

---

# Anfrage-Operationen

## Relevance Feedback & Anfrage-Erweiterung

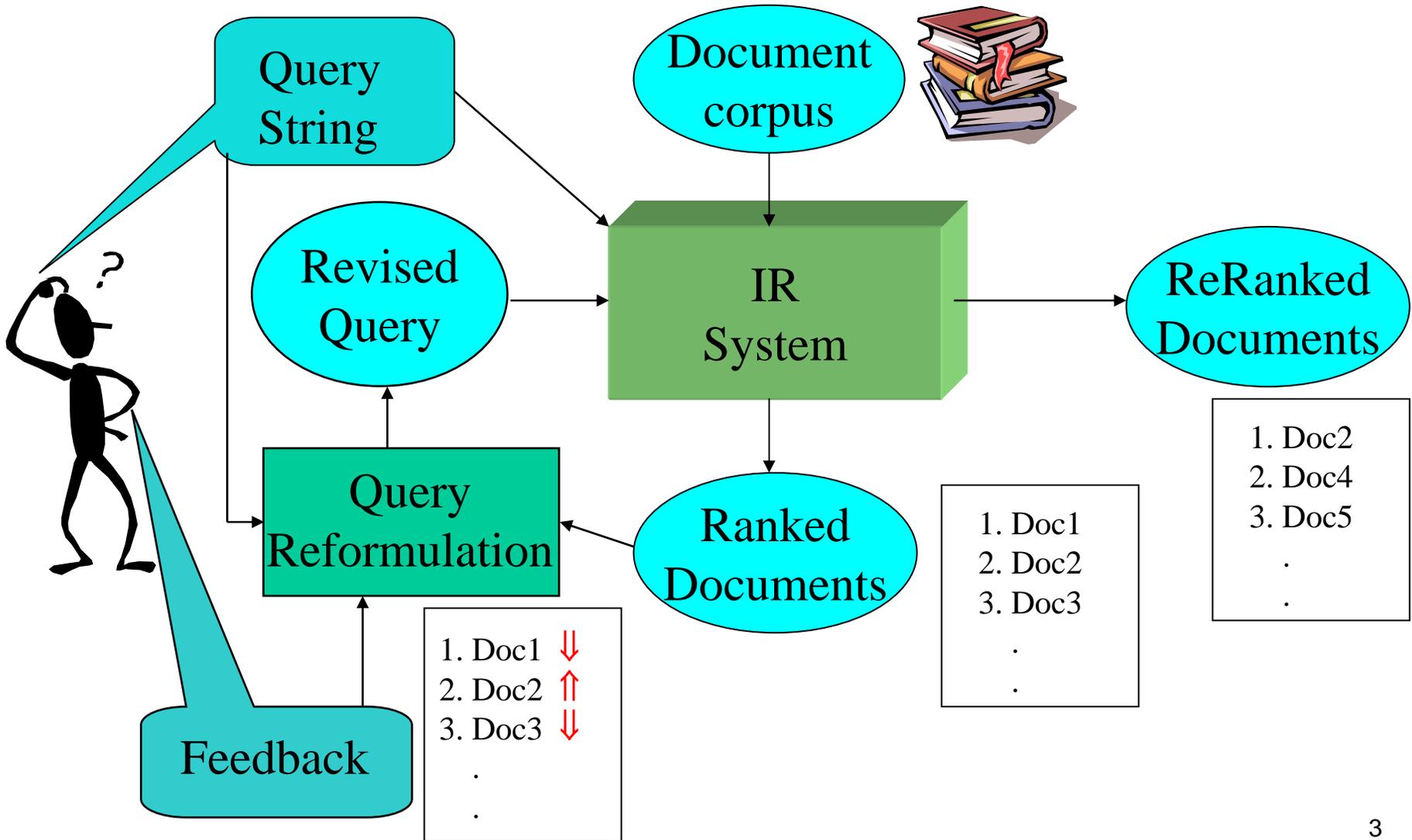
Viele Folien in diesem Abschnitt sind eine deutsche Übersetzung der Folien von Raymond J. Mooney (<http://www.cs.utexas.edu/users/mooney/ir-course/>).

# Relevance Feedback

---

- Nachdem die ersten Ergebnisse präsentiert sind, bietet das „Relevance Feedback“ dem Benutzer die Möglichkeit, Feedback zur Relevanz einzelner oder mehrerer Ergebnis-Dokumente zu geben.
- Die Feedback-Informationen werden benutzt, um die Anfrage neu zu formulieren und neue Ergebnisse auf Basis der neuen Anfrage zu berechnen.
- Dies erlaubt einen interaktiven Prozess, der mehrfach durchlaufen werden kann.

# Relevance Feedback Architektur



# Veränderung der Anfrage

---

- Änderung der Anfrage, um dem Feedback Rechnung zu tragen:
  - **Erweiterung der Anfrage (Query Expansion)**: Fügt Terme aus den relevanten Dokumenten zur Anfrage hinzu.
  - **Veränderung der Termgewichte**: Erhöht die Gewichte von Termen aus relevanten Dokumenten und reduziert die Gewichte von Termen nicht-relevanter Dokumente.
- Verschiedene Algorithmen zum automatischen Neuformulieren der Anfrage existieren.

# Veränderung der Anfrage in KSM

---

- Ändere den Anfragevektor mit Methoden der Linearen Algebra.
- **Addiere** die Vektoren der **relevanten** Dokumente zu dem Anfragevektor.
- **Subtrahiere** die Vektoren der **irrelevanten** Dokumente vom Anfragevektor.
- Dadurch werden sowohl positiv als auch negativ gewichtete Terme der Anfrage hinzugefügt. Die ursprünglichen Termgewichte können ebenfalls modifiziert werden.

# Optimale Anfrage

---

- Angenommen, die Menge  $C_r$  von relevanten Dokumenten wäre bekannt.
- Dann wäre die beste Anfrage, die die relevanten Dokumente an die Spitze stellt:

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j$$

$N$  = Gesamtzahl der Dokumente

- Die Menge  $C_r$  ist aber nicht bekannt!

# Standard Rochio-Methode

---

- Wir ändern die ursprüngliche Anfrage  $q$  mit Wissen über die **bekannt** Mengen von relevanten ( $D_r$ ) und irrelevanten ( $D_n$ ) Dokumenten.

$$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

$\alpha$ : Gewicht für ursprüngliche Anfrage.

$\beta$ : Gewicht für relevante Dokumente.

$\gamma$ : Gewicht für irrelevante Dokumente.

# Ide-Regular-Methode

---

- Da mehr Feedback den Grad der Neuformulierung erhöhen sollte, normalisiert Ide Regular nicht den Betrag des Feedbacks.

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

$\alpha$ : Gewicht für ursprüngliche Anfrage.

$\beta$ : Gewicht für relevante Dokumente.

$\gamma$ : Gewicht für irrelevante Dokumente.

# Ide-“Dec Hi”-Methode

---

- Beeinflusst in Richtung Ablehnung **nur** die am höchsten gerankten der irrelevanten Dokumente:

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{non-relevant} (\vec{d}_j)$$

$\alpha$ : Gewicht für ursprüngliche Anfrage.

$\beta$ : Gewicht für relevante Dokumente.

$\gamma$ : Gewicht für irrelevante Dokumente.

# Vergleich der Methoden

---

- Vergleichende experimentelle Ergebnisse zeigen keine klare Präferenz für eine der o.g. Methoden.
- Alle Methoden mit Feedback verbessern im allgemeinen die Retrieval-Performanz (Recall & Precision).
- Meist setzt man die Gewichte  $\alpha$ ,  $\beta$ ,  $\gamma$  jeweils auf 1.

# Evaluierung Relevance Feedback

---

- Durch ihre Konstruktion wird eine neu formulierte Anfrage explizit relevant gekennzeichnete Dokumente höher einstufen und explizit gekennzeichnete irrelevante Dokumente niedriger.
- Eine Verbesserung bei *diesen* Dokumenten sollte nicht positiv bewertet werden, da diese Information nicht vom System erzeugt wurde.
- Im Maschinellen Lernen/KDD wird dieser Fehler als “Testen mit den Trainingsdaten” bezeichnet.
- Die Evaluierung sollte nur die nicht bereits vom Benutzer bewerteten Dokumente berücksichtigen.

# Faire Evaluierung von Relevance Feedback

---

- Entferne alle Dokumente aus dem Korpus, für die Feedback geliefert wurde.
- Messe Recall/Precision der verbleibenden Dokumentensammlung.
- Verglichen mit einem kompletten Korpus können die absoluten Anzahlen von Recall/Precision abnehmen, da relevante Dokumente entfernt wurden.
- Jedoch liefert die **relative** Performance auf den verbleibenden Dokumenten faire Information über die Effektivität des Relevance Feedback.

# Warum wird Feedback nicht oft angewandt?

---

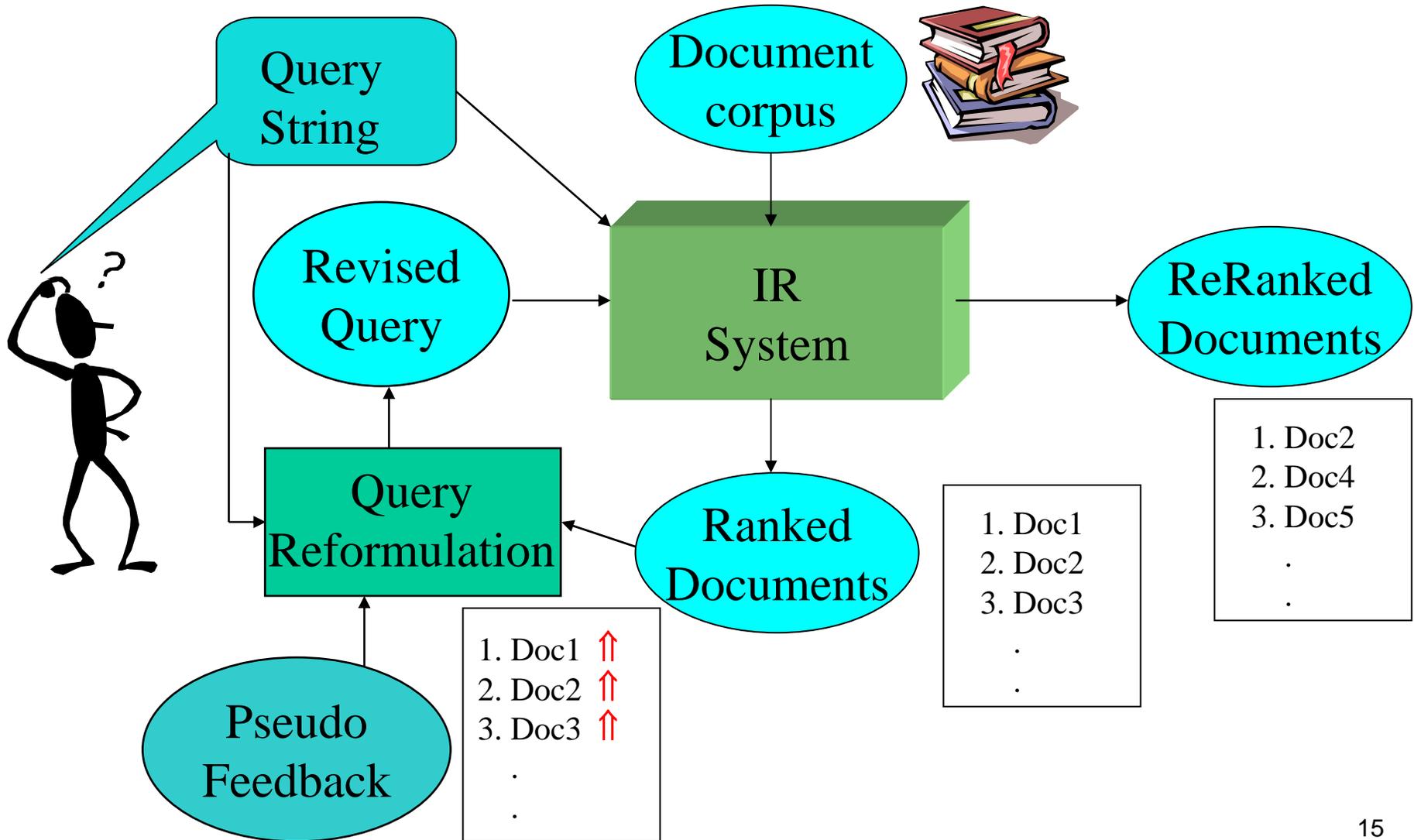
- Die Anwender zögern manchmal, explizites Feedback zu liefern.
- Rel. Feedback produziert lange Anfragen, die mehr Rechenzeit benötigen. Suchmaschinen verarbeiten oft viele Anfragen und erlauben daher wenig Zeit für jede einzelne.
- Rel. Feedback macht es schwieriger zu verstehen, warum ein spezielles Dokument gefunden wurde.

# Pseudo-Feedback

---

- Verwendet Relevance-Feedback-Methoden ohne expliziten Anwender-Input.
- Nimmt an, dass die ersten  $m$  gefundenen Dokumente relevant sind, und verwendet sie, um die Anfrage neu zu formulieren.
- Erlaubt eine Anfrageerweiterung, die Terme umfasst, die in Korrelation zu den ursprünglichen Anfragetermen stehen.

# Pseudo-Feedback-Architektur



# Pseudo-Feedback-Ergebnisse

---

- Pseudo-Feedback hat die Performanz in TREC-Wettbewerben für ad-hoc Retrieval verbessert.
- Arbeitet noch besser, wenn die besten Dokumente zusätzlichen Booleschen Bedingungen entsprechen müssen, um im Feedback verwendet zu werden.

# Thesaurus

---

- Ein Thesaurus liefert Informationen zu Synonymen und semantisch verwandten Wörtern und Phrasen.
- Beispiel:

physician

syn: ||croaker, doc, doctor, MD,  
medical, mediciner, medico, ||sawbones

rel: medic, general practitioner,  
surgeon,

# Thesaurus-basierte Anfrage-Erweiterung

---

- Erweitere jede Anfrage für jeden in ihr enthaltenen Term  $t$  mit Synonymen und verwandten Wörtern von  $t$  aus dem Thesaurus.
- Kann hinzugefügte Terme niedriger gewichten als ursprüngliche Anfrageterme.
- Erhöht im Allgemeinen den Recall.
- Kann die Precision signifikant mindern, besonders bei mehrdeutigen Termen.
  - “interest rate” → “interest rate fascinate evaluate”

# WordNet

---

- Eine lexikalische Datenbank für Englisch.
- Entwickelt von dem berühmten Kognitions-Psychologen George Miller und einem Team an der Princeton University.
- Enthält ca. 144.000 englische Nomina, Adjektive, Verben, und Adverben, die in ca. 109.000 Mengen von Synonymen gruppiert sind, die als *synsets* bezeichnet werden.

# Synset-Beziehungen in WordNet

---

- **Antonym**: front → back
- **Attribute**: benevolence → good (Nomen zu Adjektiv)
- **Pertainym**: alphabetical → alphabet (Adjektiv zu Nomen)
- **Similar**: unquestioning → absolute
- **Cause**: kill → die
- **Entailment**: breathe → inhale
- **Holonym**: chapter → text (Teil-von)
- **Meronym**: computer → cpu (Ganzes-von)
- **Hyponym**: tree → plant (Spezialisierung)
- **Hypernym**: fruit → apple (Verallgemeinerung)

# WordNet-basierte Anfrage-Erweiterung

---

- Füge Synonyme aus dem gleichen Synset hinzu.
- Füge Hyponyme hinzu, um speziellere Terme zu ergänzen.
- Füge Hypernyme hinzu, um eine Anfrage zu verallgemeinern.
- Füge weitere verwandte Terme hinzu, um die Anfrage zu erweitern.

# Statistischer Thesaurus

---

- Manuell entwickelte Thesauri sind nicht in allen Sprachen verfügbar.
- Manuell erstellte Thesauri sind vom Typ und vom Umfang der Synonymität begrenzt, sowie in den semantischen Beziehungen, die sie darstellen.
- Semantisch verwandte Terme können alternativ mit statistischen Korpus-Analysen entdeckt werden.

# Automatische Globalanalyse

---

- Bestimme die Termähnlichkeit durch eine vorberechnete statistische Analyse des kompletten Korpus.
- Berechne Assoziationsmatrizen, die die Korrelation von Termen durch die Häufigkeit ihres gemeinsamen Auftretens quantifizieren.
- Erweitere Anfragen mit den statistisch ähnlichsten Termen.

# Assoziations-Matrix

---

	$w_1$	$w_2$	$w_3$	.....	$w_n$
$w_1$	$c_{11}$	$c_{12}$	$c_{13}$	.....	$c_{1n}$
$w_2$	$c_{21}$				
$w_3$	$c_{31}$				
.	.				
.	.				
$w_n$	$c_{n1}$				

$c_{ij}$ : Korrelationsfaktor zwischen Term  $i$  und Term  $j$

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}$$

$f_{ik}$ : Häufigkeit von Term  $i$  in Dokument  $k$

# Normalisierte Assoziations-Matrix

---

- Ein auf Häufigkeit basierter Korrelationsfaktor begünstigt häufigere Terme.
- Normalisierter Assoziations-Score:

$$s_{ij} = \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}$$

- Normalisierter Score ist 1, wenn zwei Terme in allen Dokumenten die gleiche Häufigkeit besitzen.

# Metrische Korrelations-Matrix

---

- Die Assoziations-Korrelation berücksichtigt nicht die Nähe der Terme in den Dokumenten, sondern nur die Häufigkeit des gemeinsamen Auftretens in Dokumenten.
- Die metrische Korrelation berücksichtigt auch die Termnähe.

$$c_{ij} = \sum_{k_u \in V_i} \sum_{k_v \in V_j} \frac{1}{r(k_u, k_v)}$$

$V_i$ : Menge aller Vorkommen von Term  $i$  im Korpus.

$r(k_u, k_v)$ : Wortabstände zwischen Wortvorkommen  $k_u$  und  $k_v$   
(=  $\infty$ , falls  $k_u$  und  $k_v$  in verschiedenen Dokumenten vorkommen).

# Normalisierte metrische Korrelations-Matrix

---

- Normalisierter Score zur Berücksichtigung von Termhäufigkeiten:

$$s_{ij} = \frac{c_{ij}}{|V_i| \times |V_j|}$$

# Anfrageerweiterung mit Korrelations- Matrizen

---

- Für jeden Term  $i$  in der Anfrage erweitere die Anfrage mit den  $n$  Termen  $j$ , die den höchsten Wert von  $c_{ij}$  (oder  $s_{ij}$ ) haben.
- Dies fügt semantisch verwandte Terme aus der “Nachbarschaft” der Anfrageterme hinzu.

# Probleme mit der Globalanalyse

---

- Termmehrdeutigkeit kann irrelevante, statistisch korrelierte Terme einbeziehen.
  - “Apple computer” → “Apple red fruit computer”
- Da diese Terme konstruktionsbedingt hoch korreliert sind, kann es sein, dass die Erweiterung keine zusätzlichen Dokumente findet.

# Automatische Lokalanalyse

---

- Bestimme zur Anfragezeit auf dynamische Weise die ähnlichen Terme:
- Basiere für jede spezifische Anfrage die Korrelationsanalyse nur auf der “lokalen” Menge der am besten bewerteten Dokumente zu dieser Anfrage.
- Vermeide Mehrdeutigkeiten durch das Bestimmen von ähnlichen (korrelierten) Termen nur innerhalb der relevanten Dokumente.
  - “Apple computer” →  
“Apple computer Powerbook laptop”

# Global- versus Lokalanalyse

---

- Die Globalanalyse erfordert nur einmal beim Aufbau des Systems eine aufwändige Termkorrelations-Berechnung.
- Die Lokalanalyse erfordert eine intensive Termkorrelations-Berechnung zur Laufzeit bei jeder Anfrage (obwohl die Anzahl der Terme geringer ist als bei der Globalanalyse).
- Aber Lokalanalysen liefern bessere Ergebnisse.

# Globalanalyse-Verbesserungen

---

- Erweitere Anfragen nur mit den Termen, die *allen* Termen in der Anfrage ähnlich sind.

$$sim(k_i, Q) = \sum_{k_j \in Q} c_{ij}$$

- ergänze “fruit” nicht zu “Apple computer” da dies weit von Computer “computer” entfernt ist.
  - “fruit” wird ergänzt zu “apple pie” da “fruit” da sowohl nahe an “apple” als auch an “pie”.
- Verwende bei der Berechnung von Termkorrelationen weiterentwickelte Termgewichte (anstatt nur Häufigkeit).

# Schlussfolgerungen zur Anfrageerweiterung

---

- Anfrageerweiterungen mit verwandten Termen können die Performance – besonders den Recall – verbessern.
- Jedoch müssen ähnliche Terme sehr vorsichtig ausgewählt werden, um Probleme wie geringere Precision zu vermeiden.