SoSo 2018

Indexing und Klassifikation

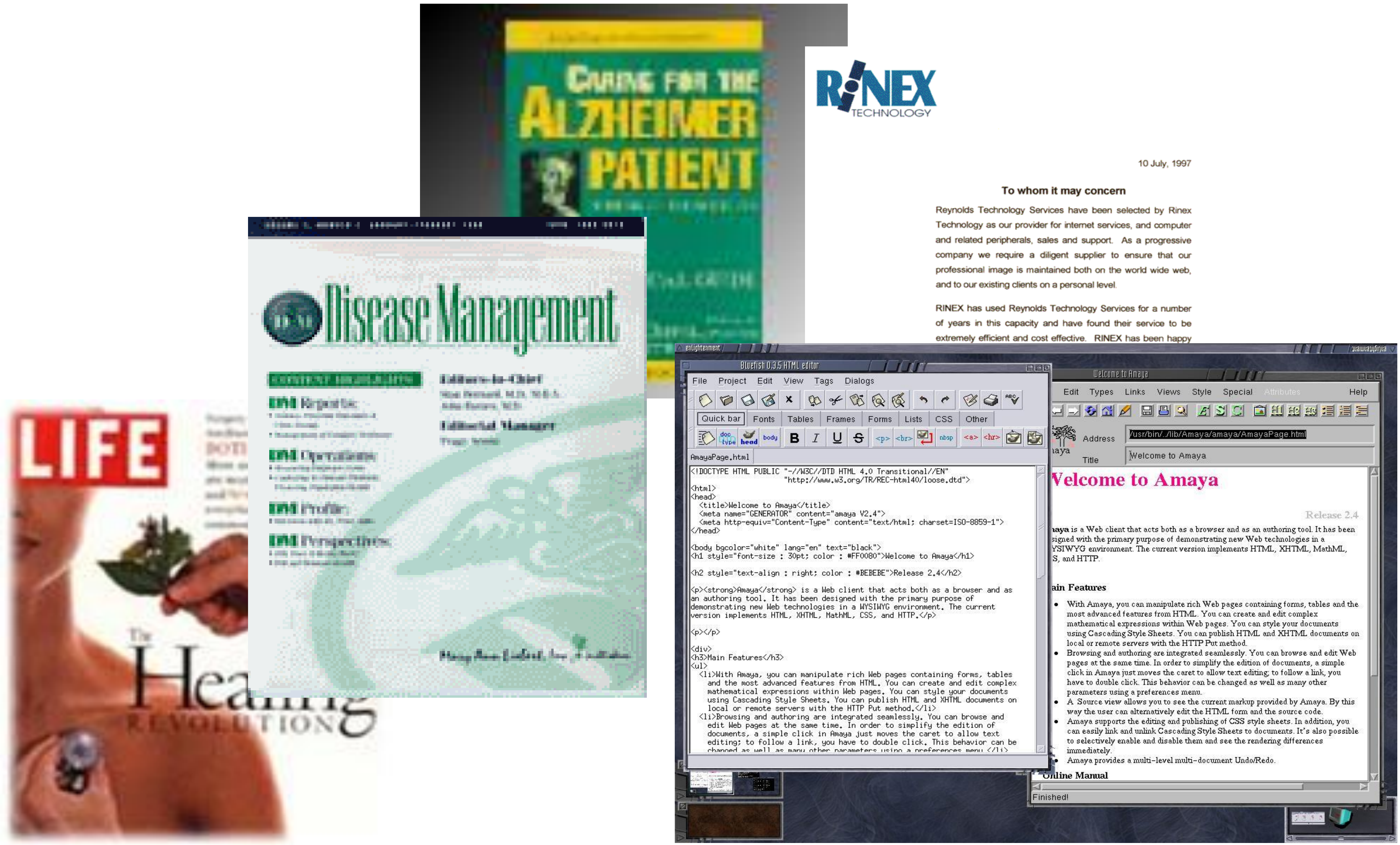
Udo Hahn

FRIEDRICH-SCHILLER-UNIVERSITÄT
JENA

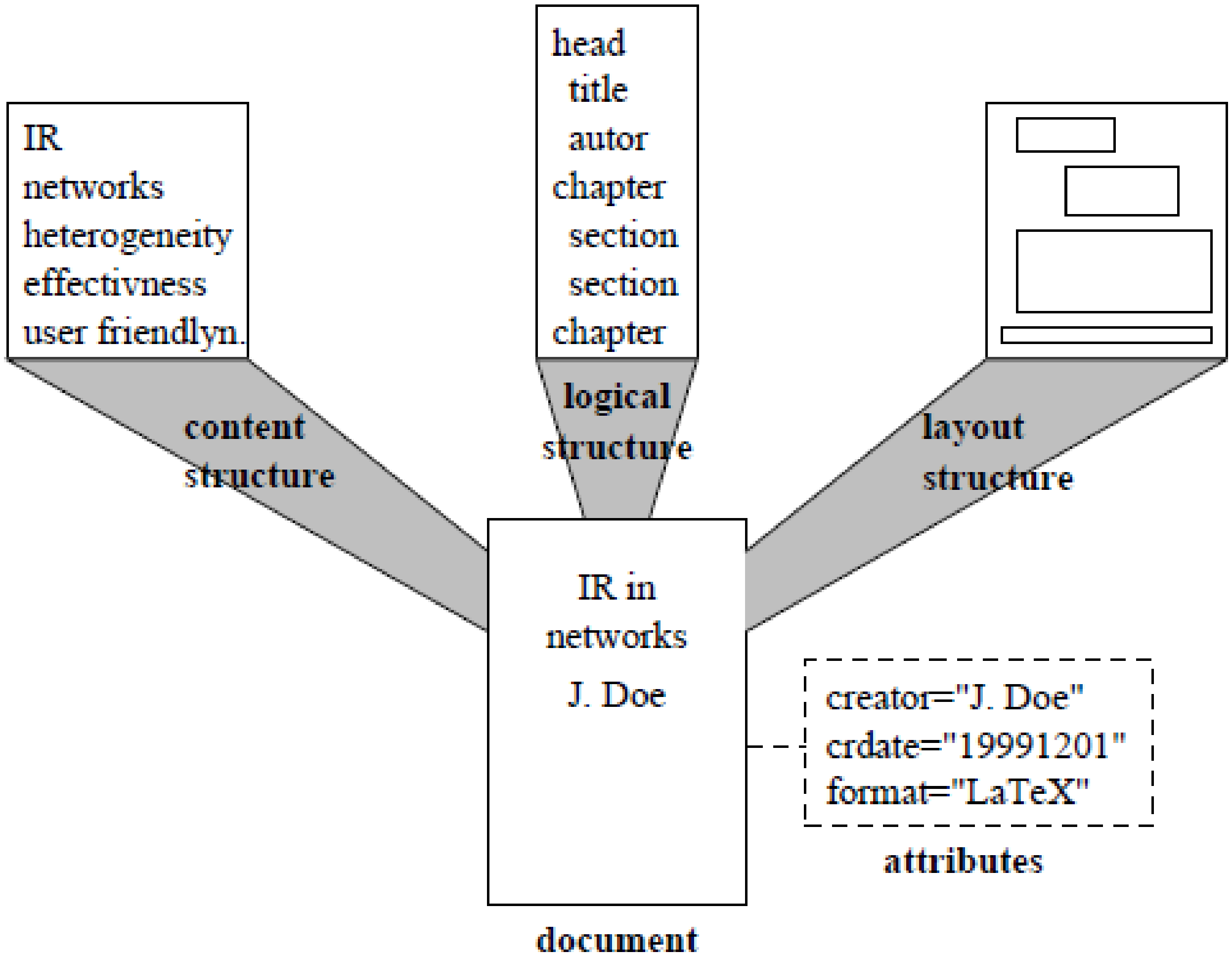
Jena University Language and Information Engineering (JULIE) Lab, Germany

www.julielab.de

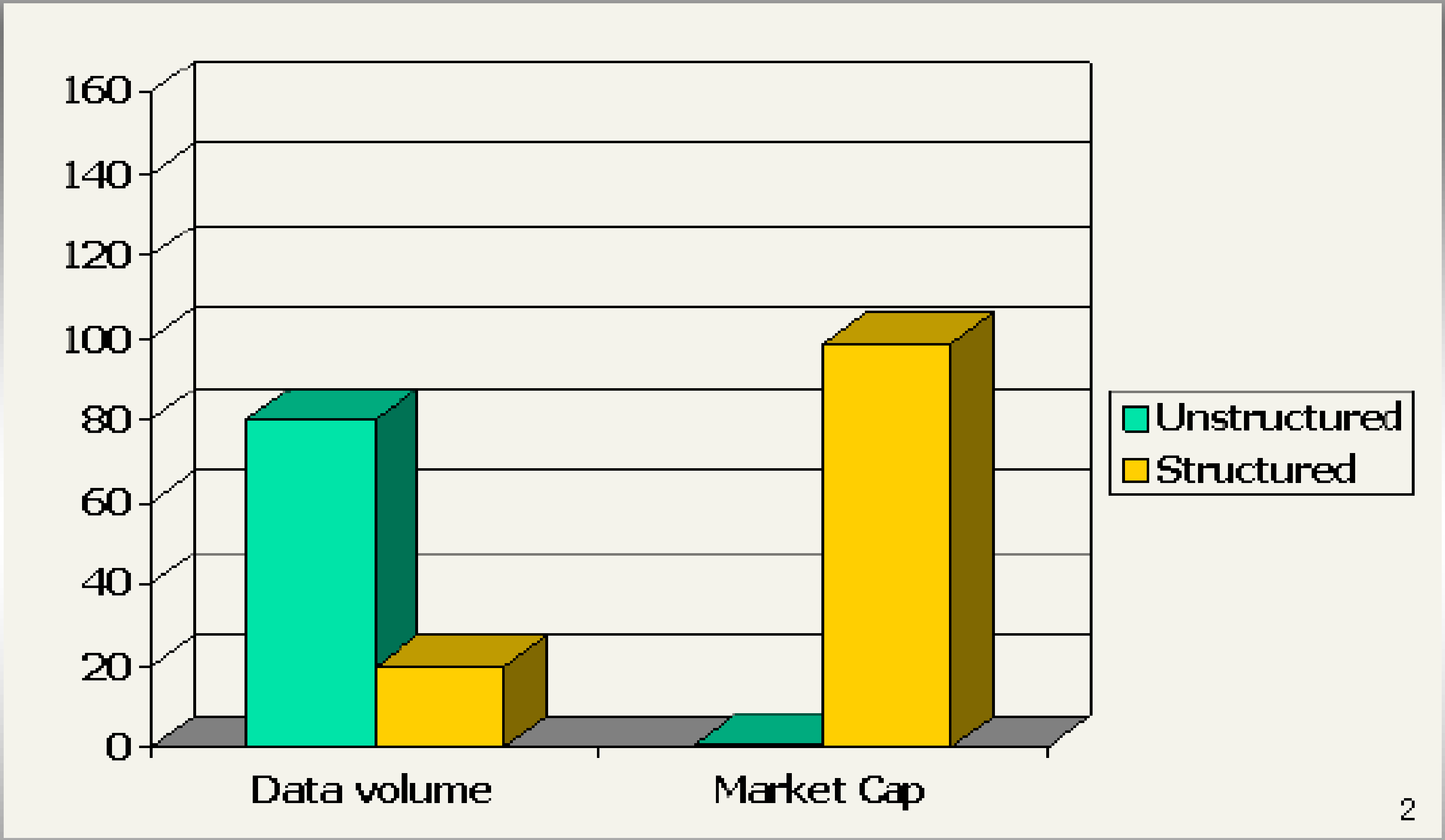
Document Content Technology



DIFFERENT VIEWS ON DOCUMENTS



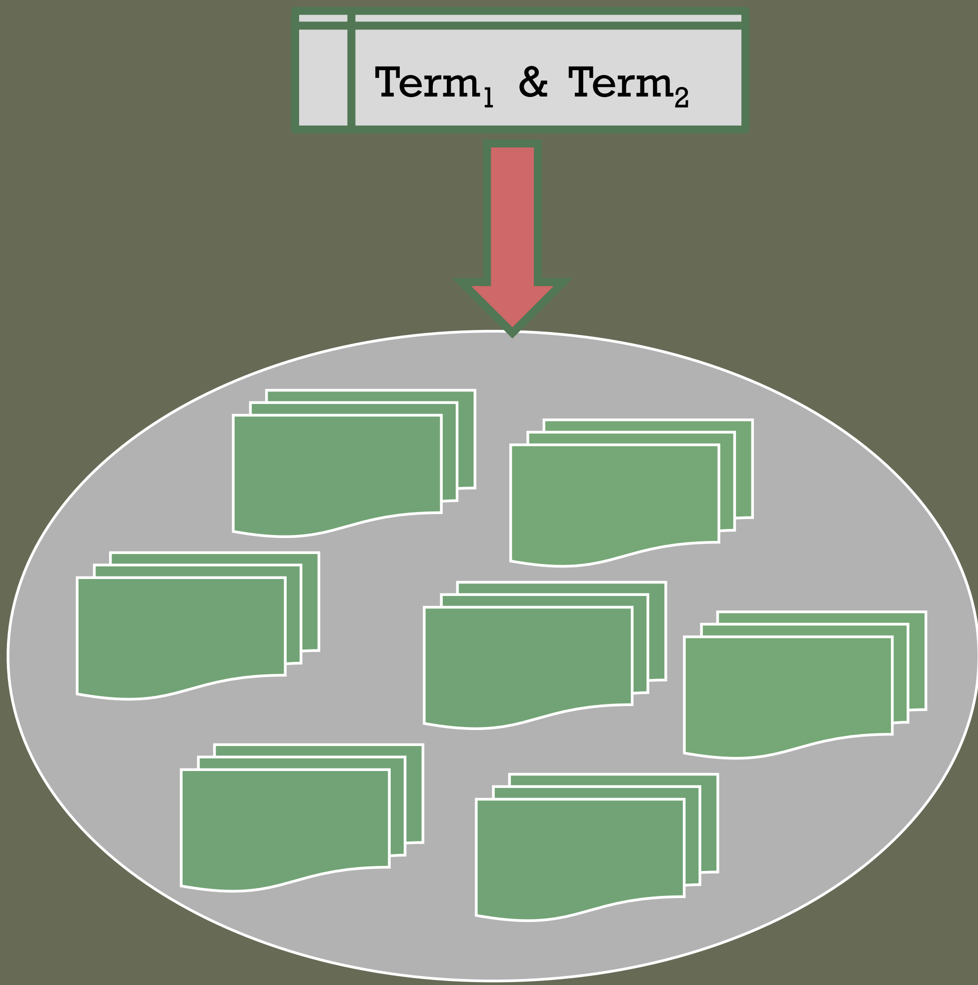
Structured vs. Unstructured Data (1996)



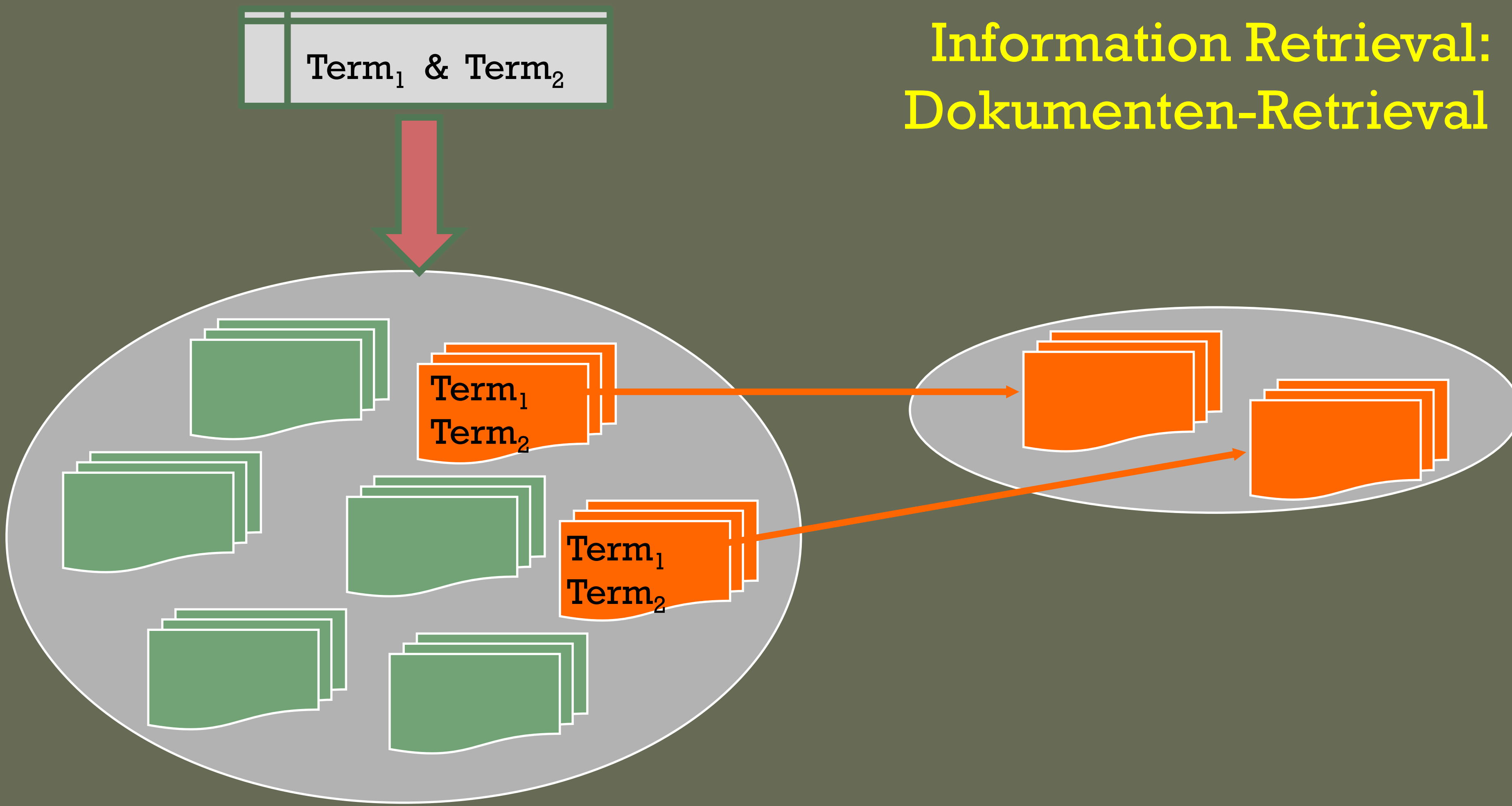
Source: Prabhakar & Raghavan, Verity (2002)

Structured vs. Unstructured Data (2006)

Information Retrieval: Dokumenten-Retrieval



Information Retrieval: Dokumenten-Retrieval



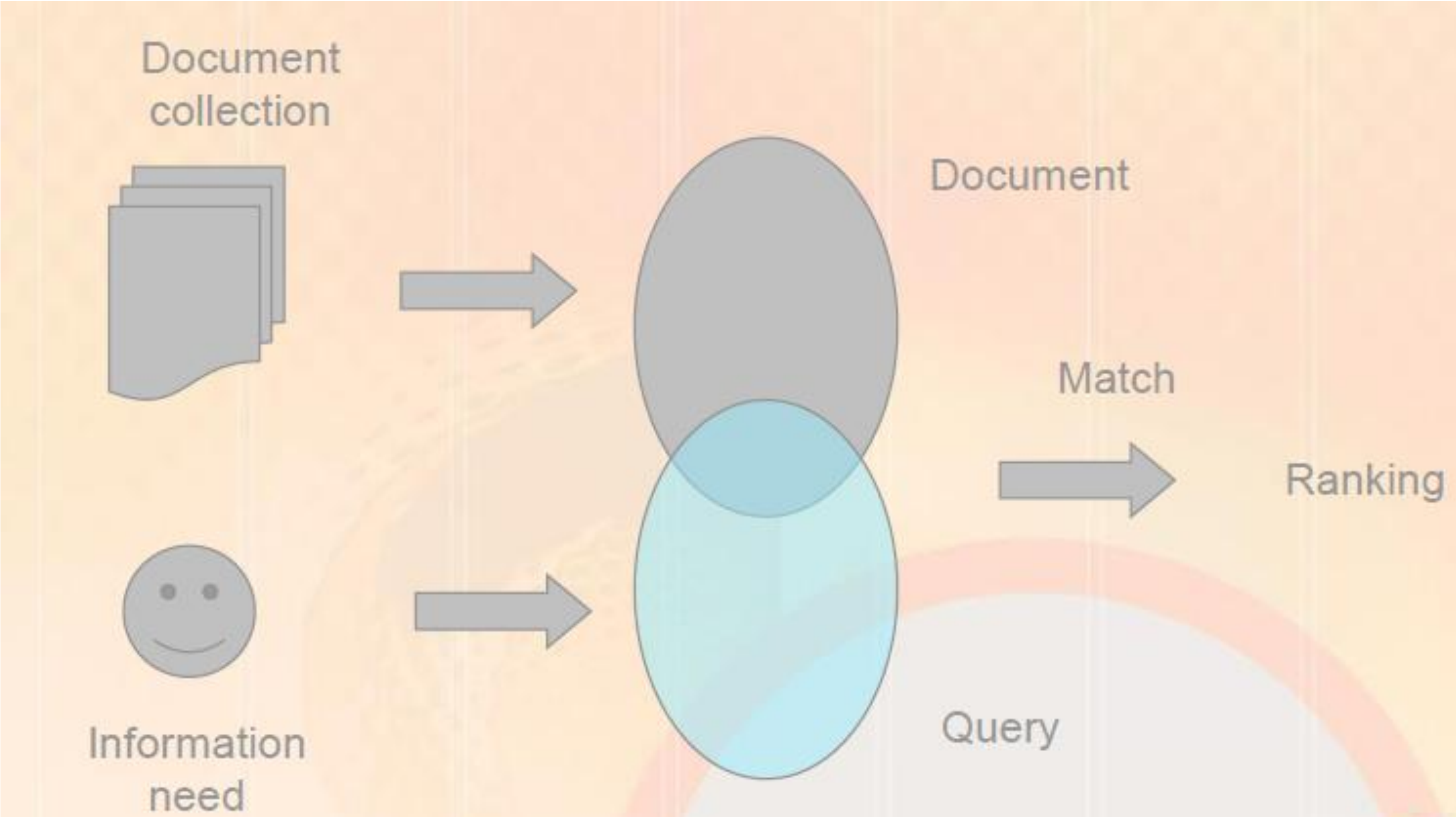
Textsuch-Modi

- Direkte Textsuche
  - Abgleich eines Anfrageterms direkt (partiell) mit einem Textterm in einem Dokument (Freitext-Retrieval)
  - Q: „history“ :: Doc „history“
    - Q: „histor$“ :: Doc1 „history“, Doc2 „historical“, Doc3 „histories“
- Metadaten-basierte Textsuche
  - Abgleich eines Anfrageterms vermittelt mit einem Konzept(-Identifier) eines Textterms
    - Q: „history“ :: Doc1 „HISTORY“ (←„history“), Doc2 „HISTORY“ (←„historical“), Doc3 „HISTORY“ (←„histories“) Doc4 „HISTORY“ (←„Geschichte“)

Boole‘sche Suche

- Einzel-Term-Suche
  - Abgleich eines Anfrageterms direkt (partiell) mit einem Textterm in einem Dokument (Freitext-Retrieval)
    - Q: „history“ :: Doc „history“
    - Q: „histor$“ :: Doc1 „history“,
Doc2 „historical“,
Doc3 „histories“
- Multi-Term-Suche
  - Abgleich von Anfragetermen direkt (partiell) mit Texttermen in einem Dokument unter Verwendung einfacher logischer Ko-okkurrenz-Bedingungen
    - Q: „history“ AND „music“ :: Doc „history“ … „music“
    - Q: „history“ OR „music“ :: Doc1 „music“, Doc2 „history“
    - Q: „history“ NOT „music“ :: Doc „history“

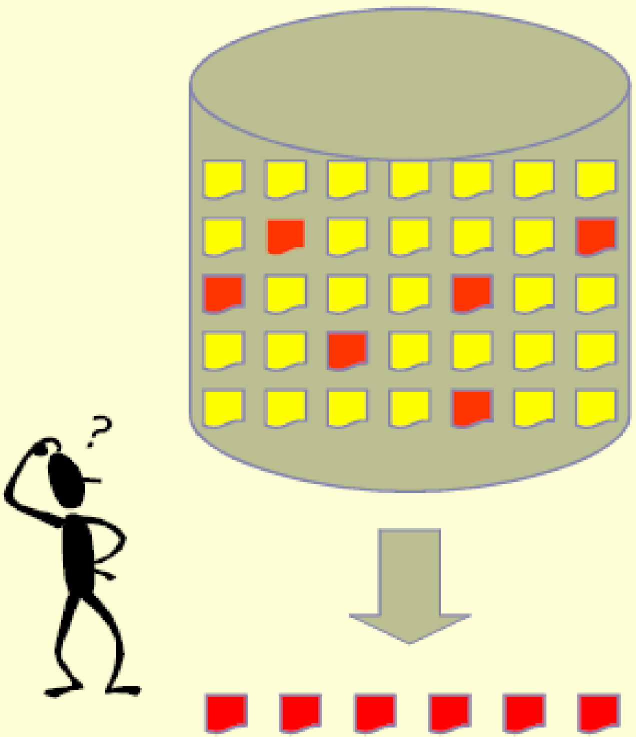
Basic Model of Information (Document) Retrieval



Flavors of Information (Document) Retrieval (1/2)

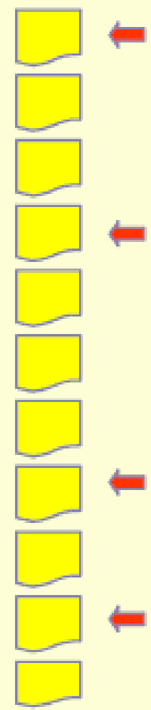
Ad-hoc retrieval

One time queries (e.g. Web search)



Filtering/Routing

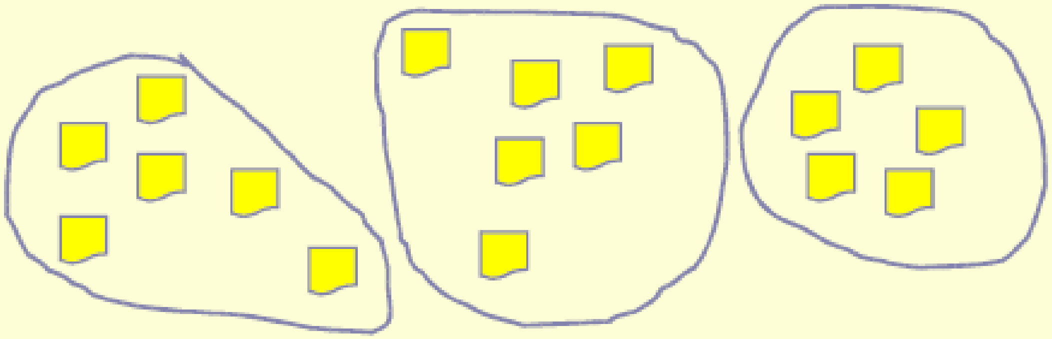
Constant search profile (e.g. Spam filtering)



Flavors of Information (Document) Retrieval (2/2)

- Categorization/Clustering:

  Group documents into predefined classes/ adaptive clusters



- Topic Detection and Tracking:

  Cluster news in stream

MANUAL AND AUTOMATIC INDEXING

INDEXING

- Indexing by Derivation
  - Index terms are derived from the document (and possibly morphologically normalized)
- Indexing by Assignment
  - Index terms are assigned to a document using an authoritative terminology (usually, a thesaurus)

INDEX TERMS

- Nouns (singletons, compounds)
  - Cell, blood cell,
- Noun phrases
  - Hot spot, regulation of cells
- Avoid too complex terms (pre-coordination)
  - The regulation of cells under laser beam exposure in vitro

MANUAL INDEXING

- Determine main topic(s)
- What‘s a relevant issue?
- Based on human (speed) reading and understanding of the document

AUTOMATIC INDEXING

- Absolute vs. relative frequency
  - Per document
  - Relative to document collection
  - Eliminate stop words (high occurrence frequency!)
- Assumption: frequency is positively correlated with relevance (denotation of main topics)
- Term frequency – inverse document frequency metric

w_{ij} = weight of term t_j in document d_i

tf_{ij} = frequency of term t_j in document d_i

N = number of documents in collection

n = number of documents where term t_j occurs at least once

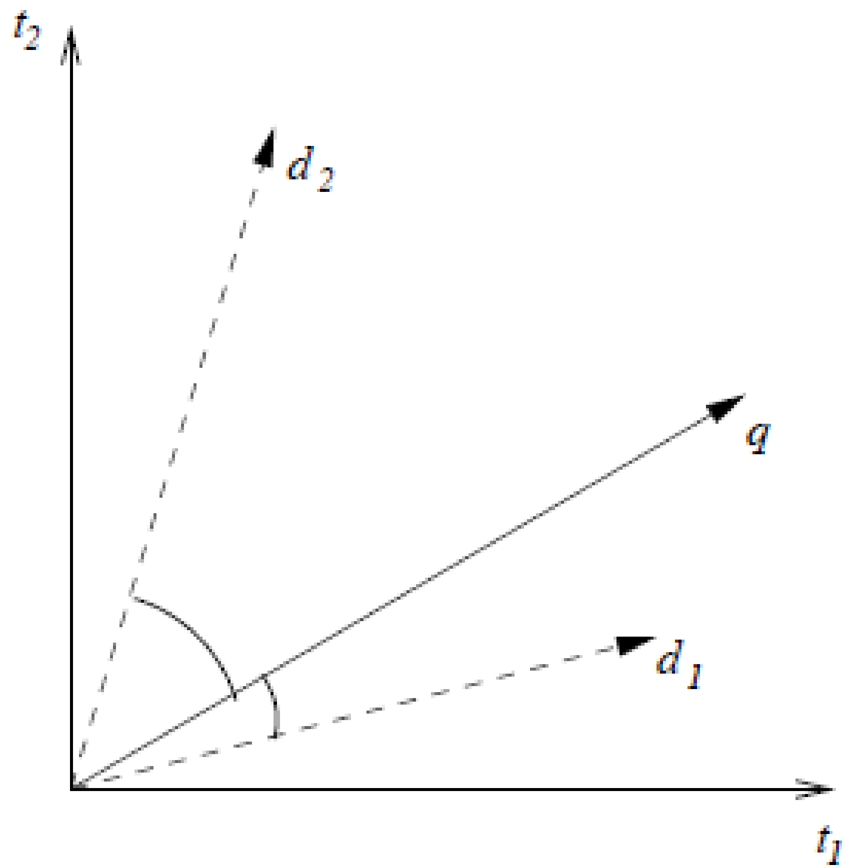
$$w_{ij} = tf_{ij} * \log_2 \frac{N}{n}$$

AUTOMATIC INDEXING (Vector Space Model)

- Bag of words: remove all stop words from a doc and normalize all terms morphologically
- Create a document term matrix from the remaining terms for each document (*n* being the max number of terms in the document collection)
  - $doc_i = (term_{i1}, term_{i2}, term_{i3}, ..., term_{in})$
    - Each component $term_{ik}$ is either ‚0‘ (absent) or ‚1‘ (realized)
- Compute the association between a document term and a query term vector ($query = (query_1, query_2, query_3, ..., query_n)$, *n* as above), e.g., using the cosine measure

$$SIM(doc_i, query) = \frac{\sum_{k=1}^{t} (term_{ik} \bullet query_k)}{\sqrt{\sum_{k=1}^{t} (term_{ik})^2 \bullet \sum_{k=1}^{t} (query_k)^2}}$$

GRAPHICAL INTERPRETATION



CLASSIFICATION

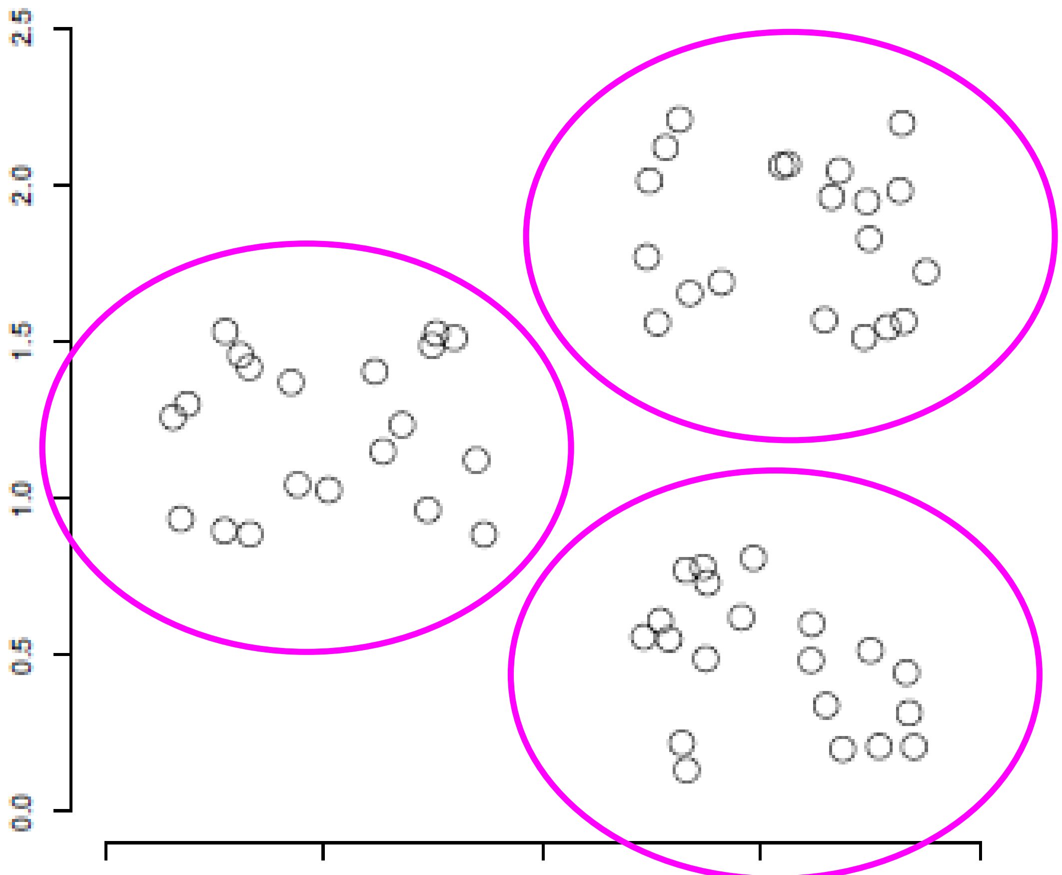
- Manual classification
  - Manual assignment of docs to pre-defined categories (classes)

- Automatic classification
  - Automatic assignment of docs to pre-defined categories (classes)
  - Grouping of docs around automatically determined (unnamed) clusters

Clustering

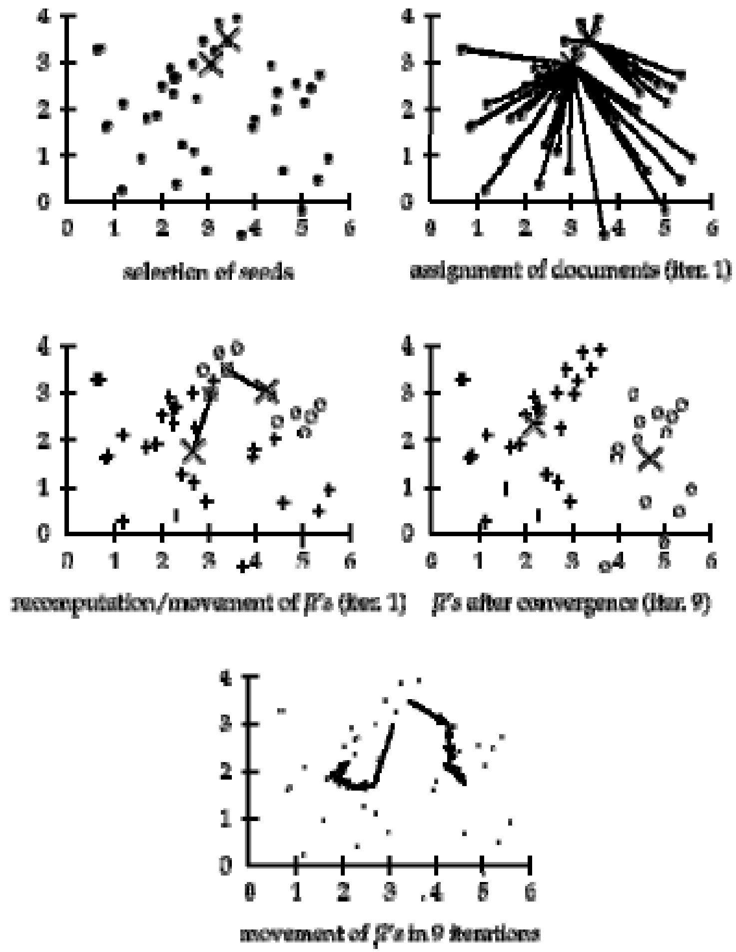
- (Document) clustering is the process of grouping a set of documents into clusters of similar documents.
- Documents within a cluster should be similar.
- Documents from different clusters should be dissimilar.
- Clustering is the most common form of unsupervised learning.
- Unsupervised = there are no labeled or annotated data.

Data Set with Clear Clustering Structure



Cluster-Modelle

- k-Means Clustering
  - flaches Clustering
  - k ist vorher bekannt
  - Dokumente werden als Vektoren repräsentiert
  - Ziel: Abstand zum Cluster-Zentrum minimieren
- Centroid
  - künstliches Zentrum eines Clusters – Mittelwert der Vektoren der Dokumente im Cluster
- RSS
  - Residual Sum of Squares
  - wie Centroid, nur quadratische Summen der Abstände
  - damit werden „Ausreißer" stärker gewichtet
- Algorithmus
  - Initialisierung: wähle zufällig k Dokumente als Centroiden
  - Iteration: ordne Dokumente nächstem Centroid zu, Centroid im Cluster neu berechnen



Quelle: Manning, Raghavan, Schütze, Introduction to Information Retrieval, 2008.

K-means Clustering

- Each cluster in K-means is defined by a centroid.
- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Recall definition of centroid:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

  where we use ω to denote a cluster.
- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid
  - recomputation: recompute each centroid as the average of the vectors that were assigned to it in reassignment

K-means Clustering Algorithm

$K\text{-MEANS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow \text{SELECTRANDOMSEEDS}(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
for $k \leftarrow 1$ **to** K
do $\vec{\mu}_k \leftarrow \vec{s}_k$
while stopping criterion has not been met
do for $k \leftarrow 1$ **to** K
    do $\omega_k \leftarrow \{\}$
    for $n \leftarrow 1$ **to** N
    do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
        $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
    for $k \leftarrow 1$ **to** K
    do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

Idee des Relevance Feedback

- Relevance feedback: user feedback on relevance of docs in initial set of results
  - *User issues a (short, simple) query*
  - *The **user** marks some results as relevant or non-relevant.*
  - *The **system** computes a better representation of the information need based on feedback.*
  - *Relevance feedback can go through one or more **iterations**.*
- Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

Annahmen zum Relevance Feedback

- User has sufficient knowledge for initial query.
- Relevance prototypes are “well-behaved”.
  - *Term distribution in relevant documents will be similar*
  - *Term distribution in non-relevant documents will be different from those in relevant documents*
    - *All relevant documents are tightly clustered around a single prototype.*
    - *Similarities between relevant and irrelevant documents are small*

Relevance Feedback (Rocchio-Algorithmus)

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

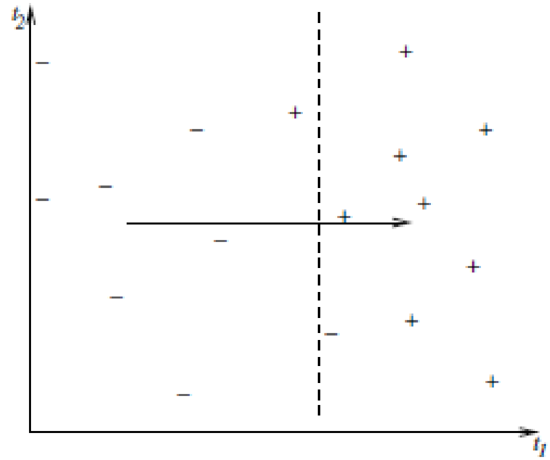
$$\vec{q}_k\,' = \vec{q}_k + \alpha \frac{1}{|D_k^R|} \sum_{d_j \epsilon D_k^R} \vec{d}_j - \beta \frac{1}{|D_k^N|} \sum_{d_j \epsilon D_k^N} \vec{d}_j$$

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

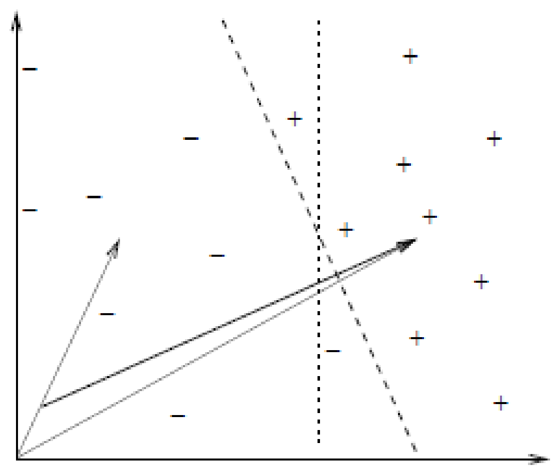
Vorgehensweise:

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

Idee des Relevance Feedback (Rocchio-Algorithmus)



unterschiedliche Gewichtung positiver und negativer Beispiele:



Rechenbeispiel zum Relevance Feedback

- Beispiel:

Original query $\begin{bmatrix} 0 & 4 & 0 & 8 & 0 & 0 \end{bmatrix}$ $\alpha = 1.0$ $\begin{bmatrix} 0 & 4 & 0 & 8 & 0 & 0 \end{bmatrix}$

$\begin{bmatrix} 2 & 4 & 8 & 0 & 0 & 2 \end{bmatrix}$ $\beta = 0.5$ $\begin{bmatrix} 1 & 2 & 4 & 0 & 0 & 1 \end{bmatrix}$ (+)

$\begin{bmatrix} 8 & 0 & 4 & 4 & 0 & 16 \end{bmatrix}$ $\gamma = 0.25$ $\begin{bmatrix} 2 & 0 & 1 & 1 & 0 & 4 \end{bmatrix}$ (-)

New query $\begin{bmatrix} -1 & 6 & 3 & 7 & 0 & -3 \end{bmatrix}$

ANTWORTEN VON INFORMATIONSSYSTEMEN

Datenbanksysteme liefern stets korrekte und vollständige Antwort auf Anfragen

- im Sinne eines Beweisverfahrens
- i.a. *nicht* bezüglich der realen Welt

→ Betrachtung von Effektivität hier nicht sinnvoll

IR-Systeme können wegen Vagheit und Unsicherheit i.a.

- weder korrekte (alle gefundenen Dokumente relevant)
- noch vollständige (alle relevanten Dokumente)

Antworten liefern.
→ Effektivität als wichtiges Qualitätskriterium

EVALUATION UND „RELEVANZ“

(“fiktive" Beziehung zwischen Anfragen und Dokumenten) als Mittel zur Beurteilung von Retrievalalgorithmen

Annahmen

- Systemantwort ist eine Menge von Dokumenten
- Qualität des Dokuments hängt nur von der Anfrage ab

Probleme

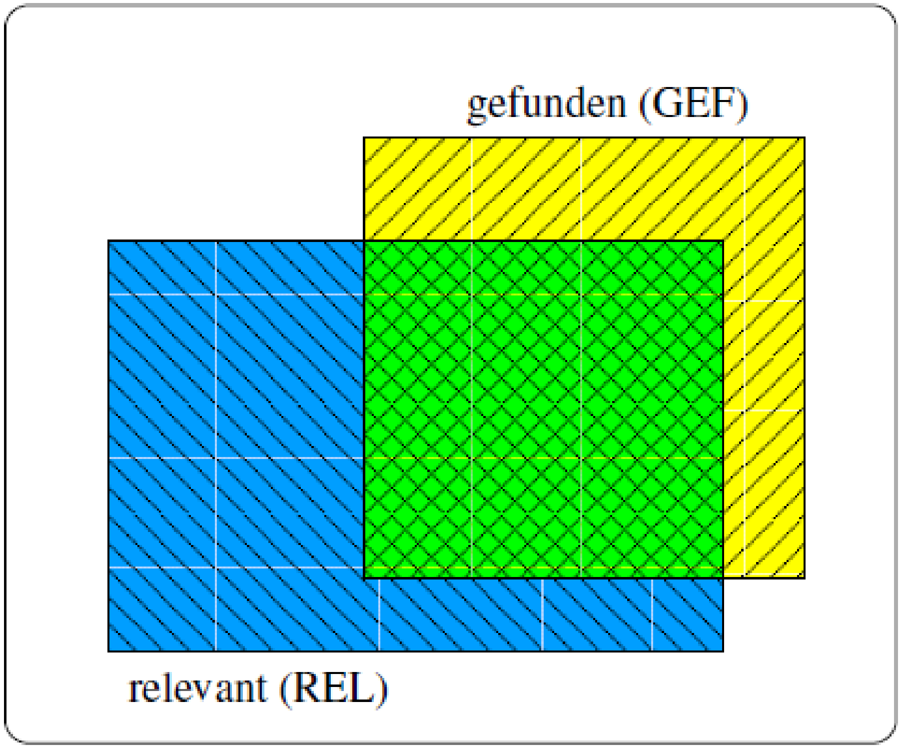
- Systemantwort kann strukturiert sein
- Dokumente nicht unabhängig
- Keine einfache Beziehung zwischen Informationsbedürfnis (umgangssprachlich/subjektiv) und Anfrage (formal)

EVALUATIONSMETRIKEN

GEF: Menge der gefundenen Antwortdokumente

REL: Menge der relevanten Dokumente in der Datenbank

ALL: Menge aller Dokumente in der Datenbank



Kollektion

$$\text{Precision} \quad p = \frac{|REL \cap GEF|}{|GEF|}$$

$$\text{Recall} \quad r = \frac{|REL \cap GEF|}{|REL|}$$

$$\text{Fallout} \quad f = \frac{|GEF - REL|}{|ALL - REL|}$$

INTEGRATION IM F-MASS

Abbildung von (r, p)-Paar auf einzelnes Maß
(definiert Kurve zur Aufteilung des 'Unentschieden-Bereichs')

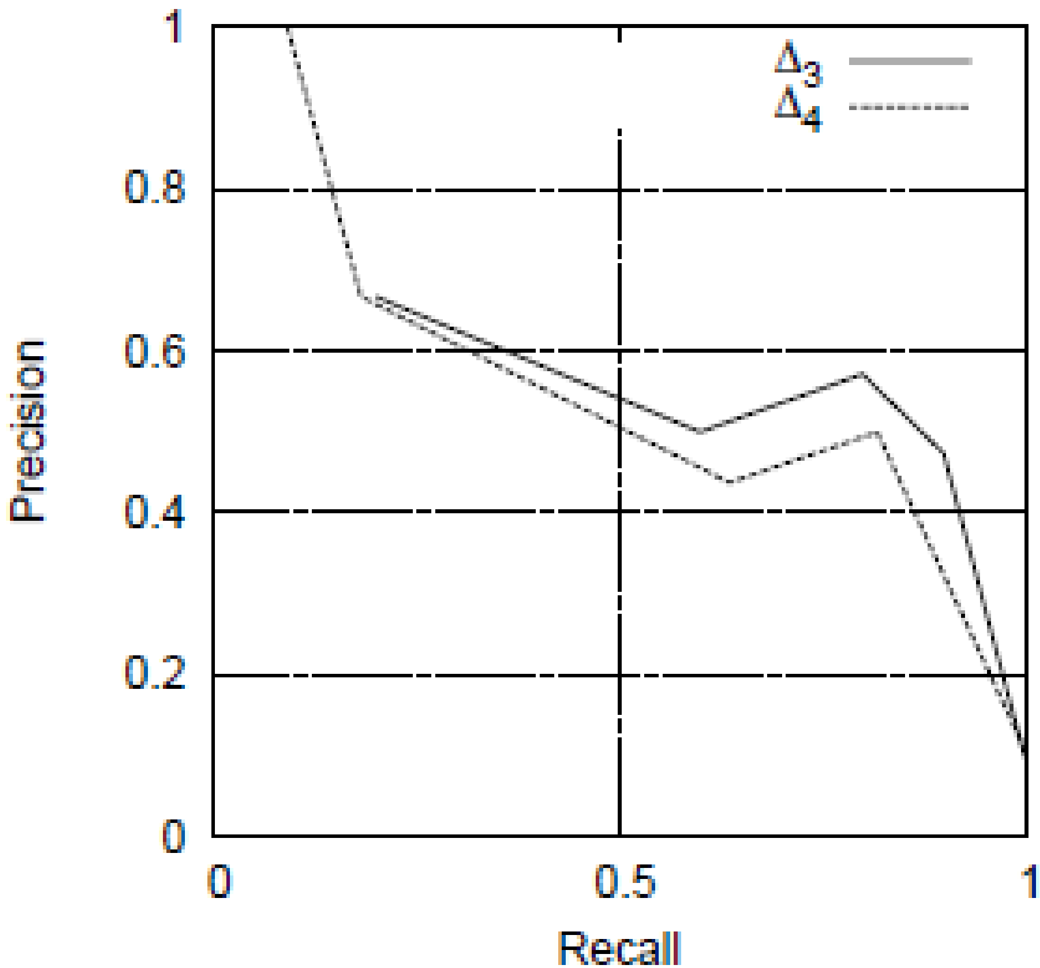
Grundidee:
harmonisches Mittel aus Recall und Precision

$$F = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

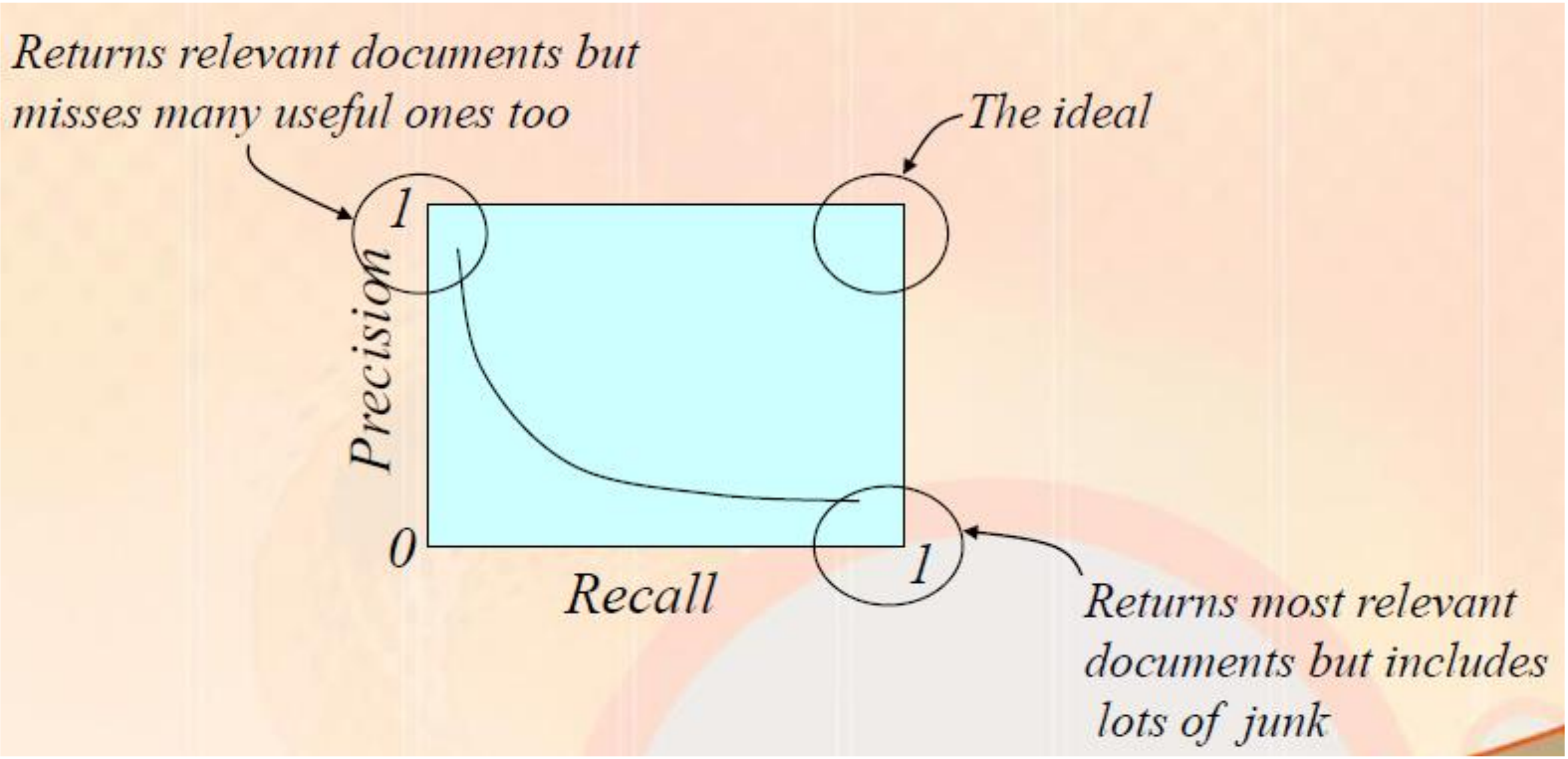
Unterschiedliche Gewichtung von Recall und Precision:
Gewichtungsfaktor β für Recall

$$F_\beta = \frac{1 + \beta^2}{\frac{1}{p} + \beta^2 \frac{1}{r}}$$

„NATURGESETZ“ DER INVERSEN P-R-BEZIEHUNG



Trade-off between Precision and Recall



Literatur

- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, 1999.
  - der Klassiker
- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008.
  - Online verfügbar unter: http://nlp.stanford.edu/IR-book/information-retrieval-book.html