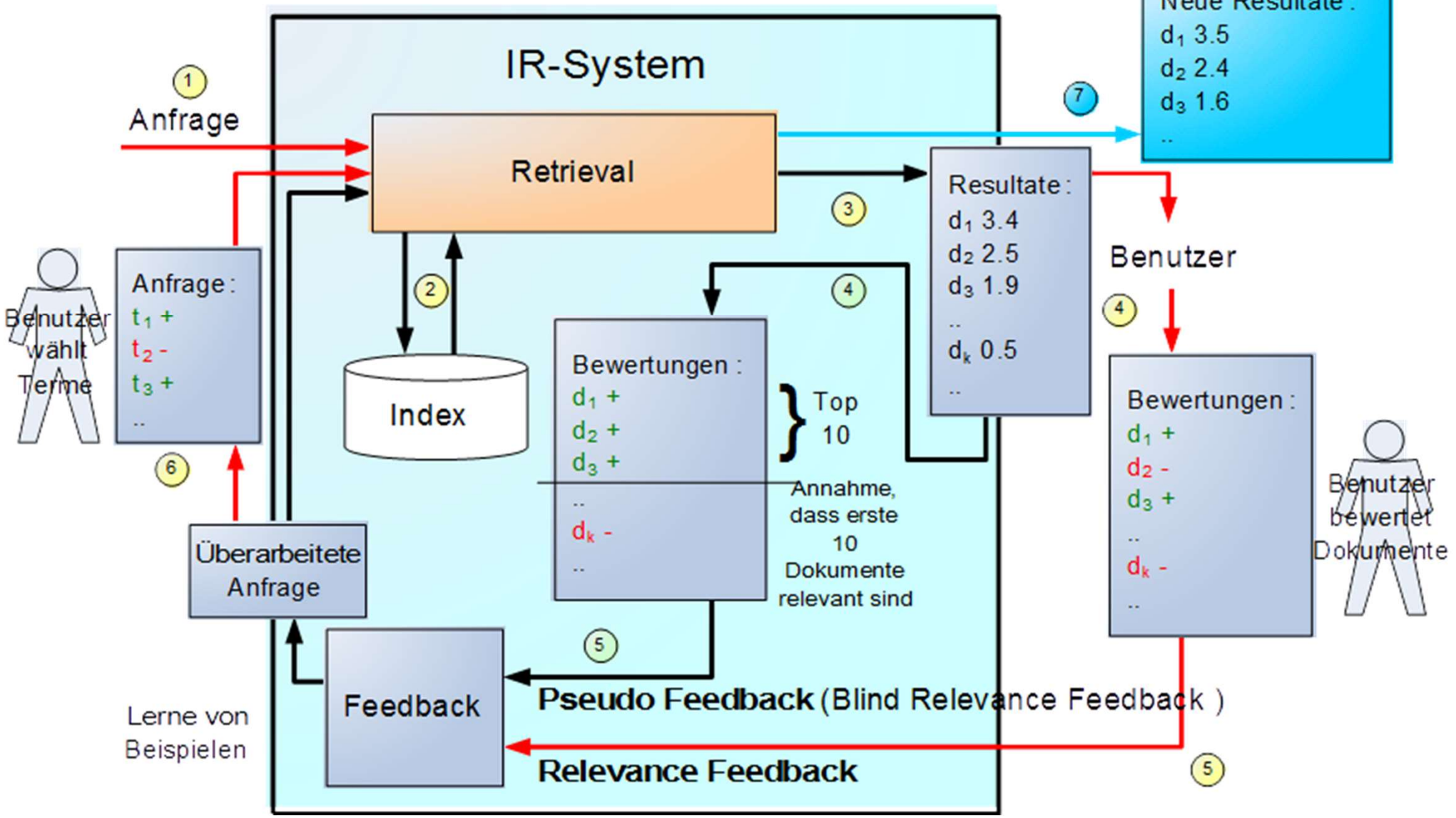


Vektormodell: Relevanzrückkoppelung

- Spielarten:
  - Originalanfrage wird neu gewichtet
  - Originalanfrage wird um neue Begriffe aus den relevanten Dokumenten ergänzt
  - Kombination aus 1) und 2)
  - Eine komplett neue Anfrage wird nur aus den relevanten Dokumenten konstruiert

Relevanzrückkoppelung

Angepasst von http://www.mias.uiuc.edu/files/tutorials/czhai02.ppt



Rocchio-Formel

- Der neue Vektor wird berechnet als:

$$\overrightarrow{q'} := \alpha \frac{\vec{q}}{||\vec{q}||} + \frac{\beta}{|D^{rel}|} \sum_{d_j \in D^{rel}} \frac{\overrightarrow{d_j}}{||\overrightarrow{d_j}||} - \frac{\gamma}{|D^{non}|} \sum_{d_k \in D^{non}} \frac{\overrightarrow{d_k}}{||\overrightarrow{d_k}||}$$

wobei $|D^{rel}|$ und $|D^{non}|$ die Anzahl der bekannten relevanten und irrelevanten Dokumente sind. α, β, γ sind Gewichtungsparameter. Oft ist γ=0.

- Beispiel für Blind Relevance Feedback (nach Jacques Savoy):

Original query	New query after pseudo relevance feedback
Who won the gold medal in the super G in Lillehammer at the Olympic Winter Games 1994?	olympic roffe-steinrotter event gold super steinrotter downhill won race time world medal slalom ski super-g skier roffe top us g
Why was the secretary general of NATO forced to resign in 1995?	clae agusta brussel mr foreign prime alliance party nato general year minister belgian belgium affair secretary scandal government resignation helicopter

Probabilistisches Retrieval

- Wir besprechen eine zweite Grosse «Familie» von Retrievalmodellen
- Idee: Retrieval ist ein Klassifizierungsprozess – die Dokumente sollen in relevant resp. nicht-relevant eingeteilt werden
- (theoretisch auch mehr als zwei Klassen denkbar, z.B. «egal» – in der Praxis selten)
- Menge der relevanten Dokumente ist << Menge der irrelevanten Dok.
- Gegeben ein bestimmtes Dokument D: berechne P(R|D)

- Unterschiedliche Schätzungen dieser Wahrscheinlichkeit führen zu unterschiedlichen Modellen.

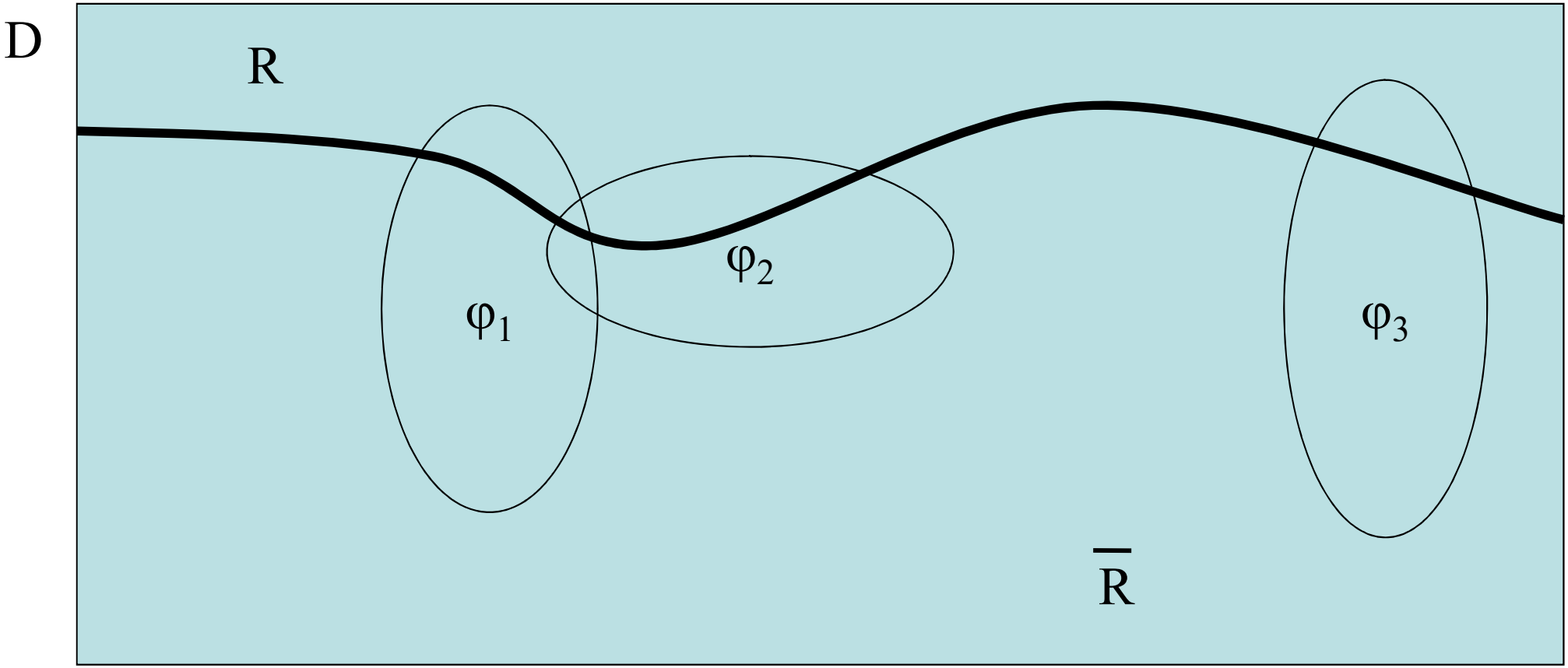
Probability Ranking Principle (PRP)

- If a *reference retrieval system's* response to each request is a ranking of the documents in the collections in order of *decreasing probability of usefulness* to the user who submitted the request, where the probabilities are *estimated as accurately as possible* on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users *will be the best* that is obtainable on the basis of that data. (S.E. Robertson, 1977)
- Das PRP ist eher eine Hypothese als ein “Prinzip” .
- *will be the best:* Es gibt Rahmenbedingungen und Annahmen, z.B. Kosten für gefundene irrelevante Dokumente und Kosten für nicht gefundene relevante Dokumente, dann kann man beweisen. PRP minimiert die Erwartungskosten.
- *Estimated as accurately as possible:* hier liegt das Problem!

Beispiel

- q =”Terrorismus, bekämpfen” = ϕ1 ϕ2
- D1 = “Gegenmassnahmen gegen Terrorismus” = ϕ3 ϕ1
- D2 = “Kampf gegen den Terror” = ϕ2 ϕ1
- D3 =”Sicherheit bei asymmetrischer Bedrohung und asymmetrische Sicherheit” = ϕ5 ϕ6 ϕ7 ϕ6 ϕ5
- D4 =”Terror bekämpfen” = ϕ1 ϕ2
- D5 =”Extremismus und Gewalt” = ϕ8 ϕ9
- D6 =”Terrorismus und innere Sicherheit” = ϕ1ϕ10 ϕ5
- Relevant erwiesen sich: R(q)={D1, D2, D3, D6}

Grafische Darstellung (q fix!)



Allgemeines zum probabilistischen Retrieval

- Probabilistisches Retrieval: Eine Gruppe von verwandten Modellen, die die wahrscheinlichkeitstheoretische Auffassung des Retrievalproblems zugrunde legen.
- Anfänge in den frühen 60ern, Maron und Kuhns.
- W. Cooper, S.E. Robertson, K. Sparck Jones kommen 1976 zu einer ersten rein probabilistischen Matching Methode (Binary Independence Retrieval, BIR).
- Die 90er: Vektorraummodell und probabilistisches Retrieval berreichern sich gegenseitig und führen zu einer Optimierung des Matchings basierend auf ff, idf, und Dokumentenlängen.
- Die Grenzen des probabilistischen Retrievals scheinen erreicht zu sein.
- Die besten probabilistischen Retrievalmodelle resp. deren Gewichtungsschemata sind immer noch kompetitiv (State-of-the-Art)

Einige “Werkzeuge” im probabilistischen Retrieval

- A, B stochastisch unabhängig, dann P(A,B)=P(A)*P(B)
- Bedingte W'keit: P(A|B)=P(A,B)/P(B).
- Bayes: P(A|B)=P(A)P(B|A)/P(B)
- Das PRP fordert RSV(q,d) = f(P(R|q,d)) + g(q), d.h., eine ordnungserhaltende Funktion auf der Wahrscheinlichkeit, dass ein Dokument d zur Anfrage q als relevant betrachtet wird
- (absolute Scores sind also für die Rangierung nicht wichtig)

Herleitung des BIR (binary independence retrieval)

Zürcher Hochschule für Angewandte Wissenschaften

- Ziel: Relevanzwahrscheinlichkeit durch Auftretenswahrscheinlichkeiten von Merkmalen darstellen
- An der Tafel! (→ siehe auch abgegebenen Text mit vereinfachter Herleitung)
- Die Annahmen/Einschränkungen sind im Wesentlichen:
  - Grundeinschränkung: $P(R|D_i)$ ist unabhängig von $P(R|D_j)$
  - Dokumente sind "Bag of Words", und ausschliesslich über diesen "Bag of Words" dargestellt
  - Dokumente und Anfrage werden durch binäre Vektoren repräsentiert
  - Einzelne Terme sind voneinander unabhängig, d.h. P(Vektor x|R,q) = Produkt der einzelnen $P(\text{Term } x_i|R,q)$
  - Terme, die nicht in der Anfrage vorkommen, sind in relevanten und irrelevanten Dokumenten gleich häufig
  - Alle nicht vom Dokument abhängigen Faktoren werden ignoriert, Logarithmus vereinfacht die Berechnung
  - Sehr wenig relevante, sehr viele irrelevante Dokumente
- Resultat RSJ (Robertson-Sparck Jones) Gewichtung. log(pi(1-qi) / (qi(1-pi)))
- → ergibt ohne Relevanzinformation eine idf-Gewichtung

Aufgabe (→ Praktikum)

- Rangieren Sie die Informationsobjekte aus unserem Beispiel mit dem BIR (RSJ-Gewichtung). Diese Rangierung ist a posteriori (weiss zum Zeitpunkt des Vergleich, welche Dokumente relevant sind).
- Vergleichen Sie mit idf-Gewichtung
- Vergleichen Sie mit tf-idf-Cosinus (Vektormodell)

Diskussion RSJ Gewichtung und BIR Modell

- Dokumente und Anfragen sind Mengen (~Boolesches Modell)
- Schätzungen nur mit Relevanzinformation möglich → ansonsten weitere Annahmen
- Bestätigt die experimentell gefundene «idf»-Gewichtung
- Modellierung des Users ist schwach!
- Merkmalshäufigkeit und andere Heuristiken sind immer noch nicht eingeflochten.
- Aber: 1994 erweitert Robertson BIR und kann eine logarithmische ff Gewichtung rechtfertigen und erreicht hohe Retrievaleffektivität (wird zitiert als BM25).

Erweiterung BIR: BM25

- Das BIR-Modell, wie oben hergeleitet, hat grobe Einschränkungen:
  - Unterstützt nur Binäre Indexierung: Term ist präsent oder nicht
  - Die resultierende Gewichtung berücksichtigt den idf-Aspekt, aber der tf fehlt
- Wie können wir Termgewichte in das Modell einbauen?

- Das sogenannte BM25-Modell (auch «Okapi»-Modell genannt) basiert auf:
  - Den RSJ-Gewichten (siehe oben): $w^{(1)}$
  - Der Termhäufigkeit im Dokument tf_{ij}
  - Der Termhäufigkeit in der Anfrage tf_{qj}
  - Der Dokumentenlänge in Relation zur durchschnittlichen Dokumentenlänge: $l(D_j)/avdl$

Gewichtungsformel BM25

- Die BM25-Formel kann dargestellt werden als:

$$RSV(D_i, Q) = \sum_{j=1}^{t} \frac{(k_1+1)\cdot tf_{ij}}{K+tf_{ij}} \cdot w^{(1)} \cdot tf_{qj}$$

$$\text{with } K = k_1 \cdot \left[(1-b) + \frac{b \cdot l(D_i)}{avdl}\right]$$

- *k1* und *b* sind dabei kollektionsspezifische Parameter. *k1* ist typischerweise zwischen 1 und 2, b zwischen 0.35 und 0.75
- «*k1* and *b* are tuning parameters for which suitable values have to be discovered by experiment» (Robertson et al., 2000)
- BM25 hat scih als erfolgreich erwiesen in vielen Retrievalszenarien: unterschiedliche Kollektionsgrössen, Sprachen, Modalitäten, …

Exact vs. Best Match

- Best Match:
  - Die Anfrage soll das bestpassende oder «gute» Dokumente beschreiben
  - Jedes Dokument ist prinzipiell zur Anfrage ähnlich
  - Resultat ist eine Rangliste aus Dokumenten
- Exact Match:
  - Die Anfrage beschreibt ein präzises Retrievalkriterium
  - Ein Dokument erfüllt das Kriterium/erfüllt dieses nicht
  - Das Resultat ist eine *ungeordnete* Menge von Dokumenten
- In der Praxis mischen Systeme oft diese beiden Paradigmen
  - Bsp: Best-Match mit einzelnen Boole'schen Operatoren

Boolesches Retrieval

- Älteste und populärste Retrievaltechnik basierend auf:
  - Existenz/Nicht-Existenz von Merkmalen
- (z.B ungewichtete Terme)
  - Boolesche Ausdrücke als Anfrage
  - Evaluation gemäss boolescher Logik
- Ergebnis einer solchen Auswertung
  - Zwei disjunkte Mengen von Objekten:
    - das Resultat
    - der Rest
  - Die Elemente dieser Mengen sind nicht nach Relevanz geordnet.

Boolean Matching

- Eine Anfrage q_i wird gegen ein Dokument $d \in D$ mit der folgenden RSV (retrieval status value) Funktion bewertet:

- $t \in T$: $RSV(t,d) = \begin{cases} \textbf{true} \text{ if } t \in d \\ \textbf{false} \text{ if } t \notin d \end{cases}$ *(einfach Anfrage)*

- $RSV(q_1$ **and** $q_2,d)$ = **true** if $RSV(q_1,d) = RSV(q_2,d)$ = **true**
- $RSV(q_1$ **or** $q_2,d)$ = **true** if $RSV(q_1,d)$ **true** oder $RSV(q_2,d)$=**true**
- RSV(**not** q,d) = **true** if RSV(q,d) = **false**

Anhaftende Probleme

- Boolesches Retrieval hat bestimmt anhaftende Probleme:
  - Das Resultat kann riesig sein, vor allem bei kurzen Anfragen.
  - Resultatmengen sind nicht nach Relevanz sortiert → kein Ranking
- Bereits kleine Anpassung bei einer Anfrage kann zu sehr unterschiedlichen Resultatmengen führen.
  - → Resultatmenge häufig zu gross oder zu klein
- Beispiel:

(data **and** compression **and** retrieval)

- versus

(text **and** compression **and** retrieval).

} kann zu Jojo-Effekt führen

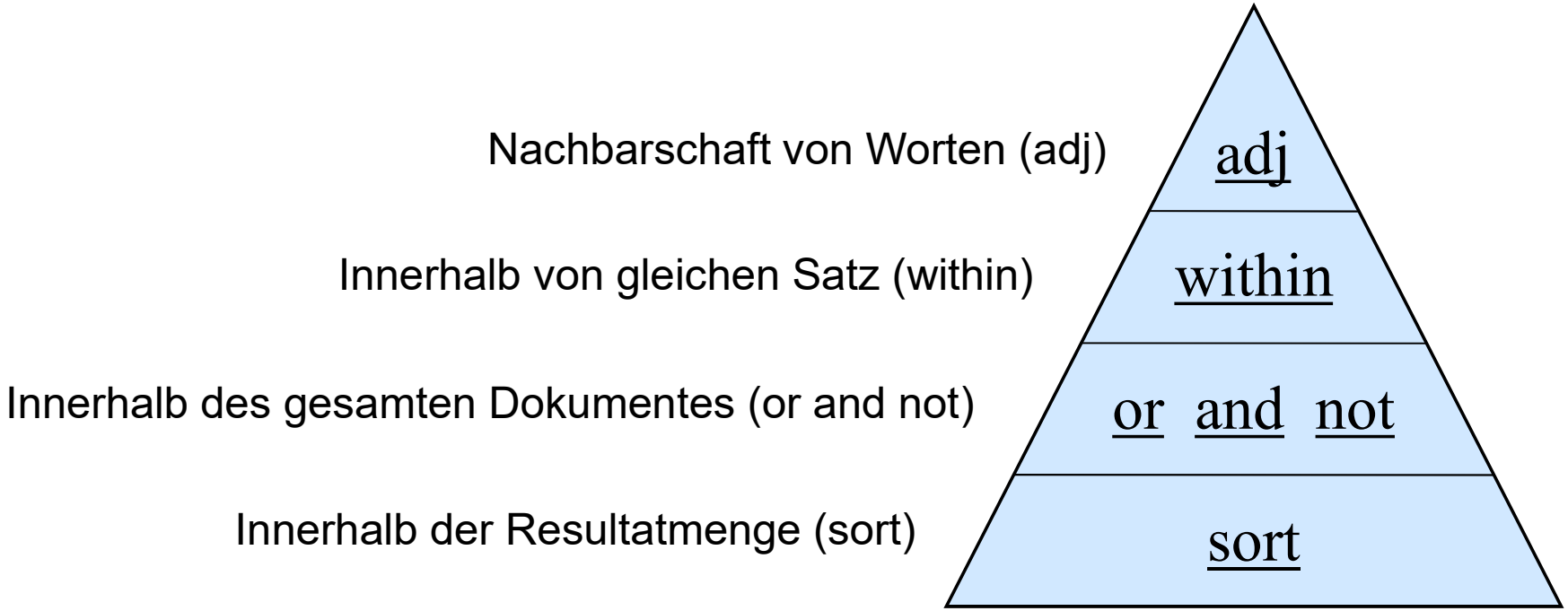
Paradox

- Weil der Benutzer wenig Einfluss auf die Grösse der Resultatmenge hat (diese aber im Gegensatz zu wahrscheinlichkeitsbasierten Rangliste wichtig ist), findet der Benutzer immer zuviel oder zuwenig
- → müsste das Resultat kennen, um eine geeignete Anfrage zu stellen

- (Eine alte Studie – Blair & Maron IP&M 1985 – zeigt, dass Nutzer den Recall solcher Systeme stark überschätzen)

Erweiterte Boolesche Operatoren

- Neben den strikten booleschen Operatoren sind speziell im WWW weitere Operatoren sind erlaubt



Zusammenfassung

- Verschiedene Retrievalmodelle.
- Liefern Ideen für “gutes” Matching (Bewältigung des Retrievalproblems)
- Nie graue Theorie allein, sondern die Kombination von Theorie, pragmatischen Experimenten und Heuristiken brachte (und bringt) Information Retrieval weiter.
- Verschiedene Retrievalmodelle sind nicht im Widerspruch sondern im Zusammenhang zu sehen, sie bereichern sich gegenseitig.
- Es gibt noch viel mehr Modelle (Logische Modelle, Language Modelling, Inferenzmodell, etc.)
- Die besprochenen Retrievalmodelle sind effektiv, aber vernachlässigen einige Grundaspekte: Subjektivität der Relevanz, das Informationsbedürfnis an sich, ein Modell für die Sprache, …