In der Vorlesung behandelt wurden die Folien bis 41. Diese sind relevant für die Klausur.
Die anderen Folien sind aber sicher trotzdem spannend.

Information Retrieval
Eine Einführung

Günter Neumann
neumann@dfki.de
LT lab, DFKI

(Verwende Folien von Raymond Mooney's IR Kurs
http://www.cs.utexas.edu/users/mooney/ir-course/)

Buchempehlung: „Finding out about“, R. K. Belew

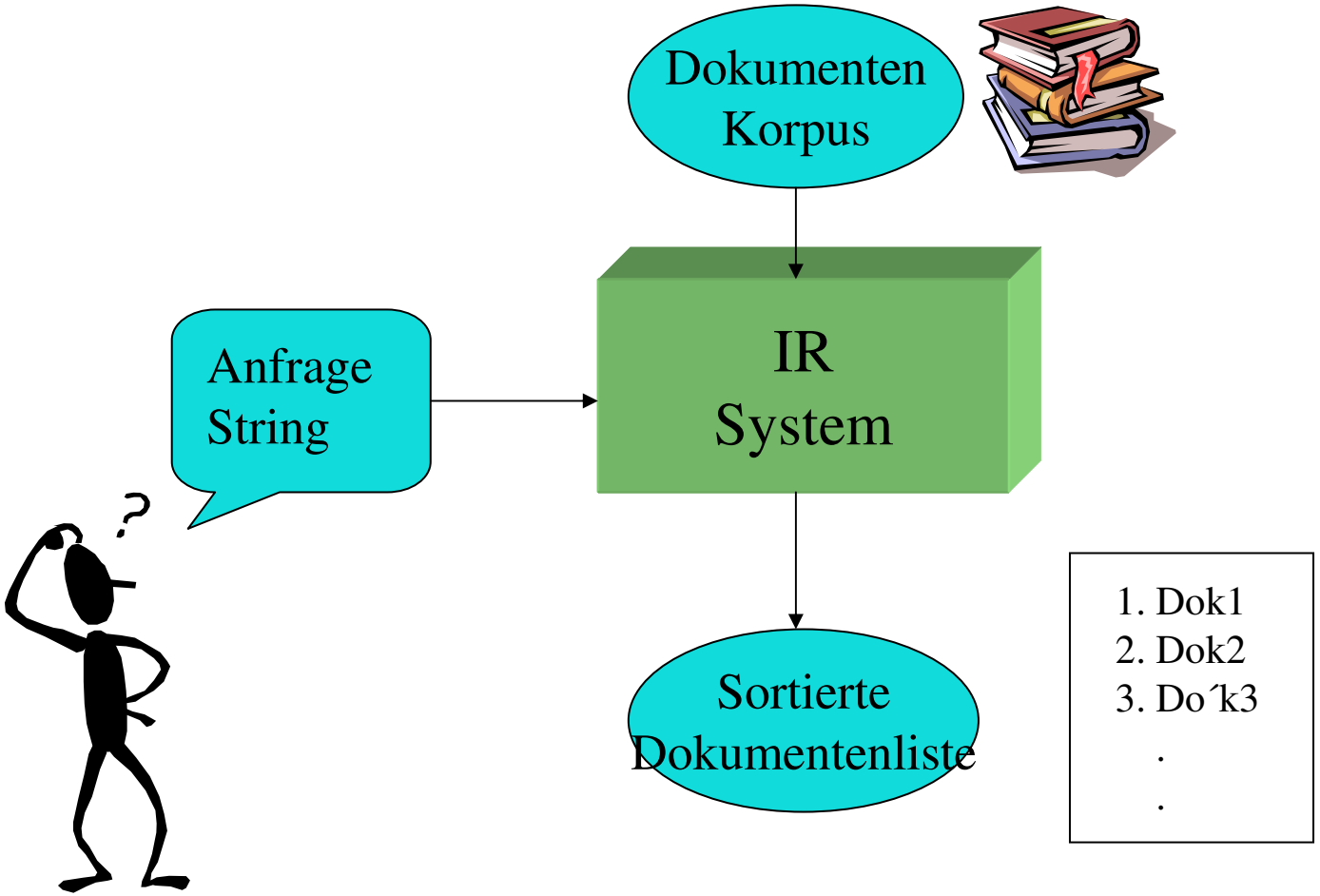
Information Retrieval (IR)

- Das Indizieren und das Abrufen von Texten.
- Das Suchen von Seiten im Web ist die aktuellste "Killer Anwendung".
- Betrachtet in erste Linie das Abrufen von *relevanten* Dokumenten für eine Anfrage.
- Betrachtet in zweiter Linie den schnellen Zugriff auf *große* Textmengen.

Typische IR Aufgaben

- Gegeben:
  - Ein Korpus von natürlichsprachlichen Dokumenten.
  - Eine Benutzer-Anfrage in Form von Schlüsselwörtern.
- Finde:
  - Eine sortierte Menge von Dokumenten, die relevant für die Anfrage ist.

IR System


Dokumenten Korpus
Anfrage String
IR System
Sortierte Dokumentenliste
1. Dok1
2. Dok2
3. Do´k3
.
.

Relevanz

- Relevanz ist eine subjektives Maß und beinhaltet:
  - Den richtigen Gegenstandsbereich betreffend
  - Den richtigen Zeitraum betreffend
  - Die richtige Autorität betreffend (vertrauenswürdige Quellen haben)
  - Die Ziele des Benutzers und die beabsichtige Verwendung der Information betreffend (*Informationsbedarf*)

Schlüsselwortsuche

- Einfachste Sicht von Relevanz ist, dass der Anfragestring wörtlich im Dokument vorkommt.
- Etwas weniger strikte Sicht ist, dass die Schlüsselwörter häufig, aber in beliebiger Reihenfolge vorkommen (*bag of words*).
  - Dokumente müssen „über die Schlüsselwörter sprechen“

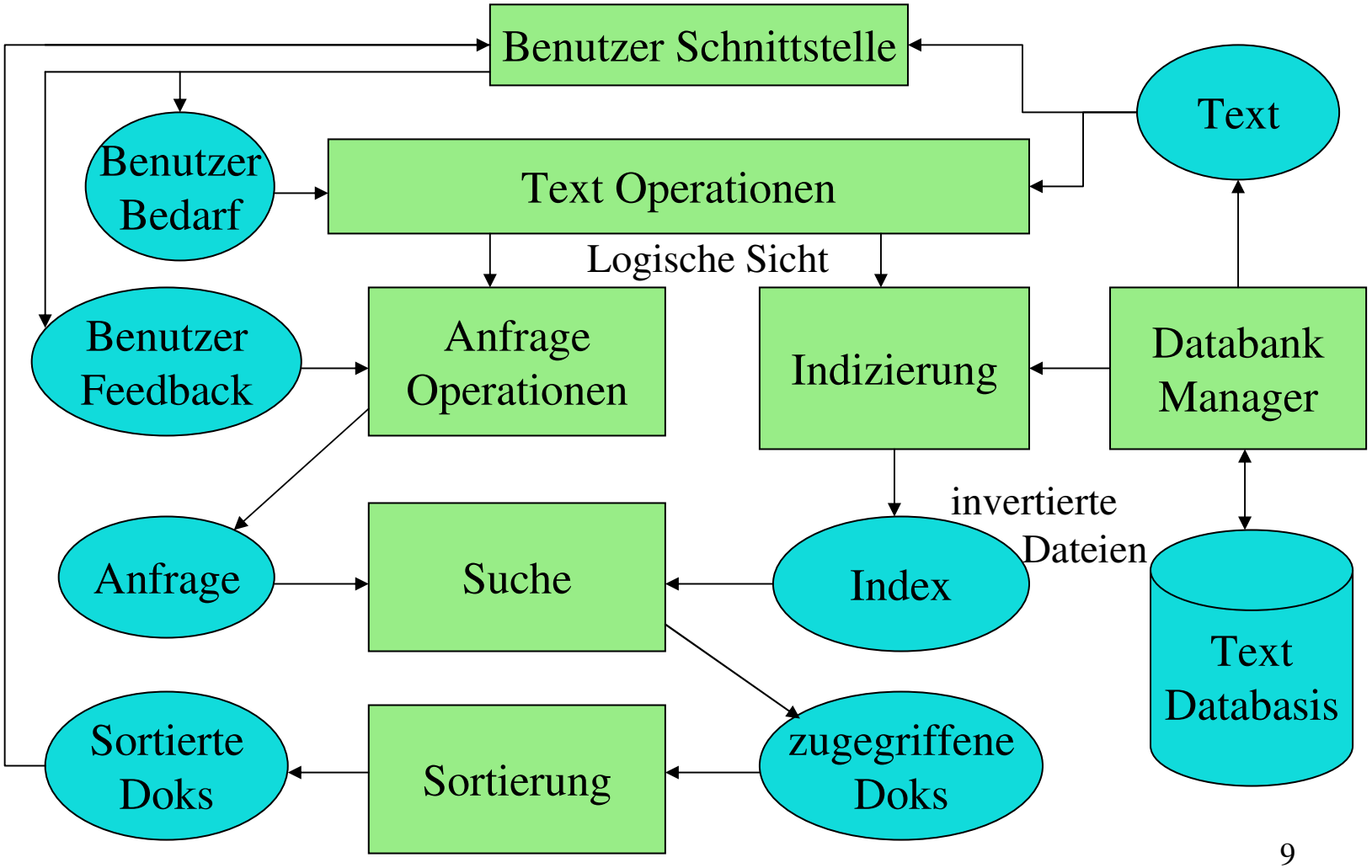
Probleme mit Schlüsselwörtern

- Können eventuell relevante Dokumente nicht abrufen, die synonyme Terme enthalten.
  - "Restaurant" vs. "Café"
  - "VRC" vs. "China"
- Können eventuell nicht relevante Dokumente abrufen, die mehrdeutige (ambige) Terme enthalten.
  - "Maulwurf" (Säugetier vs. verdeckter Informant)
  - "Apple" (Computerfirma vs. Frucht)
  - "Schröder" (Altbundeskanzler vs. Fleischwaren)

Intelligentes IR

- Die Bedeutung (*Semantik*) von Wörtern heranziehen.
- Die *Reihenfolge* von Wörtern heranziehen.
- Sich an den Benutzer *anpassen* durch direkte oder indirekte Rückmeldung (Feedback).
- Die *Autorität/Glaubwürdigkeit* der Informationsquelle heranziehen.

IR Systemarchitektur



IR Systemkomponenten

- Text Operationen berechnen Index-Wörter (Tokens/Terme).
  - Entfernung von Stoppwörtern
  - Berechnung von Wortstämmen (Stemming)
- Indizierung konstruiert eine invertierten Index zum Verweisen von Wörtern auf Dokumente.
- Suche ruft über den invertierten Index Dokumente ab, die ein gegebenes Schlüsselwort enthalten.
- Sortierung (Ranking) gewichtet alle abgerufenen Dokumente gemäß eines Relevanzmetrik.

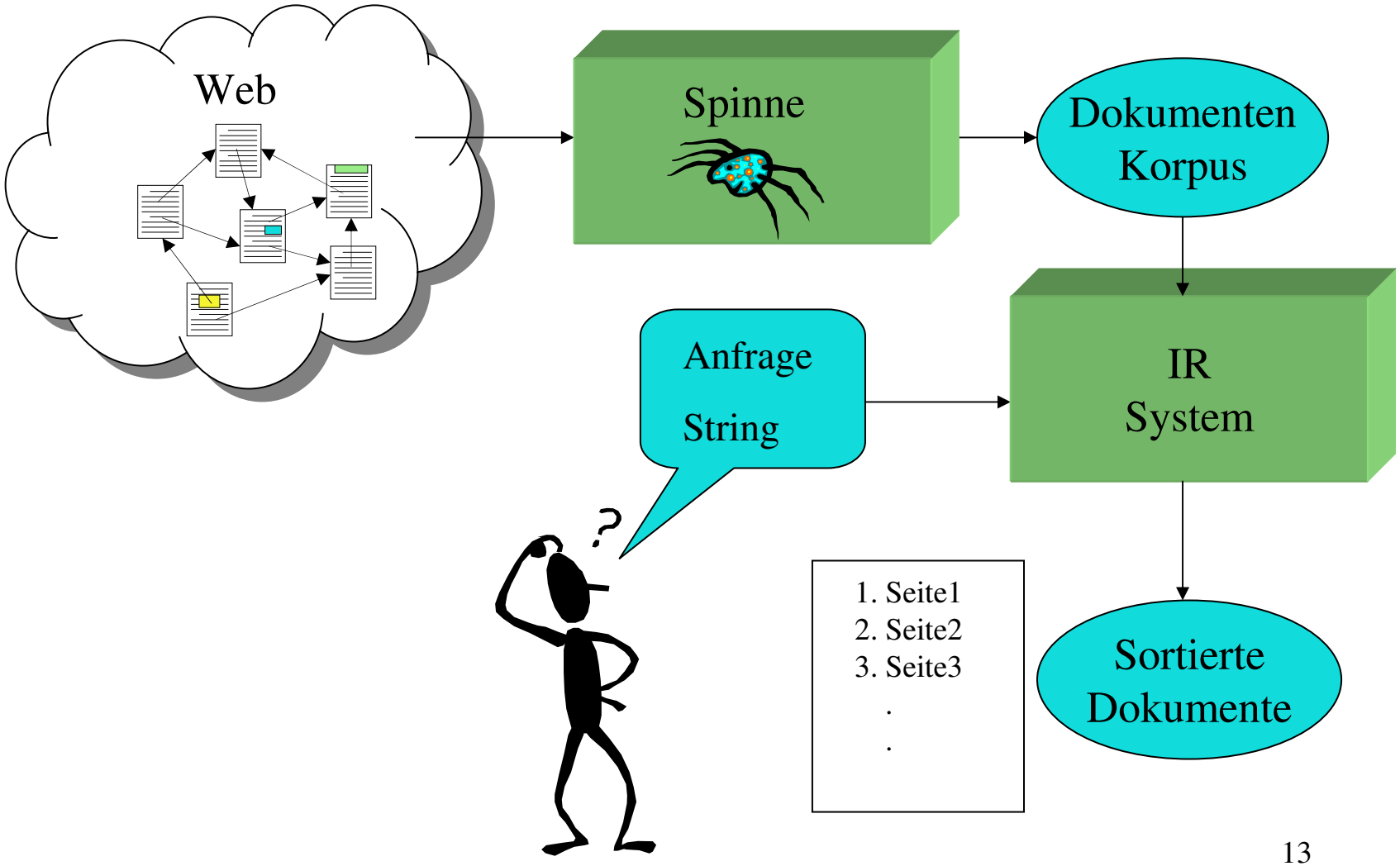
IR Systemkomponenten (fortgesetzt)

- Benutzerschnittstelle verwaltet die Interaktionen mit dem Benutzer:
  - Anfrageeingabe und Dokumentenausgabe.
  - Relevanz-Rückmeldung.
  - Darstellung der Resultate.
- Anfrageoperationen transformieren die Anfrage, um den Informationsabruf zu verbessern:
  - Expansion der Anfrage mittels eines Thesaurus.
  - Transformation der Anfrage mittels Relevanz-Rückmeldung.

Web-Suche

- Anwendung von IR auf HTML-Dokumente im WWW (World Wide Web):
- Unterschiede:
  - Erstellt den Dokumentenkorpus durch herumlaufen im Web (spidering, crawling).
  - Kann die strukturelle Layoutinformation in HTML (XML) ausnutzen.
  - Dokumente ändern sich unkontrolliert.
  - Kann die Verbindungsstruktur (links) im Web ausnutzen.

Web-Such-System



Aktuelle IR-Geschichte

- 2000er
  - Verbindungsanalyse für die Web-Suche
    - Google
  - Automatische Informationsextraktion
    - Whizbang
    - Fetch
    - Burning Glass
  - Frageantwort (Question Answering)
    - TREC Q/A track
    - Clef multilingual QA track

Aktuelle IR-Geschichte

- 2000er fortgesetzt:
  - Multimediale IR
    - Bilder
    - Videos
    - Audio und Music
  - Sprachübergreifende (cross-lingual) IR
    - DARPA Tides
    - Clef
  - Dokumentenzusammenfassung
    - DUC

Verwandte Bereiche

- Datenbankverwaltung
- Bibliotheks- und Informationswissenschaft
- Künstliche Intelligenz
- Natürlichsprachliche Verarbeitung
- Maschinelles Lernen

Datenbankverwaltung

- Fokussiert auf strukturierte Daten, die in relationalen Tabellen gespeichert sind, anstatt auf formfreie Texte.
- Fokussiert auf die effiziente Verarbeitung von wohldefinierten Anfragen in Form einer formalen Sprache (SQL).
- Klarere Semantiken für Daten und Anfragen.
- Die aktuelle Orientierung in Richtung semistrukturierten Daten (XML) bringt es näher heran an IR.

Bibliotheks- und Informationswissenschaft

- Fokussiert auf die Mensch-Maschine-Aspekte von IR (Mensch-Maschine-Interaktionen, Benutzerschnittstellen, Visualisierung).
- Beschäftigt sich mit der effektiven Kategorisierung von menschlichem Wissen.
- Beschäftigt sich mit der Analyse von Zitierungen und Bibliometrie (Struktur von Information).
- Aktuelle Arbeiten im Bereich der digitalen Bibliotheken bringt es näher heran an Informatik und IR:
  - http://citeseer.ist.psu.edu/
  - http://libra.msra.cn/

CiteSeer.IST
Scientific Literature Digital Library

Künstliche Intelligenz

- Fokussiert auf die Darstellung von Wissen, Inferenz und intelligenten Aktionen.
- Formalismen für die Darstellung von Wissen und Anfragen:
  - Prädikatenlogik erster Stufe
  - Bayesche Netzwerke
- Aktuelle Arbeiten in den Bereichen Web-Ontologien und intelligente Informationsagenten bringen es näher heran an IR:
  - Semantic Web

Natürlichsprachliche Verarbeitung

- Fokussiert auf die syntaktische, semantische und pragmatische Analyse von natürlicher Sprache und Diskurs.
- Die Fähigkeit zur Analyse von Syntax (Phrasenstruktur) und Semantik könnte einen Informationsabruf mit Bedeutung erlauben anstatt mit Schlüsselwörtern.

Natürlichsprachliche Verarbeitung: IR Richtungen

- Methoden zur Bestimmung der „richtigen“ Bedeutung eines mehrdeutigen Wortes basierend auf Kontext (*word sense disambiguation*).
- Methoden zur Identifikation einer spezifischen Information in einem Dokument (*information extraction*).
- Methoden zur Beantwortung von spezifischen natürlichsprachlichen Fragen in Dokumenten (*open domain QA*).

Maschinelles Lernen

- Fokussiert auf die Entwicklung von Computersystemen, die ihre Performanz durch Erfahrung verbessern können.
- Automatische Klassifikation von Beispielen auf der Basis des Erlernens von Konzepten mit Hilfe von ettiketierten Trainingsbeispielen (*supervised learning*).
- Automatische Methoden zur Verteilung (Clustering) von nicht ettiketierten Beispielen in sinnvolle Gruppen (*unsupervised learning*).

Maschinelles Lernen: IR Richtungen

- Textkategorisierung
  - Automatische hierarchische Klassifizierung (Yahoo).
  - Anpassbares Filtern/Planen/Empfehlen.
  - Automated spam filtering.
- Text Clustering
  - Clustern von Ergebnissen von IR Anfragen.
  - Automatisches Formieren von Hierarchien (Yahoo).
- Lernen für Informationsextrakton
- Text Mining

Topiks, die auf den nächsten Folien behandel werden

- Vector space model
- (Text processing aspects
- Evaluation
- Concept-based IR)

Das kann natürlich
nur einen Überblick geben!

Stichpunkte für Vektorraummodel

- Wir bestimmt man wichtige Wörter in einem Dokument?
  - Wortbedeutung?
  - Wort-N-Gramm (und Phrasen, Idiome,…) → Terme
- Wie bestimmt man den Grad der Wichtigkeit eines Terms in einem Dokument und in einer gesamten Dokumentenmenge?
- Wie bestimmt man den Grad der Ähnlichkeit zwischen einem Dokument und der Anfrage?
- Im Falle des Webs, was ist ein Korpus und welchen Einfluß haben Links, Formatierung, etc.?

Das Vektorraummodel (VRM)

- Nimm an, dass nach einer Vorverarbeitung t *unterschiedliche* Terme übrigbleiben; nenne sie die Indexterme oder das Vokabular.
- Diese "orthogonalen" Terme formen einen Vektorraum.

  Dimension = t = |Vokabular|
- Jeder Term i in einem Dokument oder Anfrage j erhält ein reellwertiges Gewicht w_{ij}
- Dokumente und Anfragen werden jeweils als t-dimensionale Vektoren dargestellt:

  $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$

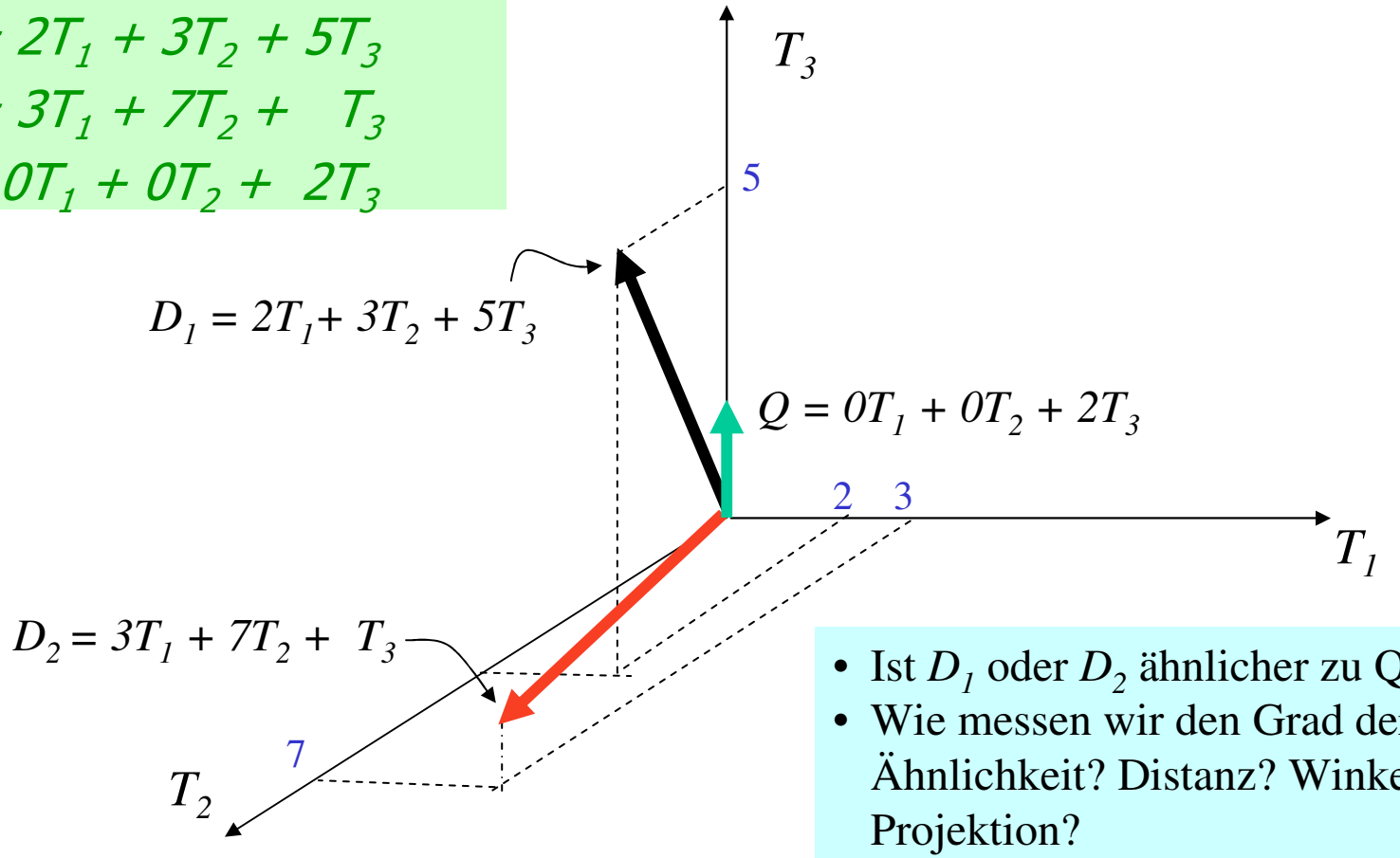
Graphische Darstellung

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$



- Ist D_1 oder D_2 ähnlicher zu Q?
- Wie messen wir den Grad der Ähnlichkeit? Distanz? Winkel? Projektion?

27

Dokumentenkollektion

- Eine Kollektion von *n* Dokumenten kann in einem Vectorraummodel durch eine Term-Dokumenten Matrix dargestellt werden.
- Ein Eintrag in dieser Matrix entspricht dem Gewicht eines Terms in dem Dokument; Null bedeutet, dass der Term keine Relevanz in dem Dokument hat oder das er ganz einfach in dem Dokument nicht vorkommt.

$$\begin{array}{c|cccc} & T_1 & T_2 & \ldots. & T_t \\ D_1 & w_{11} & w_{21} & \ldots & w_{t1} \\ D_2 & w_{12} & w_{22} & \ldots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \ldots & w_{tn} \end{array}$$

Gewichte von Termen: Häufigkeit von Termen

- Häufigere Terme in einem Dokument sind wichtiger als weniger häufigere Terme, d. h. sie sind bezeichnender für den Topik.

  f_{ij} = Frequenz des Terms *i* in Dokument *j*

- Wir wollen die Termfrequenz (*tf*) über einen gesamten Korpus hinweg normalisieren:

  $tf_{ij} = f_{ij} / max\{f_{ij}\}$

Gewichte von Termen : Umgekehrte Dokumentenfrequenz

- Terme, die in vielen verschiedenen Dokumenten auftreten, sind für den übergeordneten Topik nicht kennzeichnend.

  df_i = Dokumentenfrequenz von Term i

  = Anzahl der Dokumente, die Term i enthalten

  idf_i = umgekehrte (inverse) Dokumentenfrequenz von Term i,

  = $\log_2(N/df_i)$

  (N: Gesamtanzahl der Dokumente)
- Ein Hinweis auf das Unterscheidungspotential eines Terms.
- Logarithmus hilft, den Einfluss relativ zu *tf* zu dämpfen.

Gewichtung mit TF-IDF

- Ein typischer kombinierter Termindikator ist die A *tf-idf Gewichtung*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i)$$

- Ein Term, der häufig in einem Dokument vorkommt, aber seltener im Rest der Dokumentensammlung bekommt (für dieses Dokument) ein hohes Gewicht, also eine hohe Relevanz.
- Es wurden viele andere Methoden vorgeschlagen, wie Gewichte von Termen zu bestimmen sind.
- Aus experimenteller Sicht konnte aber gezeigt werden, das *tf-idf* sehr gut arbeitet.

Berechnung von TF-IDF - Ein Beispiel

- Gegeben sein ein Dokument, dass folgende Terme mit ihren Gewichten enthält:
  - A(3), B(2), C(1)
- Nimm eine Sammlung an, die 10.000 Dokumente enthät und für diese Terme folgende Dokumentenfrequenz:
  - A(50), B(1300), C(250)
- Dann ergeben sich folgende kombinierten Termgewichte:
  - A: tf = 3/3; idf = log(10000/50) = 5.3; tf-idf = 5.3
  - B: tf = 2/3; idf = log(10000/1300) = 2.0; tf-idf = 1.3
  - C: tf = 1/3; idf = log(10000/250) = 3.7; tf-idf = 1.2

Anfragevektor

- Der Anfragevektor wird typischerweise als Dokument betrachtet und entsprechend via tf-idf gewichtet.
- Alternativ könnte eine Benutzer die Gewichte für die einzelnen Schlüsselwörter selber vornehmen.

Ähnlichkeitsmaße

- Ein Ähnlichkeitsmaß A ist eine Funktion, die den Grad der Ähnlichkeit zwischen zwei Vektoren berechnet.

- Die Verwendung eines Ähnlichkeitsmaßes zwischen der Anfrage und jedem Dokument erlaubt:
  - Es ist Möglichkeit, die abgerufenen Dokumente nach ihrer angenommenen Relevanz zu sortieren.
  - Es ist möglich einen Schwellwert zu erzwingen, mit dem die Anzahl der abgerufenen Dokumente kontrolliert werden kann.

Ähnlichkeitsmaß – Innere Produkt

- Ähnlichkeit zwischen den Vektoren des Dokuments und der Anfrage kann als das innere Vektorprodukt berechnet werden:

$$\text{sim}(\boldsymbol{d}_j, \boldsymbol{q}) = \boldsymbol{d}_j \bullet \boldsymbol{q} = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$

  wobei w_{ij} das Gewicht von Term *i* im Dokument *j* ist und w_{iq} das Gewicht von Term *i* der Anfrage.

- Für binäre Vektoren ist das innere Produkt gerade die Anzahl der passenden Anfrageterme im Dokument (also die Größe der Schnittmenge).
- Für gewichtete Termvektoren ist es die Summe der Produkte der Gewichte der passenden Terme.

Eigenschaften des inneren Produktes

- Bevorzugt lange Dokumente mit einer großen Anzahl von eindeutigen Termen.
- Misst, wie viele Terme passen, aber nicht, wie viele Terme *nicht* passen.

Inneres Produkt – Ein Beispiel

Binär:

	retrieval	database	architecture	computer	text	management	information
– D =	1,	1,	1,	0,	1,	1,	0
– Q =	1,	0 ,	1,	0,	0,	1,	1

Größe vom Vektor = Größe des Vokabulars = 7
0 bedeutet, dass der entsprechende Term nicht im Dokument oder der Anfrage vorkommt.

sim(D, Q) = 3

Gewichtet:

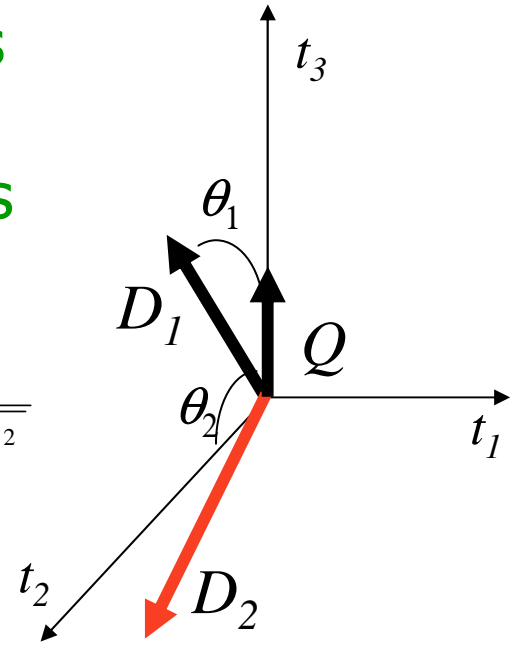
$D_1 = 2T_1 + 3T_2 + 5T_3 \qquad D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$

$\text{sim}(D_1 , Q) = 2*0 + 3*0 + 5*2 = 10$
$\text{sim}(D_2 , Q) = 3*0 + 7*0 + 1*2 = 2$

Ähnlichkeit mittels Kosinus

- Kosinus-Ähnlichkeit mißt den Kosinus des Winkels zwischen zwei Vektoren.
- Entspricht dem inneren Produkt, dass via Vektorenlänge normalisiert wird.

$$\text{CosSim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$



$D_1 = 2T_1 + 3T_2 + 5T_3$ $\quad \text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$

$D_2 = 3T_1 + 7T_2 + 1T_3$ $\quad \text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$

$Q = 0T_1 + 0T_2 + 2T_3$

D_1 ist 6mal besser als D_2 bei Verwendung der Kosinusähnlichkeit, aber 5mal besser bei Verwendung des inneren Produktes.

Naive Implementation

- Konvertiere alle Dokumente in der Sammlung D nach tf-idf gewichteten Vektoren $\boldsymbol{d_j}$, mit Schlüsselwort-vokabular V.
- Konvertiere die Anfrage nach tf-idf gewichtetem Vektor $\boldsymbol{q}$.
- Für jedes $\boldsymbol{d_j}$ in D tue:
  - Berechne Punktzahl $s_j = \text{cosSim}(\boldsymbol{d_j}, \boldsymbol{q})$
- Sortiere die Dokumente nach absteigender Punktzahl.
- Präsentiere das oberste Dokumente als Resultat dem Benutzer.

- Zeitkomplexität: $O(|V| \cdot |D|)$ Schlecht für große V & D !
- $|V| = 10.000$; $|D| = 100.000$; $|V| \cdot |D| = 1.000.000.000$

Kommentare zum Vektorraummodell

- Einfache, mathematisch fundierte Methode.
- Berücksichtigt lokale (*tf*) als auch globale (*idf*) Frequenzen von Wortvorkommen.
- Unterstütz partiellen Mustervergleuch und sortierte Resultate.
- Scheint in der Praxis sehr gut zu arbeiten trotz einiger offensichtlichen Schwächen.
- Erlaubt effiziente Implementationen für sehr große Dokumentenkolletionen.

Problem mit dem Vektorraummodell

- Fehlende semantische Information (z. B. Wortbedeutungen).
- Fehlende syntaktische Information (z. B. Phrasenstruktur, Wortstellung, Distanzinformation).
- Annahme, dass Terme unabhängig sind (z. B. ignoriert Synonyme).
- Verfügt nicht über die Kontrolle eines Booleschen Modells (z. B. , *erzwingen,* dass ein Term in einem Dokument vorkommen muß).
  - Gegeben eine Anfrage mit zwei Termen "A B", bevorzugt eventuell ein Dokument, dass A sehr oft, B aber garnicht enthält, gegenüber einem Dokument, das beide Terme A und B enthät, aber mit jeweils geringerer Frequenz.

Die folgenden Folien sind nicht mehr in der Vorlesung besprochen worden ...

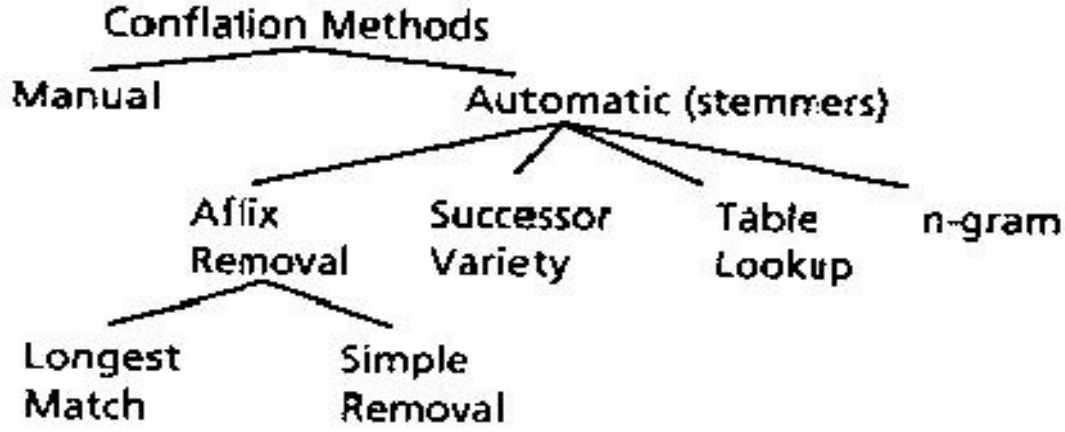
- ... aber reinschauen tut ja nicht weh !

Generierung von Merkmalen: Bag of Words (Wortkörbe)

- Textdokumente werden mit den Wörtern (und ihren Häufigkeiten), die es enthält, dargestellt
  - Z. B. “Lord of the rings” $\rightarrow$ {“the”, “Lord”, “rings”, “of”}
  - Sehr effizient
  - Macht das Lernen sehr viel einfacher
  - Die Wortstellung ist für viele Anwendungen nucht wichtig
- Berechnung von Wortstämmen/Stemming
  - Reduziert die Dimensionalität (Größe der Dokumentenmatrix)
  - Identifiziert ein Wort durch dessen Stamm
  - Z. B. flying, flew $\rightarrow$ fly
- Stoppwörter
  - Bestimmt die allgemeinsten Wörter, die statistisch gesehen, keine Signifikanz haben.
  - Z. B. “the”, “a”, “an”, “you”

Stemming

- Stemming ist eine Technik mit der morphologische Varaianten eines Schlüsselwortes bestimmt werden können.
- Wird verwendet, um die Effektivität des Abrufens von Dokumenten zu verbessern und um die Größe der Indexdateien zu verringern.
- Klassifikation von Stemming-Algorithmen



Stemming (fortgesetzt)

- Kriterien zur Beurteilung von Stemmers
  - Korrektheit
    - Overstemming: es wird zu viel von einem Term entfernt.
    - Understemming: es wird zu wenig von einem Term entfernt.
  - Effektivität für das Abrufen von Dokumenten
    - wird via Recall und Präzision gemessen und betrachtete auch Zeit und Platzkomplexität etc.
  - Kompressionsperformanz
  - Sprachunabhängigkeit
    - Für alle/Nur für wenige Sprachen ohne Aufwand einsetzbar

Typen von Stemming-Algorithmen

- Methoden, die in Tabellen nachschlagen
  - Hash-Arrays
  - Buchstabenbäume (Tries)
- Successor Variety
  - zur Bestimmung der Morpheme eines Terms
  - Untersucht die Verteilung von Nachfolgezuständen von Morphemen
- N-Gramm Stemmers
  - Statistik= {st, ta, at, ti, is, st, ti, ik}
  - statistisch= {st, ta, at, ti, is, st, ti, is, sc, ch}

$$sim(w_1, w_2) = \frac{2 * \sum eqngram(w_1, w_2)}{|ngram(w_1)| * |ngram(w_2)|}$$

- Affix Removal Stemmers

Porter Stemmer

- Einfache Methode, um bekannte Affixe einer Sprache zu entfernen ohne Verwendung eines Lexikons.
- Kann eventuell Stämme erzeugen, die keinen Wörtern entsprechen:
  - "computer", "computational", "computation" werden alle reduiziert zu "comput"
- Kann eventuell unterschiedliche Wörter auf denselben Stamm reduzieren.
- Kann nicht alle morphologischen Varianten erkennen.

Porter Stemmer Fehler

- Fehler durch "Abzug":
  - organization, organ → organ
  - police, policy → polic
  - arm, army → arm
- Fehler durch "Versäumnis":
  - cylinder, cylindrical
  - create, creation
  - Europe, European

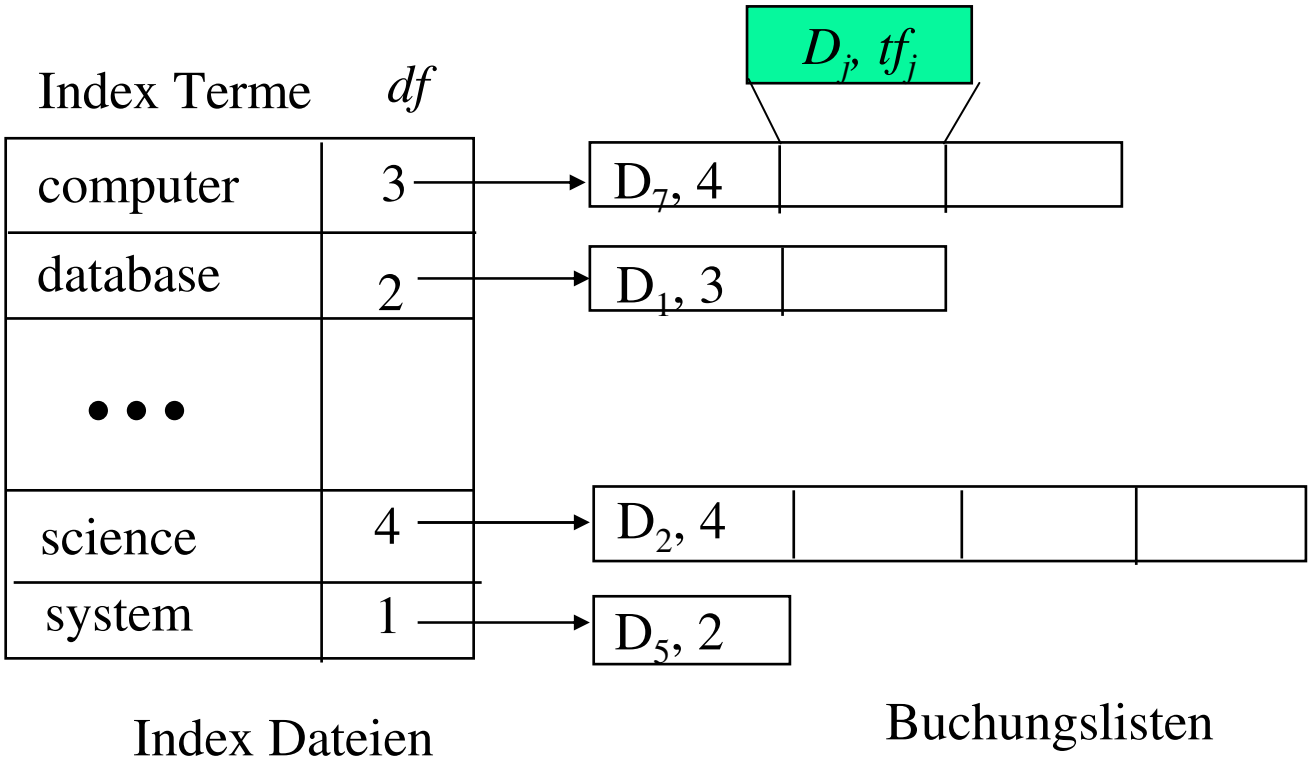
Karge Vektoren

- Vokabular und damit die Dimensionalität von Vektoren kann sehr groß werden, $\sim 10^4$.
- Allerdings enthalten die meisten Dokumente und Anfragen nicht viele der weit verbreiteten Wörter, sodass die Vektoren in der Regel sehr karg sind (also viele Nullen enthalten).
- Daher werden effiziente Methoden zur Speicherung und Berechnung von kargen Vektoren benötigt.

Implementationen, die auf invertieren Dateien basieren

- In der Praxis werden Dokumentenvektoren nicht direkt gespeichert; eine invertierte Organisation bietet eine viel bessere Effizient.
- Der Schlüsselwort-zu-Dokument Index kann mit Hash-Tabellen, sortierten Feldern oder einer baumbasierten Datenstruktur (trie, B-tree) implementiert werden.
- Kritisch ist ein logarithmischer oder zeit-konstanter Zugriff auf die Information von Termen.

Invertierter Index

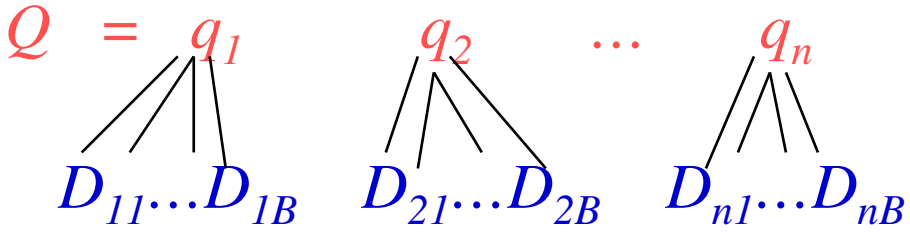


Abrufen mit invertieren Index

- Terme, die nicht in der Anfrage und dem Dokument sind, haben keinen Einfluss auf die Kosinusähnlichkeit.
  - Das Produkt der Termgewichte ist Null und trägt daher nicht zum inneren Produkt bei.
- In der Regel ist eine Abfrage sehr kurz und damit der entsprechende Vektor sehr karg.
- Benutze den invertierten Index um eine kleine Menge von Dokumenten zu bestimmen, die zumindest eines der Schlüsselwörter enthält.

Effizienz der Invertierte Anbfrage-Abruf

- Nimm an das durchschnittlich eine Schlüsselwort in *B* Dokumenten vorkommt:



- Dann ist die Abrufzeit O(|*Q*| *B*), was typischerweise sehr viel besser ist, als das naive Abrufen, welches all *N* Dokumente untersucht:

  – O(|*И*| *N*), weil |*Q*| << |*И*| and *B* << *N*.

Evaluation

Was soll evaluiert werden?

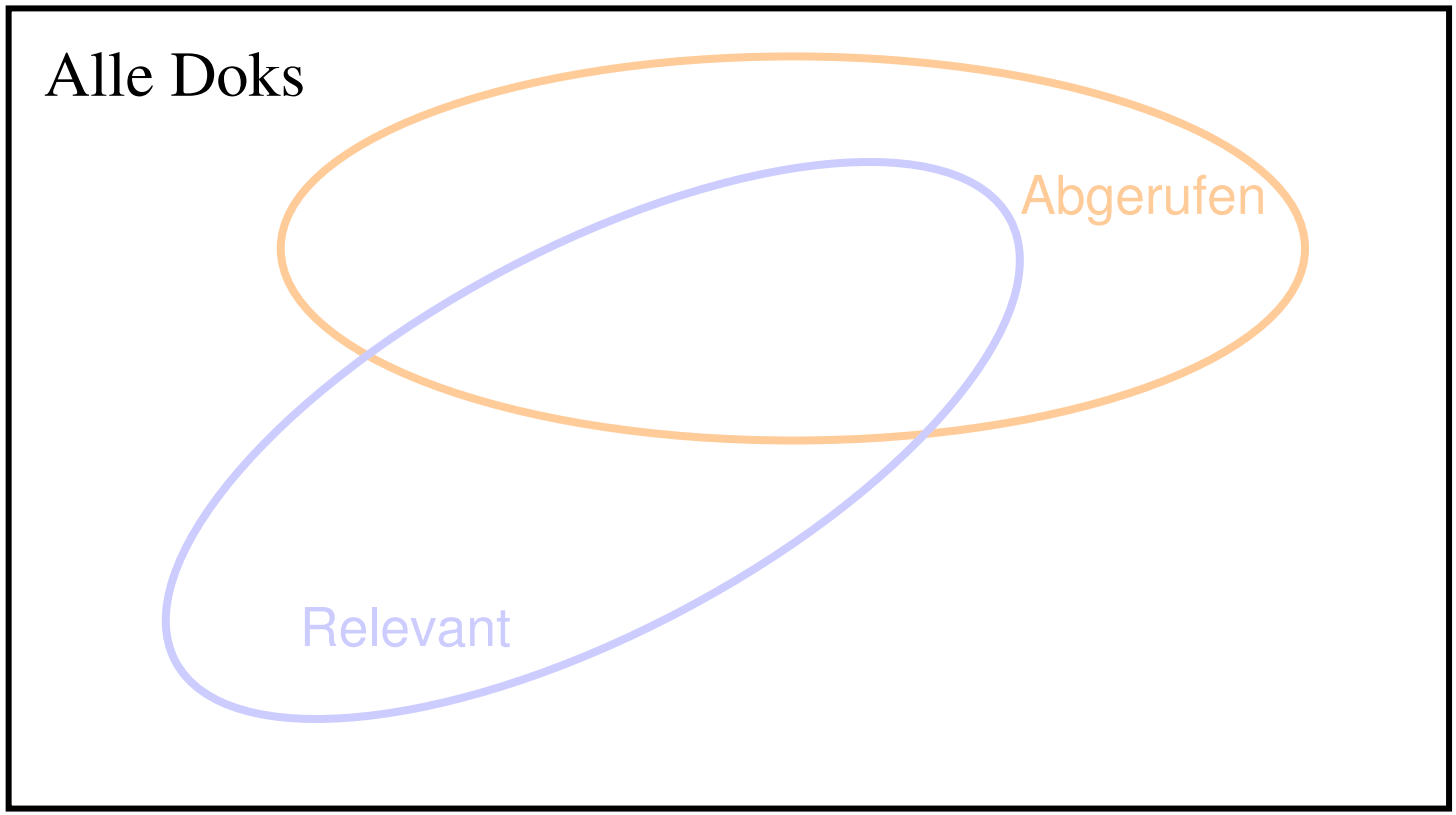
Was kann gemessen werden, was Aufschluß über die Fähigkeit eines Benutzers gibt, ein System benutzen zu können? (Cleverdon 66)

- Abdeckung der Information
- Form der Präsentation
- Erforderlicher Aufwand/Einfachheit der Verwendung
- Zeit- und Platzkomplexität

Effektivtät

- Recall (Abruf)
  - Das Verhältnis von relevantem Material zu tatsächlich abgerufenem Material
- Precision (Präzision)
  - Das Verhältnis von abgerufenem Material zu aktuell relevantem Material

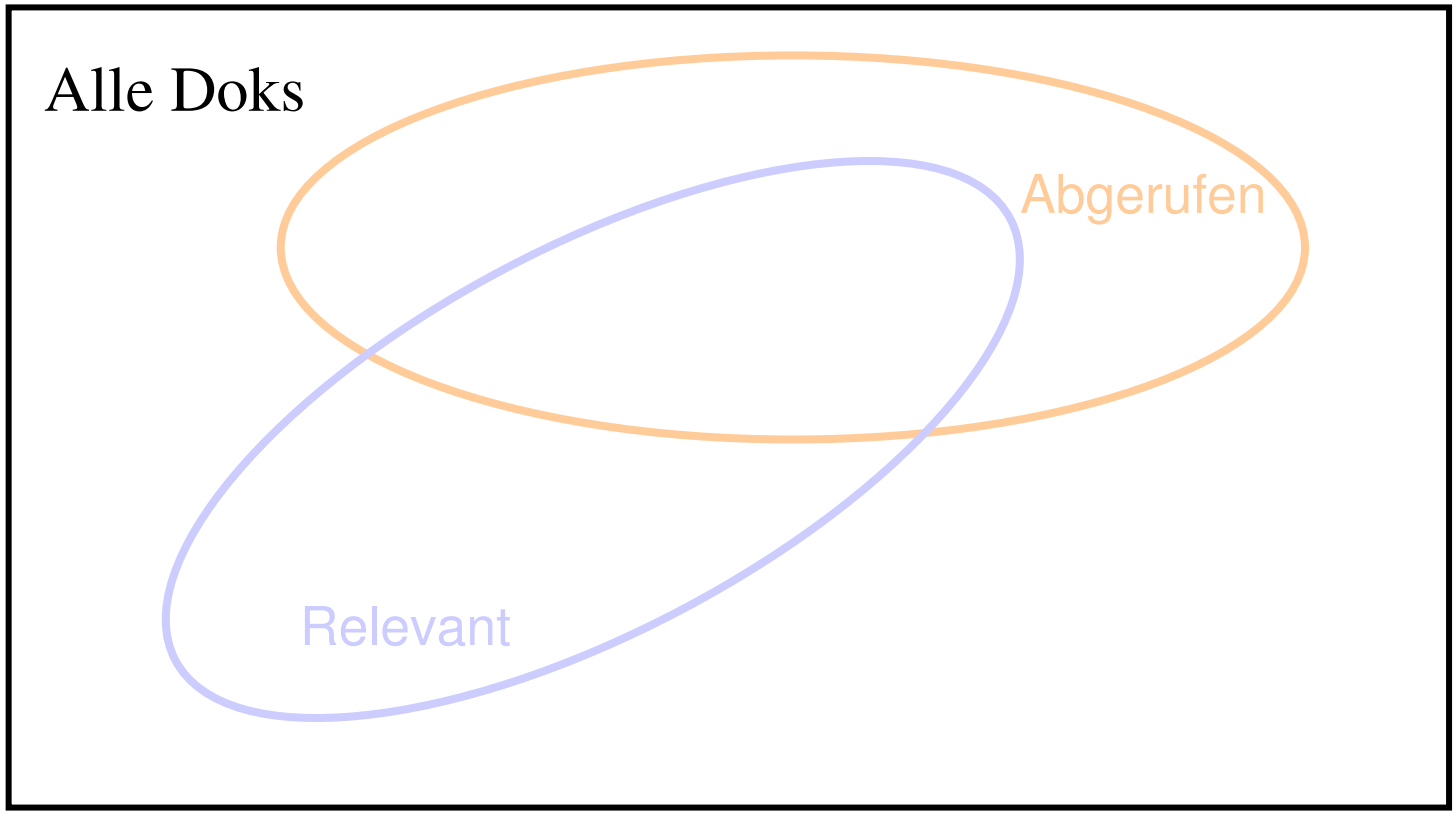
Relevant vs. Abruf



Precision vs. Recall

$$\text{Precision} = \frac{|\,\text{RelAbgerufen}\,|}{|\,\text{Abgerufen}\,|}$$

$$\text{Recall} = \frac{|\,\text{RelAbgeruf en}\,|}{|\,\text{Rel in der Kollektion}\,|}$$

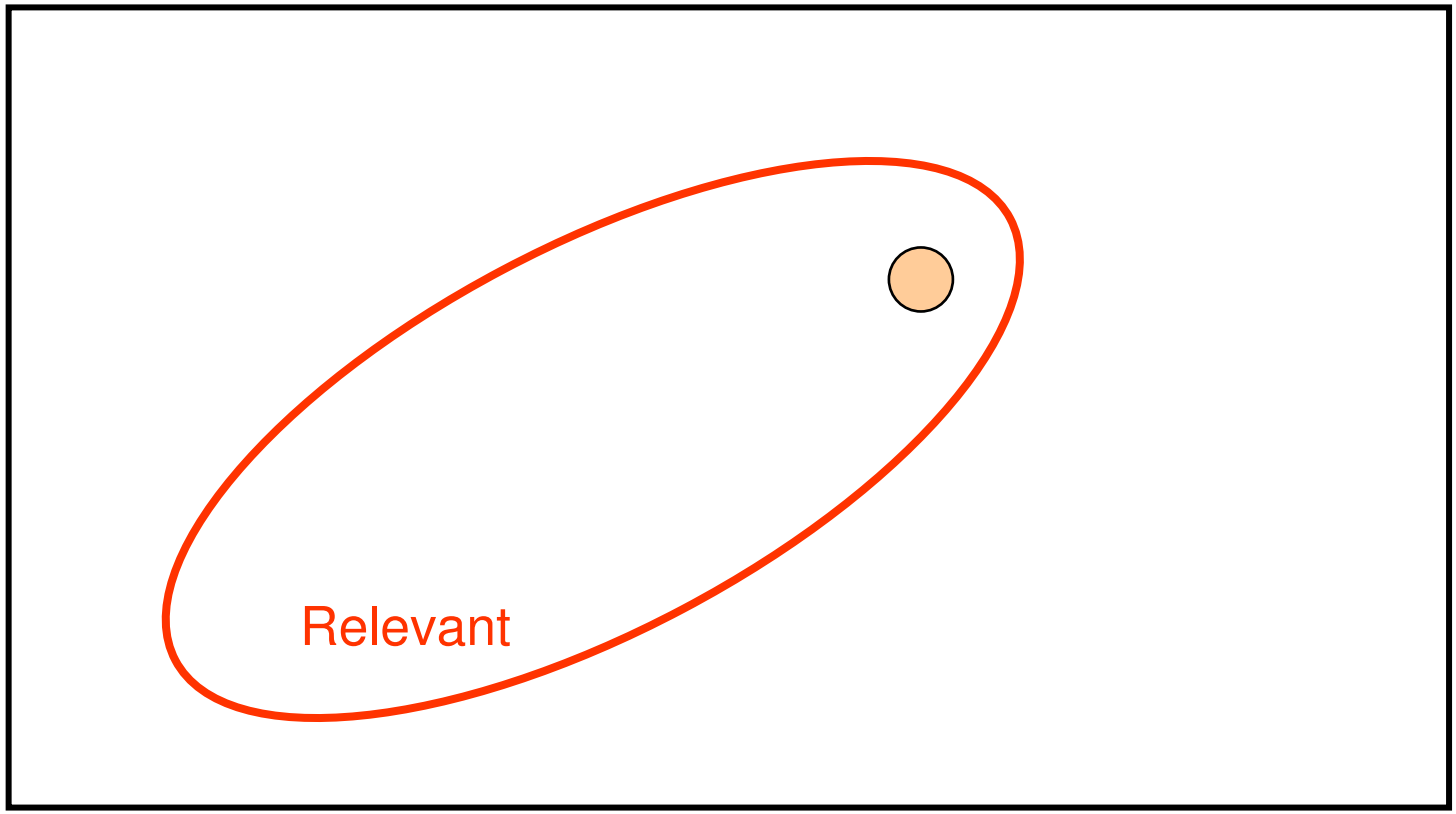


Warum Precision und Recall?

Erhalte so viel gutes Material und gleichzeitig so wenig schlechtes Material wie möglich.

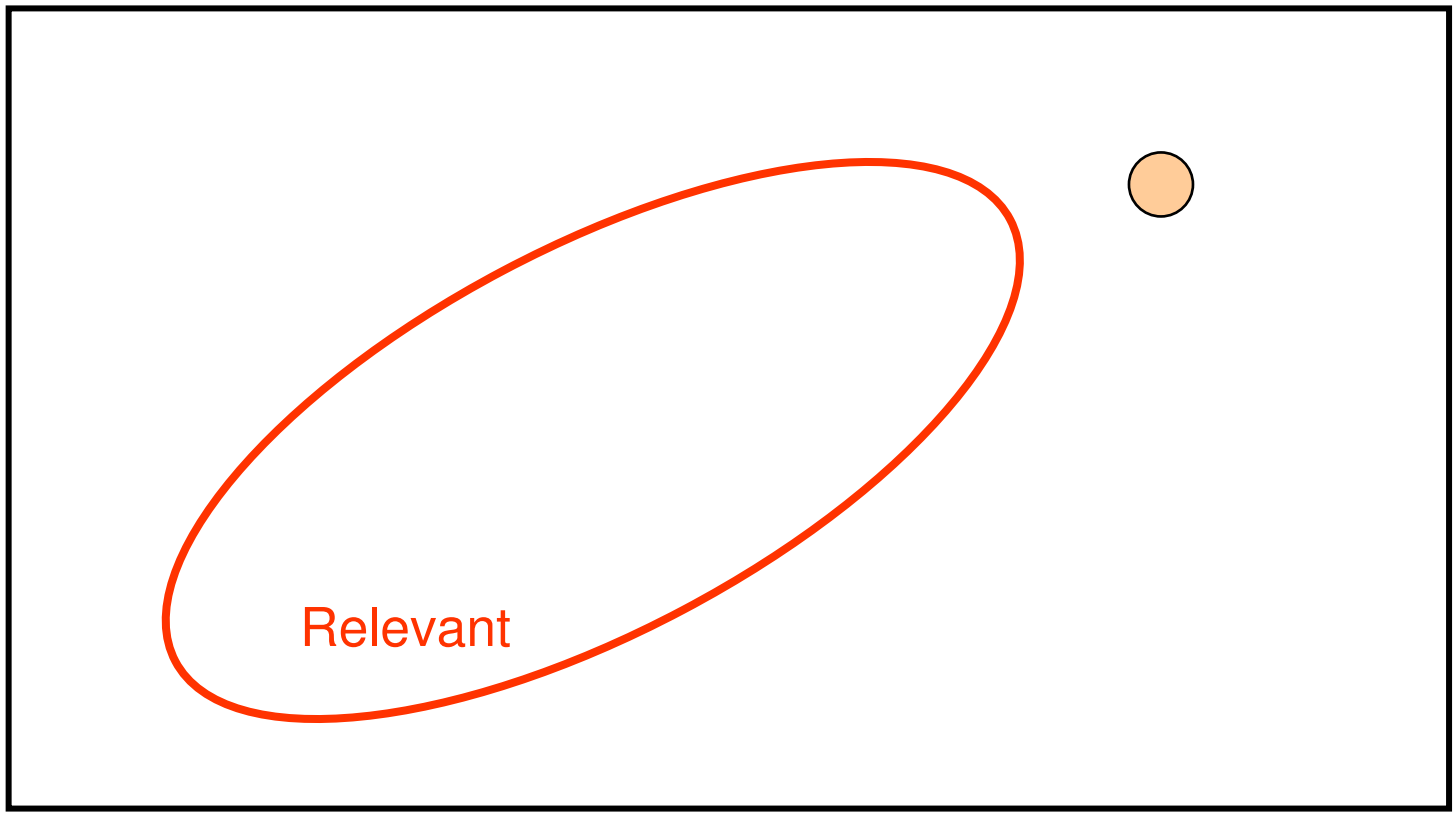
Abgerufene vs. relevante Dokumente

Sehr hohe Präzision, aber sehr geringer Recall



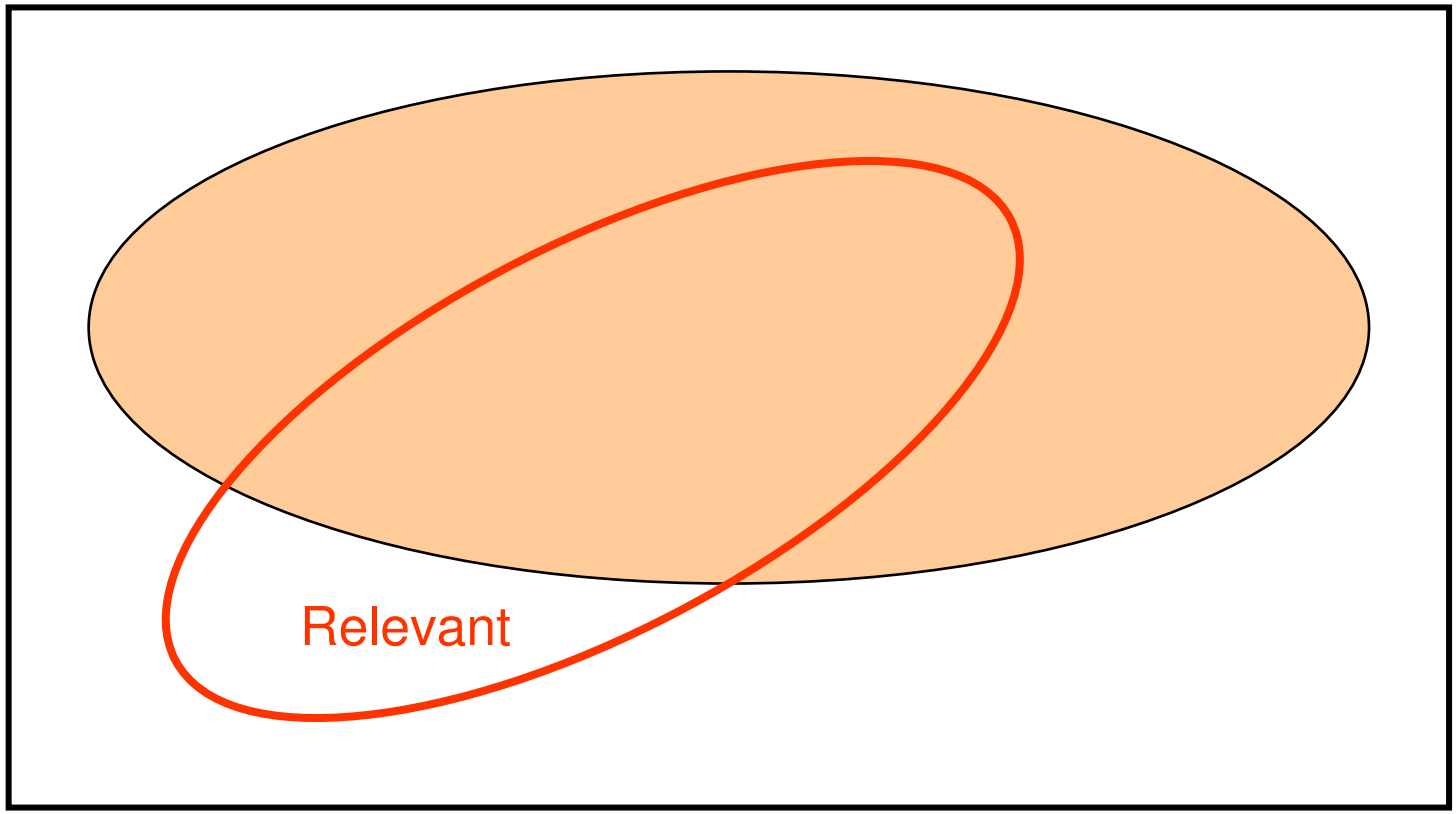
Abgerufene vs. relevante Dokumente

Sehr geringe Präzision, sehr geringer Recall (in der Tat 0)



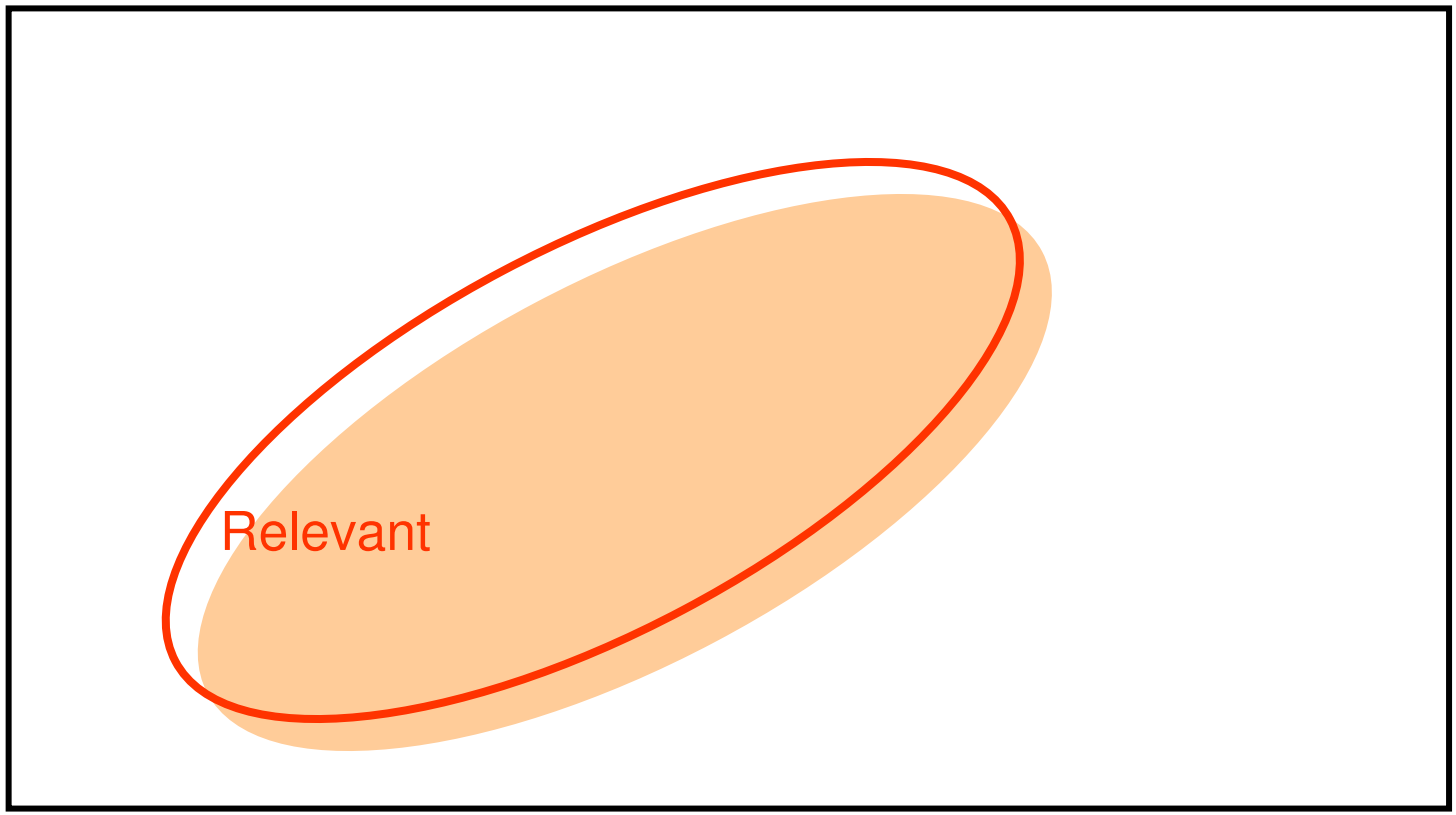
Abgerufene vs. relevante Dokumente

Hoher Recall, aber geringe Präzision



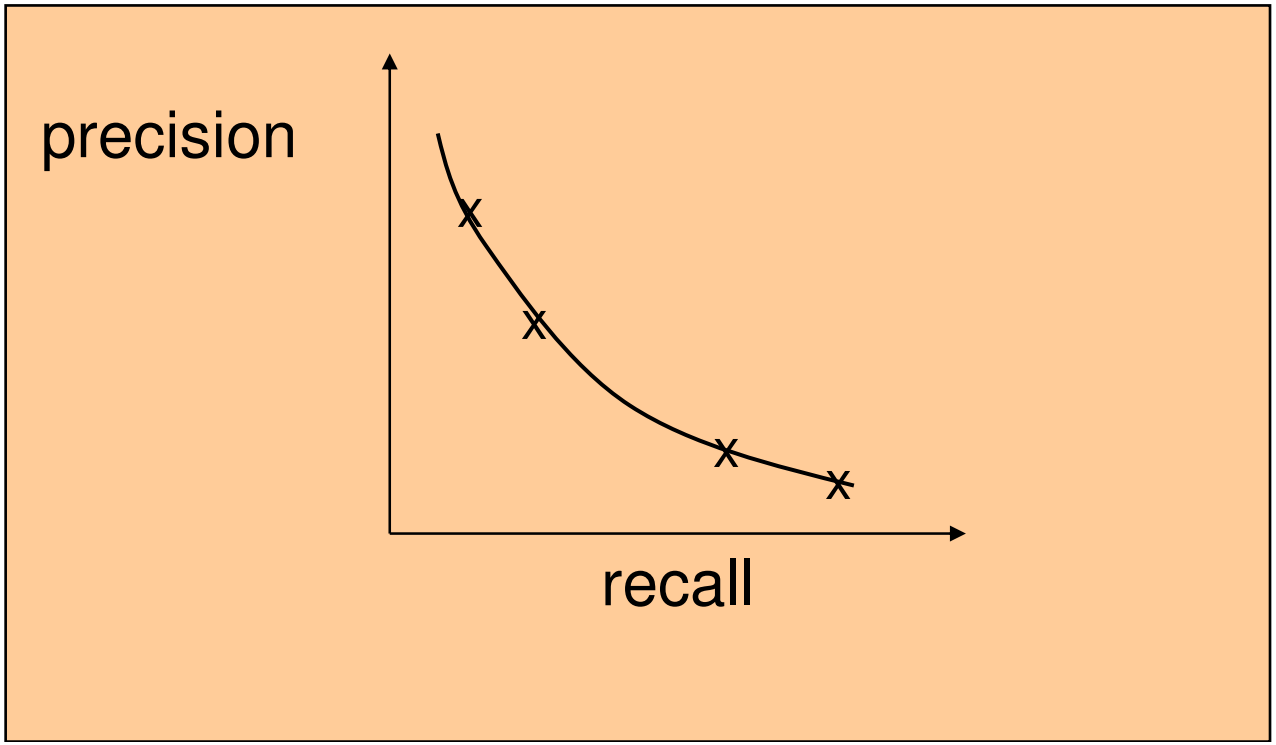
Abgerufene vs. relevante Dokumente

Hohe Präzision, hoher Recall (endlich!)



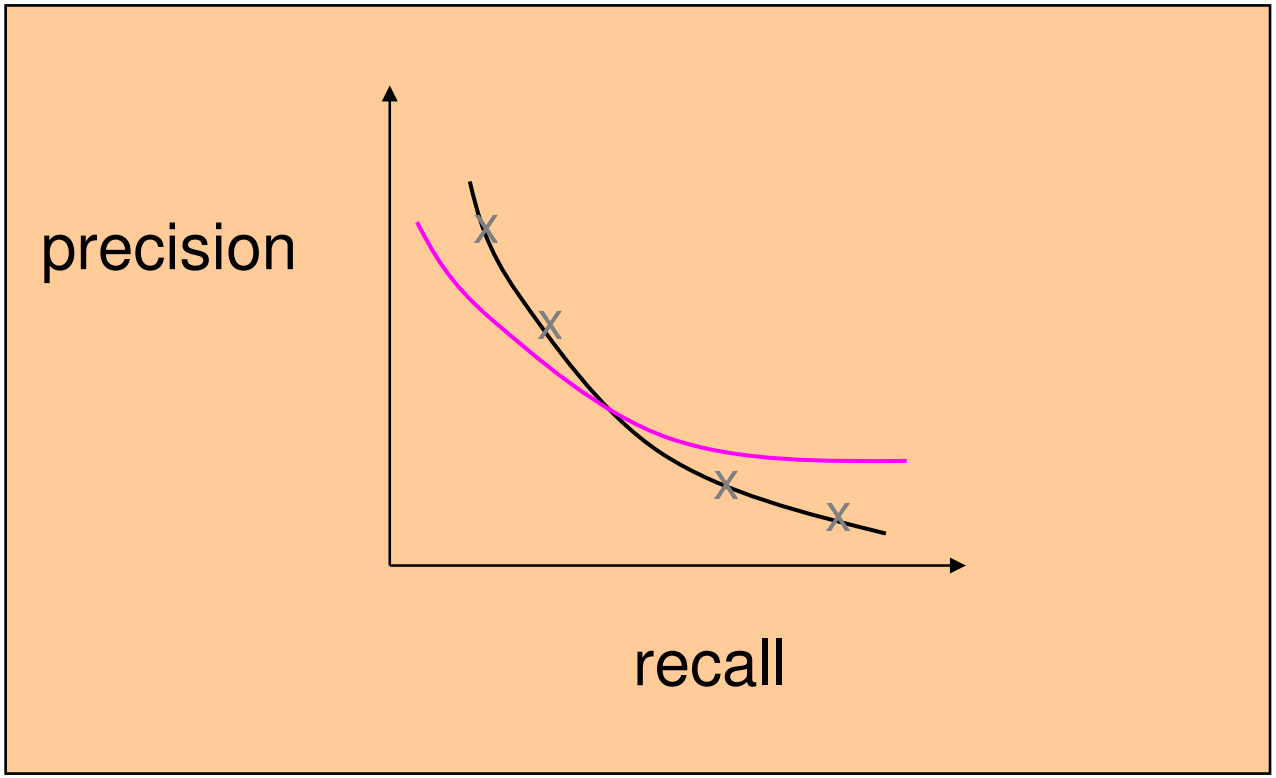
Precision/Recall Kurven

- Es gibt eine Wechselwirkung zwischen Precision und Recall
- Meße daher Precision mit verschiedenen Graden von Recall.
- Beachte: dies ist ein DURCHSCHNITT über VIELE Anfragen

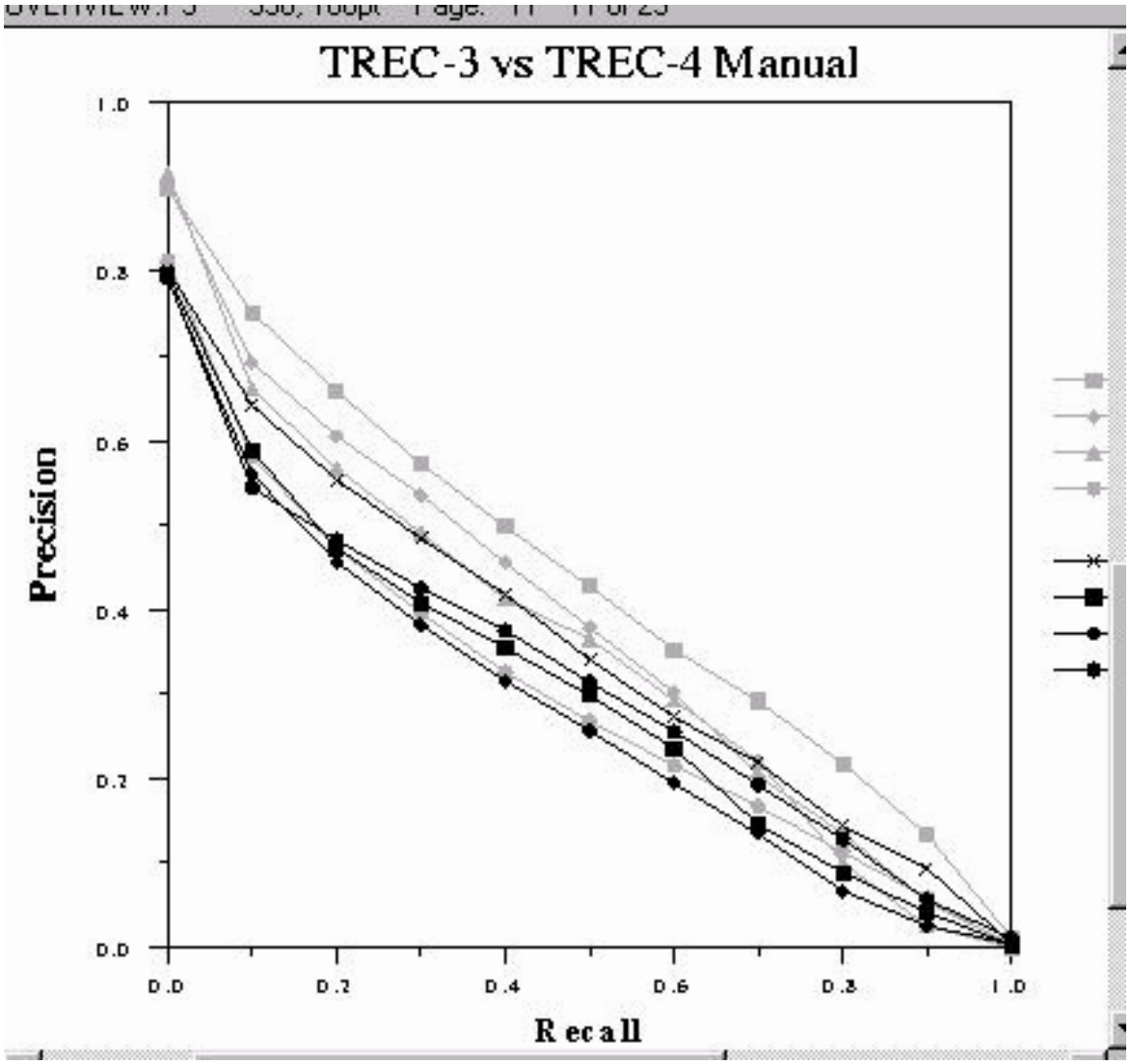


Precision/Recall Kurven

- Schwierig zu sagen, welches dieser zwei hypothestischen Ergebnisse besser ist.



Precision/Recall Curves



Größe der Ergebnisliste

- Eine andere Strategie zur Evaluation:
  - Fixiere die Anzahl der abgerufenen Dokumente Fix für verschiedene Testreihen
    - top 5
    - top 10
    - top 20
    - top 50
    - top 100
    - top 500
  - Messe Precison für jede Testreihe
  - Bestimme den (gewichteten) Durchschnitt über alle Ergebnisse
- Dies ist eine Möglichkeit zu untersuchen, wie gut ein System ist, wenn es die k-ersten Dokumente liefert.

Probleme mit Precision/Recall

- Kaum möglich, die wirkliche Recall-Menge zu kennen
  - Praktisch nur für kleine Kollektionen möglich
- Precision/Recall sind verwandt
  - Ein kombiniertes Maß erscheint manchmal angemessener
- Nimmt einen Batch-Modus an:
  - Interaktive IR ist ebenfalls wichtig und besitzt andere Kriterien für eine erfolgreiche Suche
- Nimmt an, dass eine strikte Sortierung selbstverständlich ist.

Bezug mit Verteilungstabelle

	Dok ist Relevant	Dok ist nicht relevant
Dok ist abgerufen	a	b
Dok ist nicht abgerufen	c	d

- Akkuratheit: (a+d) / (a+b+c+d)
- Precision: a/(a+b)
- Recall: ?
- Warum wird denn nicht Akkuratheit für IR verwendet?
  - (Annahme einer sehr großen Kollektion)
  - Die meisten Dokumente sind nicht relevant
  - Die meisten Dokumente werden nicht abgerufen
  - Treibt den Wert der Akkuratheit hoch

F-Maß

- Ein Performanzmaß, dass Precision und Recall in betracht zieht.
- Harmonischer Mittelwert von Recall und Precision:

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R}+\frac{1}{P}}$$

$$\bar{x}_{\text{harm}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Beispiel für das harmonische Mittel von 5 und 20:

$$\frac{2 \cdot 5 \cdot 20}{5+20} = 8$$

oder

$$\frac{2}{\frac{1}{5}+\frac{1}{20}} = \frac{2}{\frac{1}{4}} = 8$$

- Verglichen mit dem arithmetischen Mittel, müssen beide Werte hoch sein, damit auch das harmonische Mittel hoch ist.

$$\bar{x}_{arith} = \frac{\sum_{i=1}^{n} x_i}{n}$$

E-Maß (parametrisiertes F-Maß)

- Eine Variante des F-Masses, die eine gewichtete Betonung von Precision über Recall erlaubt.

$$E = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R}+\frac{1}{P}}$$

- Der Wert von β kontrolliert die Wechselwirkung:
  - β = 1: Gleiches Gewicht für Precision & Recall (E=F).
  - β > 1: Gewichte Recall mehr.
  - β < 1: Gewichte Precision mehr.

Aspekte einer Schlüsselwortbasierten Suche

- Informationsüberladung
- Mehrfache Vokabulare
- Synonymie
- Polysemie

Informationsüberladung

- Die Notwendigkeit von effektiven IR-Systemen wird immer wichtiger, da die computer-basierten Informationssammlungen immer größer und vielschichtiger werden.
- Informationsüberladung stellt sich dann ein, wenn die Benutzer von Systemen von der Anzahl der verfügbaren Information und dem kontinuierlichen hinzufügen von neuer Information überwältigt werden
- und weil die Benutzer normalerweise nicht genug Wissen über den Gegenstandsbereich und das System haben, was sie aber bräuchten, um die benötigte Information abrufen zu können.

Mehrfache Vokabulare

- Benutzer müssen auch mit mehrfachen Vokabularen umgehen können, die auf Grund der unterschiedlichen Hintergründe und Expertisen von Benutzern entstehen, die Information vom System abrufen.
- Dies führt oftmals zu einem geringen Recall, wobei die Datenmenge, in der der Benutzer sucht, zu groß oder zu klein ist.
- Dies führt wiederum dazu, das auch die Precision gerade für die Information, nach der der Benutzer sucht, zu gering sein kann.

Synonymie

- Synonymie behandelt die Gleichheit der Bedeutung von Wörtern, wobei verschiedene Wörter dasselbe Ding bedeutet order Synonyme sind.
- Ein Beispiel kann die Suche nach einem bestimmten Geschäftstyp in Gelben Seiten sein.

Polysemie

- Polysemie betrachtet das entgegengesetzte Problem, nämlich wo ein einzelnes Wort mehrere Bedeutung haben kann, abhängig vom Kontext, in dem es verwendet wird.

- Ein Beispiel könnte die Suche mit dem Wort „pen“ sein, was in Abhängigkeit des Kontextes, einmal ein Schreibgerät bezeichnet oder eine Strafanstalt.

Alternativen zu einer Schlüsselwortbasierten Suchmaschine

- Konzeptbasierte IR schaut auf die wachsenden Schwierigkeiten einer schlüsselwortbasierten Suche mit einer intuitiven Methode, mit der zuerst die Daten gemäß ihrer Beziehungen zueinander sortiert werden und dann die Daten nach spezifischer Information durchsucht wird.

- Durch das Sortieren der Daten zuerst gemäß ihrer Beziehungen zueinander, können wir einen guten Recall für die Menge von Daten erzielen, die wir auch wollen und durch Suche in diesen in Beziehung stehenden Daten, können wir eine hohe Precision erreichen.

- Die Forschung im Bereich der konzeptbasierten IR ist eine vitale Alternative, steckt aber noch in ihren Kinderschuhen.

Konzeptbasierte Suche: zwei verschiedene Felder

- Konzeptbasiertes IR basiert auf zwei verschiedenen Feldern: Taxonomien und Ontologien
- Taxonomie: Unterteilung in geordnete Gruppen oder Kategorien. Die Wissenschaft, die Gesetzmäßigkeiten order Prinzipien von Klassifikation.
- Ontologie: Der Bereich der Metaphysik, der die Natur der Dinge behandelt. Die *Relation zwischen Objekten* in der realen Welt.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Taxonomie behandelt die Klassifikation von Daten, wobei die verschiedenen Objekte in Abhängigkeit ihrer Charakteristiken zu verschiedenen Kategorien gehören.
- Ein Beispiel kann der Term *laptop* sein, der unter folgende die Kategorie gehört: computer→hardware→portable device.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Ontologien beschäftigen sich mit den Beziehungen zwischen Objekten der realen Welt, wie z. B. Bestandteile (“Teil-von” Beziehung, wie etwa in “ein Motor ist Teil eines Autos”) und Vererbung (“ist-ein” Beziehung, wie etwa „der Mensch ist ein Tier”).
- Ontologien beantworten die Frage „Welche Arten von Objekten existieren in einer oder mehrerer Domänen und wie stehen sie miteinander in Beziehung?“

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Zwei zentrale Bausteine von Ontologien sind *Typen* und *Rollen*, wobei Typen Instanzen von Objekten sind, deren Merkmale sich nicht ändern, wogegen Rollen Instanzen von Objekten sind, die sich unter umständen ändern können.
- Ein Beispiel für eine Typ wäre eine Pflanze, die bestimmte Merkmale darstellt, z. B. das sie ihr gesamtes Leben stets eine Pflanze sein wird.
- Eine Rolle wäre etwa ein Student an einer Universität, der selbst, nachdem er seinen Abschluss erhalten hat (und damit kein Student mehr ist), trotzdem ein Individuum bleibt.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Es ist wichtig, diese zwei Bereiche zu trennen, da Objekte, die zu einem bestimmen Konzept gehören, in Abhängigkeit der Modellierungsperspektive auf viele verschiedene Weisen klassifiziert werden können, wogegen das Objekt aus verschiedenen Blickwinkeln vom Benutzer betrachtet werden kann.
- Die Sichtweise entspricht der einer Rolle eines Objektes, sodass das da eine eins-zu-eins Abbildung zwischen dem Benutzer und der Funktion des Objektes ist.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Der Vorgehensweise, die Daten so zu sortieren, dass die Vorteile von Ontologie und Taxonomie gleichermaßen beachtet werden, ist die Erzeugung einer Konzeptabbildung der aktuellen Information.
- Eine Konzeptabbildung ist eine visuelle Wissensrepräsentation, die benutzt wird, um die Beziehung zwischen Ideen auszudrücken.
- Beispiele, wo Konzeptabbildungen Verwendung finden sind Brainstorming, Planung, Dokumentation, Präsentation und Software-Blaupausen.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Ein praktisches Beispiel der Konzeptmappingtechnik (zum Zwecke der Darstellung von Beziehungen zwischen Objekten) ist das semantische Netzwerk, wo die Knoten im gerichteten Graphen den Ideen entsprechen und die Kanten den Relationen zwischen den Ideen entsprechen.
- Die Konstruktion eines solchen konzeptuellen Netzwerkes von Ideen und Objekten gäbe dem Benutzer einen guten Recall für die abgerufene Information (basierend auf Ideen-Clustern) und bezüglich der Information, die an den Benutzer geliefert wird, könnte der Benutzer die Pfade im Netzwerk wählen, die Knoten größter Relevanz verbinden.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Aktuell gibt es eine Anstrengung die Daten im Web (per Definition und Verlinkung) so zu sortieren, sodass das es von Maschinen verwendet werden kann – nicht nur zur Darstellung, sondern auch für viele verschiedene Anwendungen.
- Der Name der Anstrengung ist "The Semantic Web" (http://www.semanticweb.org/)
- Dort werden Ontologien und Taxonomien als Problemslösungsmethoden untersucht.

Konzeptbasierte Suche: zwei verschiedene Felder (fortgesetzt)

- Ein anderes Beispiel für eine Konzeptabbildungstechnik in einer Suchmaschine ist das „Information Mapping Project at Stanford".
- http://www.csli.stanford.edu/semlab/infomap.html (Homepage)
- http://infomap.stanford.edu/webdemo (search engine)

Schluß

- Die Informationsmenge wird zunehmen, sodass Methoden exploriert werden müssen, die das Ernten der Daten in einer kohärenten Weise unterstützen, wo ein Benutzer seine relevante Information finden kann.
- Die Kombination der zwei Bereiche Ontologie und Taxonomie ist eine nützlicher Ansatz für eine konzeptbasiertes Model der IR.