

# Information Engineering 1: Information Retrieval

Kategorisierung/Recommender mittels  
Information Retrieval

Kapitel 5

Martin Braschler

# Agenda

- Recommender basierend auf Kategorisierung/Klassifikation
- Content-based Verfahren
  - Rocchio
  - kNN
  - Bayes
- Collaborative Filtering
- Thesauri
- Klassifikationen
- Social Tagging

Material u.a. von Prof. H.-P. Frei, Ellery Smith

# Dokumentkategorisierung

- Definition: Bei der Kategorisierung werden Dokumente anhand ihres Inhaltes einer speziellen Kategorie (im Allg. Teil einer Informationsstruktur) zugewiesen.
- Eine Kategorie sollte innerhalb einer bestimmten Domäne einen wohl definierten Bereich darstellen. Mit anderen Worten: Es geht um den Umgang mit Bedeutung, obwohl wir in den meisten Fällen nur "rohe Daten" zur Verfügung haben
- Abgrenzung: wir betrachten Methoden, die Kategorisierung als Suchproblem behandeln, und für die wir die besprochenen Ansätze (Vektorraummodell etc.) adaptieren können

# Kategorisierung Aufgaben

- Es gibt eine Menge unterschiedlicher Aufgaben:
- Indexierung (Verschlagwortung)
  - **Manuelle Methoden** der Indexierung sind für Online-Kollektionen schwerfällig und kostenintensiv. Die Frage ist, wie man die menschliche Indexierung semi-automatisieren kann. Das kontrollierte Vokabular der Indexierungssprache bildet die Kategorien.
- Recommender (klassisch: Routing/Filtering)
  - Die einkommenden Dokumente werden regelmässig gegen ein **Userprofil resp. Themenprofil** getestet, um zu bestimmen, wo erstere einzuordnen sind (Dokumenten-Feed). Verwandt: Push-Dienste

# Kategorisierung Aufgaben (cont.)

## ■ Clustering

- Gruppieren von Kollektionen (z.B. Memos, E-Mails) in eine Menge von sich gegenseitig ausschliessenden Kategorien – die aus den Daten gebildet werden. Schwierig wenn Dokumente kurz sind. → Es gibt Ausreisser!

## ■ Annotation

- Gruppieren von Dokumenten (z.B. wissenschaftliche Artikel) mit weiterführender, dazugehöriger Information.

# Unterschied Kategorisierung/Klassifikation

- Kennen Sie den Unterschied Kategorisierung/Klassifikation?

# Unterschied Kategorisierung / Klassifikation

- Eine „Mitgliedschaft“ in einer **Kategorie** ist nicht exklusiv, die Grenzen können verschwimmen/-überlappen → In diesem Sinne sind nicht notwendigerweise alle einer Kategorie zugeordneten Informationseinheiten gleich gute Repräsentanten für die Kategorie. Ein Objekt/Dokument kann zu mehreren Kategorien gehören
- eine **Klassifikation** besteht aus starren, exakt disjunkten Klassen, die hierarchisch angeordnet sind.
- Auch in der Literatur ständig falsch verwendet ☹

# Content-based vs. Collaborative

Wir unterscheiden grundsätzlich:

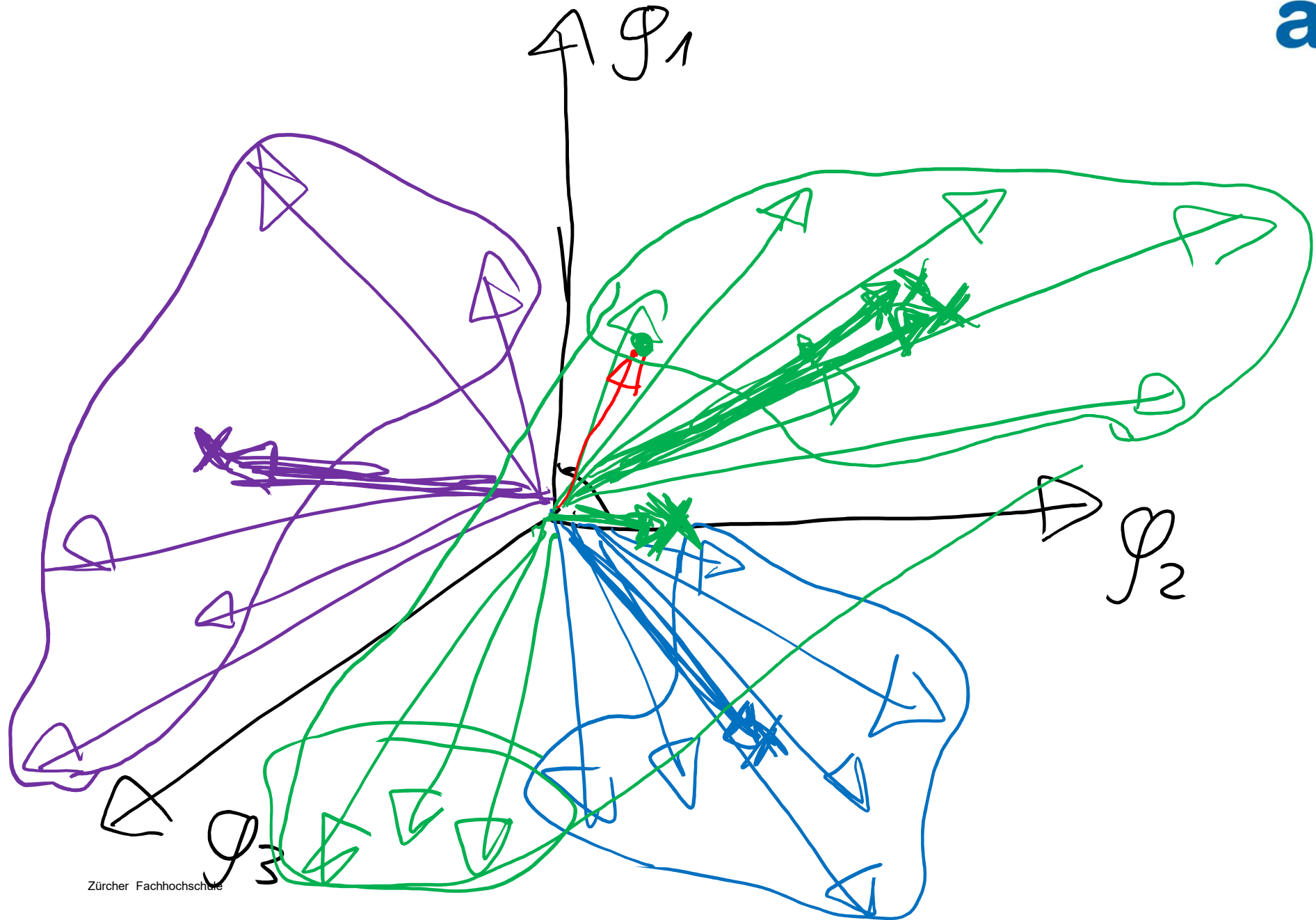
1. «Content-based» Verfahren: die Kategorisierung erfolgt aufgrund des «Inhalts» (resp. der Beschreibung) des Objektes (welches deshalb vollständig digital vorliegen muss)
2. «Collaborative»-Verfahren: die Kategorisierung erfolgt aufgrund von externen Signalen, wie z.B. Bewertungen oder Clicks (es reicht daher ein digitales Surrogat)



# (Content-Based)-Verfahren mit Wurzeln im Information Retrieval

- Rocchio Model
- kNN, Nearest Neighbor Algorithmus
- Bayes Klassifizierung

# Skizze Rocchio



# Rocchio Model

- Rocchio modelliert eine Kategorie C mittels einem Kategorie-Repräsentanten c. Der Repräsentant  $c = (c_1, \dots, c_n)$  ist ein Vektor und wird konstant aktualisiert.
- Die Komponenten vom (neuen) c sind:

$$c_j = \alpha c'_j + \beta \frac{1}{n_c} \sum_{D \in C} d_j - \gamma \frac{1}{n - n_c} \sum_{D \notin C} d_j$$

- Wobei:
  - D: Dokumente der Kollektion ( $d_j$  = einzelnes Dokument)
  - n: Totale Anzahl von Dokumenten in der Kollektion
  - $n_c$ : Anzahl von Dokumenten in Kategorie C
  - $\alpha, \beta, \gamma$ : kontrolliert den relativen Einfluss der drei Gewichtungskomponenten

# Rocchio Algorithmus

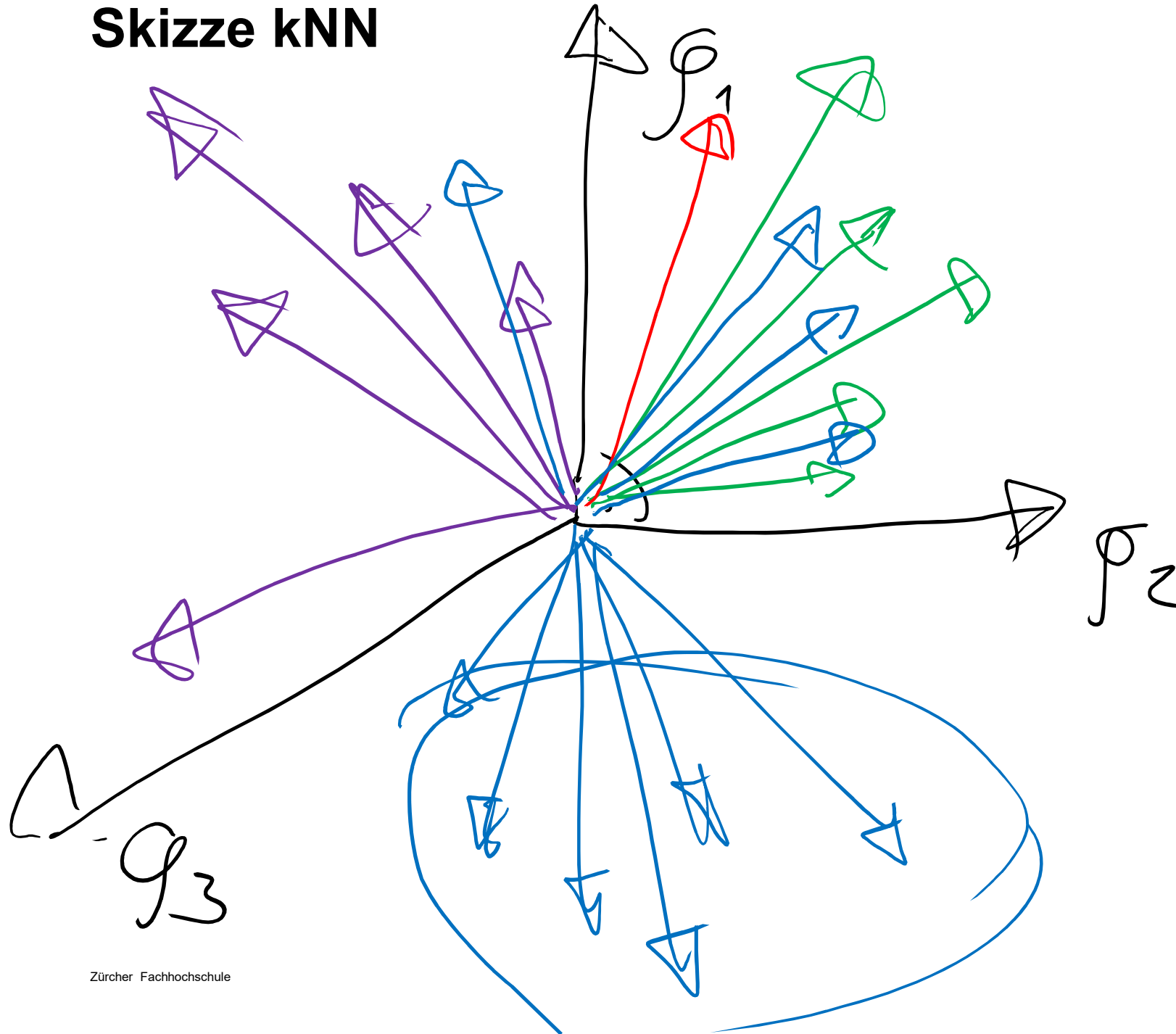
- Der Kategorie-Repräsentant  $c$  ist wie ein normales Dokument mit dem Unterschied, dass er nicht real existiert („hypothetisch“). Er wird z.B. initial aus positiven Beispielen generiert.
- Die Idee besteht darin, den Repräsentanten in Richtung der positiven Beispiele und weg von den negativen Beispielen zu bewegen. → kommt Ihnen das bekannt vor?
- Rocchio's Kategorisierung
  - Neue Dokumente werden als Anfrage zum Vergleich mit den Repräsentanten verwendet (z.B. Winkel berechnen, Vektormodell)
  - Falls  $s(D,C) > \text{delta}$  wird das Dokument  $D$  der Kategorie  $C$  zugeteilt.  $\text{delta}$  ist ein Grenzwert, der geeignet bestimmt werden muss.

# Bewertung Rocchio Methode

## ■ Dieser Algorithmus

- ist einfach zu implementieren und sehr effizient (→ wieso?). Er wird vielfach als Grundlage in Kategorisierungs-Experimenten gebraucht.
- ein gravierender Nachteil ist, dass er nicht robust ist (vor allem wenn die Anzahl von negativen Instanzen gross wird).
- die Festlegung der Parameter ist **knifflig** und hängt stark von der Art und Grösse der Kollektion ab.
- hat Probleme mit Kategorien, die mehrere Facetten haben (→ wieso?).
- Verbesserte Versionen von Rocchio können deutlich effektiver sein (ähnlich komplizierteren Verfahren).
- Um den Vorgang zu starten, ist eine Trainingskollektion notwendig (→ wozu?)
- verwandt mit dem Relevance Feedback-Verfahren des gleichen Urhebers

# Skizze kNN



1. G
2. G
3. V
4. V
5. G
6. G
7. .
8. .
- . .
- . .

# Nearest Neighbor Algorithmus (kNN)

- Die kNN (k Nearest Neighbor) Methode verwendet ein Ähnlichkeitsmass (Euklidische Distanz, Kosinus) und eine Regel, wie Dokumente D Kategorien zuzuordnen sind.
- Einfache Regeln:
  - bestimme die k Dokumente, die am ähnlichsten zu Dokument D sind, d.h. die k nächsten "Nachbarn".
  - ordne Dokument D einer oder mehreren Kategorien zu, die bereits den Nachbarn zugeordnet sind
- Beachte:
  - Um den Vorgang zu starten, ist eine (ziemlich grosse) Trainingskollektion notwendig.

# Erweiterte kNN Klassifikation

## ■ Idee:

- Je weiter ein Dokument  $D$  vom Nachbar  $D_j$  entfernt ist (Ähnlichkeitsmass!), desto weniger trägt es zur Entscheidung bei, Dokument  $D$  in die Kategorie  $C_j$  zuzuordnen.
- Mit anderen Worten: Berechne Wert  $s_c$  für jede potentielle Klasse  $C_j$  (eine simple Variante ist wie folgt):

$$sc(C_j, D) = \sum_{D_i \in kNN(D)} sim(D, D_i) \cdot a_{i,j}$$

- Wobei  $kNN(D)$  die Menge von  $k$  nächsten Nachbarn von  $D$  ist.
- $a_{i,j}=1$  falls Dokument  $D_i$  zu Klasse  $C_j$  gehört und  $a_{i,j}=0$  andernfalls.
- Probleme:
  - Richtige Wahl von  $k$
  - Richtige Wahl der Funktion  $s_c$ , und der maximalen Anzahl zugeordneter Kategorien
  - Schwellwert bei Dokumenten, die mehreren Kategorien angehören sollten



# Bewertung kNN

- Effektiv
- Relativ einfach, stabile Schwellwerte zu finden
- Langsam
- Die Wahl eines einzelnen Wertes "k" ist zu simpel

# Bayes Klassifizierung

- *Gegeben:* Kategorie  $C_i$  mit einer angemessenen Anzahl von bereits zugeordneten Objekten (Trainingsdaten).
- *Methode:* Bilde statistische Modelle aus diesen Kategorien. Benutze diese, um vorherzusagen, zu welcher Klasse ein neues Objekt  $D$  gehört.
- Wir kennen  $P(t|C_i) \forall t, C_i$ , sind aber eigentlich interessiert an:  $P(C_i|t)$  oder noch spezifischer in  $P(C_i|D)$ 
  - wobei  $D$  für die *Menge von Merkmalen* in Objekt/Dokument  $D$  steht
- Die Wahrscheinlichkeit, dass  $D$  zu  $C_i$  gehört ist (Bayes'Rule):

$$P(C_i | D) = \frac{P(D | C_i)P(C_i)}{P(D)}$$

# Bayes Klassifizierung

- Mit anderen Worten: „alte“ Objekte der Klasse  $C_i$  bestimmen für uns:
  - Die Merkmale nach den zu suchen ist
  - Erwartete Merkmalshäufigkeit in „neuen“ Objekten

- Mit  $D=(t_1, \dots, t_n)$  und in Beziehung zu Klasse  $C_i$ , wir können sagen:

$$P(D | C_i) = \prod_{j=1}^n P(t_j | C_i)$$

- Was bedeutet diese Annahme? Ist dies eine brauchbare Annahme?
- Die vorherigen Wahrscheinlichkeiten  $P(C_i)$  und  $P(D)$  müssen berechnet werden. Es gibt verschiedenen Wege um  $P(t|C_i)$  zu berechnen: zähle die Anzahl Merkmale, binär (Vorkommen/Nicht-Vorkommen), gewichtet...

# Bewertung Bayes

- Sauberes Modell
- Einfach zu implementieren
- Performt schlecht, typischerweise schlechter als andere einfache Verfahren
- Die Unanbhängigkeitsannahme ist wohl zu simpel

# Exkurs: Regelbasierte Methode

- Expertensysteme versuchen, gewünschte Kategorien mittels geeigneter Regeln zu beschreiben.
- Beispielhaftes Vorgehen
  - Selektion von geeigneten Beispieldokumenten
  - Manuelle Selektion von Stichwörtern, Verknüpfung mittels logischem Ausdruck: Was für eine Suchanfrage ergibt diese Beispieldokumente als Resultat?
  - Auch: automatische Herleitung von Regeln (hier nicht besprochen)
- Beachte Sie, dass es extrem schwierig ist, beständige (lange gültige) Anfragen zu formulieren (sogenannte Benutzerprofile).

## Regelbasierte Methode (cont.)

- Regelbasierte Kategorisierung funktioniert relativ gut für sehr "scharfe" Konzepte. Sie können auch ergänzend als Filter eingesetzt werden.
- Beispiel: US Dollar vs. Australian Dollar
  - Sehr ähnliche Terminologie
  - Die (Trainings- und Test-)dokumente lassen sich im Vektorraum nicht gut voneinander abgrenzen
- Beispiel: Dokumente von der Credit Suisse vs. über die Credit Suisse
  - Der Unterschied «von/über» schlägt sich nicht wirklich im Vokabular nieder
  - Metadaten sind entscheidend
- Verbesserungen sind insbesondere dann möglich, wenn die zu suchenden Konzepte in speziellen Feldern vom Text auftreten (z.B. Metdaten, Titel etc.).

# Collaborative Filtering anhand «The Netflix Prize»



- Collaborative Filtering ist eine Alternative zu Content-based Categorization, die grundlegend anders funktioniert: die Empfehlungen entstehen aufgrund von externen Signalen, wie Bewertungen
- Wir illustrieren die Idee anhand des «Netflix Prize». Dies ist aber nur der Aufhänger; die Überlegungen sind allgemeingültig.

# Hintergrund «The Netflix Prize»

- Netflix existiert bereits länger als der gleichnamige Streamingdienst. In dieser «Frühzeit» war Netflix ein Versandanbieter für Miet-DVDs.
- Grundlegendes Problem: die Kunden/innen konnten den Film nicht anspielen, bevor sie diesen bestellen – und ein Fehlgriff war ärgerlich (Kosten, Zeitverlust).
- Bestmögliche Empfehlungen waren also essentiell, und Netflix ein Vorreiter in dieser Hinsicht.



# Hintergrund «The Netflix Prize»

- Ausgelobt wurde ein Preisgeld von \$1 Million. Ziel war es, den Hausalgorithmus «CineMatch» um mindestens 10% zu schlagen
- Bessere Resultate als CineMatch wurden bereits nach 6 Tagen veröffentlicht, aber es waren 2 Jahre nötig, um die gewünschten 10% zu erreichen
- In Hollywood-Manier kam es zum Schluss zu einem Fotofinish: ein Team gewann mit 20 Minuten Vorsprung.

# Collaborative Filtering: Setup

- Gegeben sei ein unvollständiges Datenset (als Matrix User x Film interpretierbar)
- Der Algorithmus muss die fehlenden Bewertungen (Skala 1-5 Sterne) ergänzen. Aus diesen folgen dann die Empfehlungen.

	Inception	Avatar	Titanic	The Godfather	...
User 1	★★★★★	★★★★★★	★★★★★★	★★	...
User 2	★	★★	—	★★★★★	...
User 3	★★★★★★	—	★★★	—	...
User 4	???	★★★★	???	★★	...

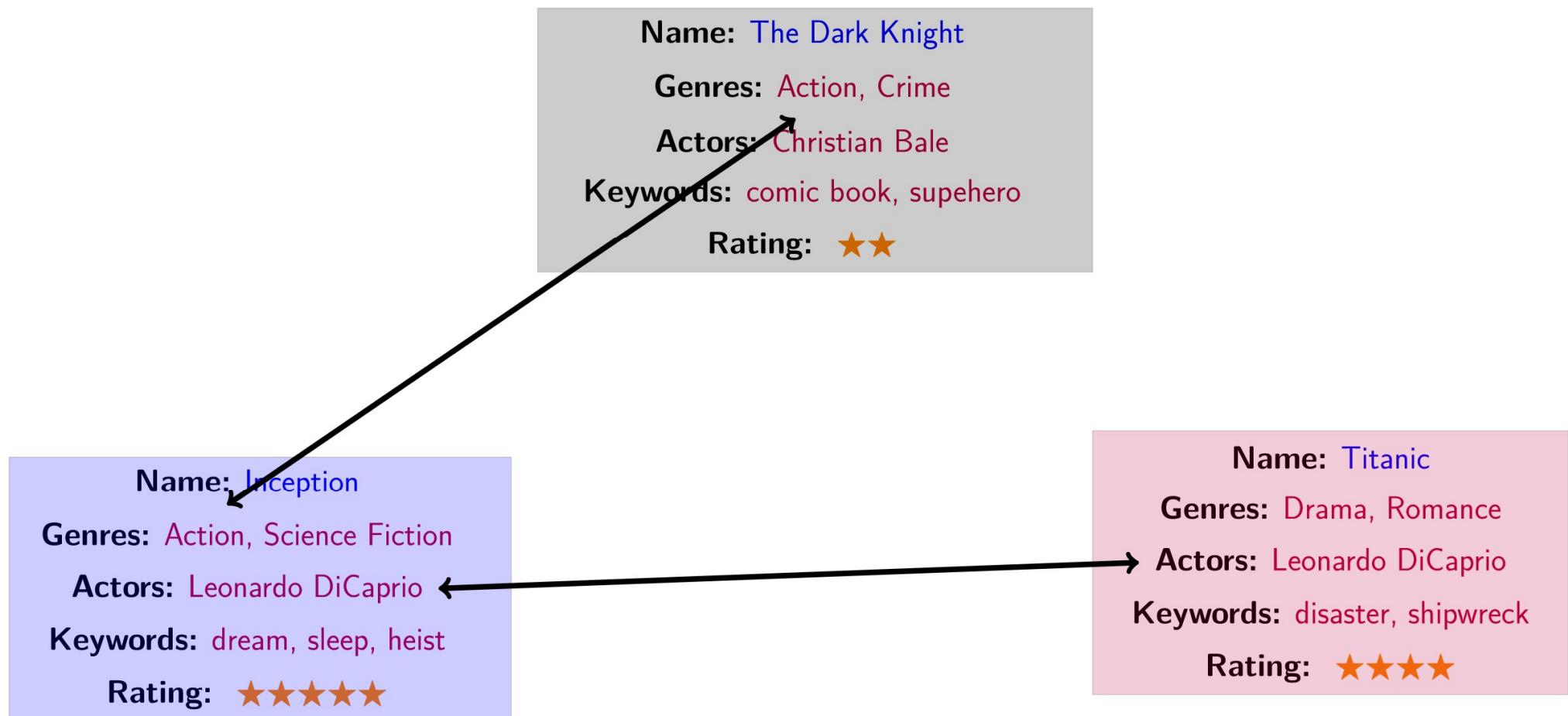
# Warum Collaborative Filtering?

- Die Empfehlungen folgen aus den Bewertungen, nicht aus einer inhaltlichen Ähnlichkeit der Filme
- Beispiel: **User 1** scheint ähnliche Vorlieben wie **User 4** zu haben
- Wir folgern daraus, dass **User 4** *Titanic* und *Inception* mögen wird.

	Inception	Avatar	Titanic	The Godfather	...
User 1	★★★★★	★★★★★★	★★★★★★	★★	...
User 2	★	★★	—	★★★★★	...
User 3	★★★★★★	—	★★★	—	...
User 4	???	★★★★	???	★★	...

# Content-Based Filtering

So, wie wir das Problem bis jetzt behandelt haben, müssten wir eine inhaltliche Ähnlichkeit z.B. auf Metadaten feststellen



# Collaborative Filtering

Collaborative Filtering hat das Potential, abstrakte Beziehungen zwischen Daten und Vorlieben der Nutzer/innen aufzudecken

## Beispiele

- Ist Jason Statham ein ähnlicher Schauspieler wie Vin Diesel?
- Sollen Tracks einer Tribute Band (z.B. "Kings of Floyd" o.ä.) vorgeschlagen werden?
- Soll ein Remake eines alten Filmes (z.B. "Ghostbusters" Reboot) empfohlen werden?

# Collaborative Filtering

## Content-basiert:

- beide spielen in vielen Actionstreifen. Aber auch Schauspielerinnen wie Scarlett Johansson haben ein ähnliches Portfolio – wir müssen also solche Tatsachen sauber gewichten
- Tribute Bands spielen die gleichen Songs, daher sehr ähnlich – aber auch in derselben Qualität?
- Reboots nutzen die selben Charaktere und Plotlines, daher sehr ähnlich – aber will der/die Nutzer/in nochmals “denselben” Film sehen?

## Collaborative:

- Wenn die meisten Nutzer/innen, die Jason Statham mögen, auch gerne Filme mit Vin Diesel schauen, dann sind sie “ähnlich”
- Wenn die meisten Nutzer/innen die Tribute Band ignorieren, dann nicht “ähnlich”
- Wenn die meisten Nutzer/innen den Reboot ablehnen, dann nicht “ähnlich”

# Collaborative Filtering

- Solche impliziten Beziehungen schlummern häufig in Daten, und die Nutzer/innen sind sich ihrer oft nicht bewusst
  - Nutzer/innen folgen in ihren Präferenzen auch Mustern, z.B.
    - Nutzer/innen, die Action mögen, wollen keine Romantic Comedies
    - Kinder schauen einfache, kurze Filme, etc.
- Ähnlichkeiten zwischen Nutzern können oft ein besserer Indikator für gute Empfehlungen sein als der eigentliche “Inhalt”

# Collaborative Filtering

Gegeben: zwei ähnliche Nutzer/innen

User 1



User 2



User 1 mag wahrscheinlich Star Wars  
User 2 mag wahrscheinlich keine spanischen  
Horrorfilme



# Collaborative Filtering: kNN

## Anpassung: k-Nearest Neighbours (kNN) Filtering

**Ziel:** Gesucht ist die Bewertung für Film  $M$  durch Nutzer/in  $U$

1. Identifiziere die  $k$  ähnlichsten Nutzer/innen für  $U$  ( $U$  als Vektor darstellen, z.B. Cosinus-Ähnlichkeit)
2. Bilde Untermenge der Nutzer/innen, die Film  $M$  bewertet haben
3. Berechne den Durchschnitt der Scores

# Collaborative Filtering

## Beispiel: Amazons Item-to-Item Filtering

**Ziel:** Weitere Produkte zum Kauf vorschlagen

(“Nutzer/innen, die **X** kaufen, kaufen auch **Y**”)

1. Grundlage ist eine (binäre) Nutzer x Produkt-Matrix

	<i>Item 1</i>	<i>Item 2</i>	<i>Item 3</i>	<i>Item 4</i>	<i>Item 5</i>
<i>User 1</i>	1	1	0	0	1
<i>User 2</i>	0	1	0	0	1
<i>User 3</i>	1	0	1	1	0
<i>User 4</i>	0	0	1	1	0

# Collaborative Filtering

## Beispiel: Amazons Item-to-Item Filtering

- Wir berechnen die Cosinus-Ähnlichkeit zwischen den Produktevektoren
- Ähnlichstes Produkt (oder Produkte mit  $\text{sim} > \text{delta}$ ) wird empfohlen

	<i>Item 1</i>	<i>Item 2</i>	<i>Item 3</i>	<i>Item 4</i>	<i>Item 5</i>
<i>User 1</i>	1	1	0	0	1
<i>User 2</i>	0	1	0	0	1
<i>User 3</i>	1	0	1	1	0
<i>User 4</i>	0	0	1	1	0

**Produkt 2** → **Produkt 5**, **Produkt 4** → **Produkt 3**

# Collaborative Filtering

- Wir haben hiermit einerseits eine andere Interpretation von “Ähnlichkeit” – Vorlieben vs. inhaltliche Ähnlichkeit
- Dies ist insbesondere auch interessant, um eine grössere Diversität in die Resultate zu bekommen: nicht nur viele “Quasi-Doubletten”, sondern auch spannende “Ausreisser”
- Dies ist in vielen Bereichen entscheidend: wer einen Kühlschrank gekauft hat, braucht keine weiteren Kühlschränke mehr

# Das Netflix Dataset

- Das Dataset bestand aus ca. **10,000,000 Bewertungen** von ca. **500,000 Nutzern/innen**
- Jedes System resp. Experiment musste weitere **3,000,000 Bewertungen** liefern, in der Form von Fließkommawerten zwischen 1.0 to 5.0
- Die Baseline hatte einen Fehler von ca. **0.95 stars**

# Eine Bewertung dekonstruiert

**Overall Average Rating:**  $+3.1 \times$  ★

**User-Critic Effect:**  $-0.3 \times$  ★

**Movie-Specific Deviation:**  $+0.7 \times$  ★

**Unkown Factor:**  $+0.6 \times$  ★

---

**Final Score:**  $4.1 \times$  ★

Eine Bewertung kann als eine Summe von Komponenten aufgefasst werden. Einige dieser Komponenten können einfach bestimmt werden.

“Overall Average” ist der Durchschnitt über alle Filme

Die Frage ist also: wie weicht der konkrete Film ab?

# Eine Bewertung dekonstruiert

**Overall Average Rating:**  $+3.1 \times$  ★

**User-Critic Effect:**  $-0.3 \times$  ★

**Movie-Specific Deviation:**  $+0.7 \times$  ★

**Unkown Factor:**  $+0.6 \times$  ★

---

**Final Score:**  $4.1 \times$  ★

Der “User-Critic effect” misst den Nutzer-Faktor: ist dies im Vergleich zur ganzen Population eine kritische oder eine wohlwollende Person?  
Sinngemäss wirkt die “Movie-Specific Deviation”: ist dies im Prinzip ein populärer oder ein ungeliebter Film?

# Eine Bewertung dekonstruiert

**Overall Average Rating:**  $+3.1 \times \star$

**User-Critic Effect:**  $-0.3 \times \star$

**Movie-Specific Deviation:**  $+0.7 \times \star$

**Unkown Factor:**  $+0.6 \times \star$

---

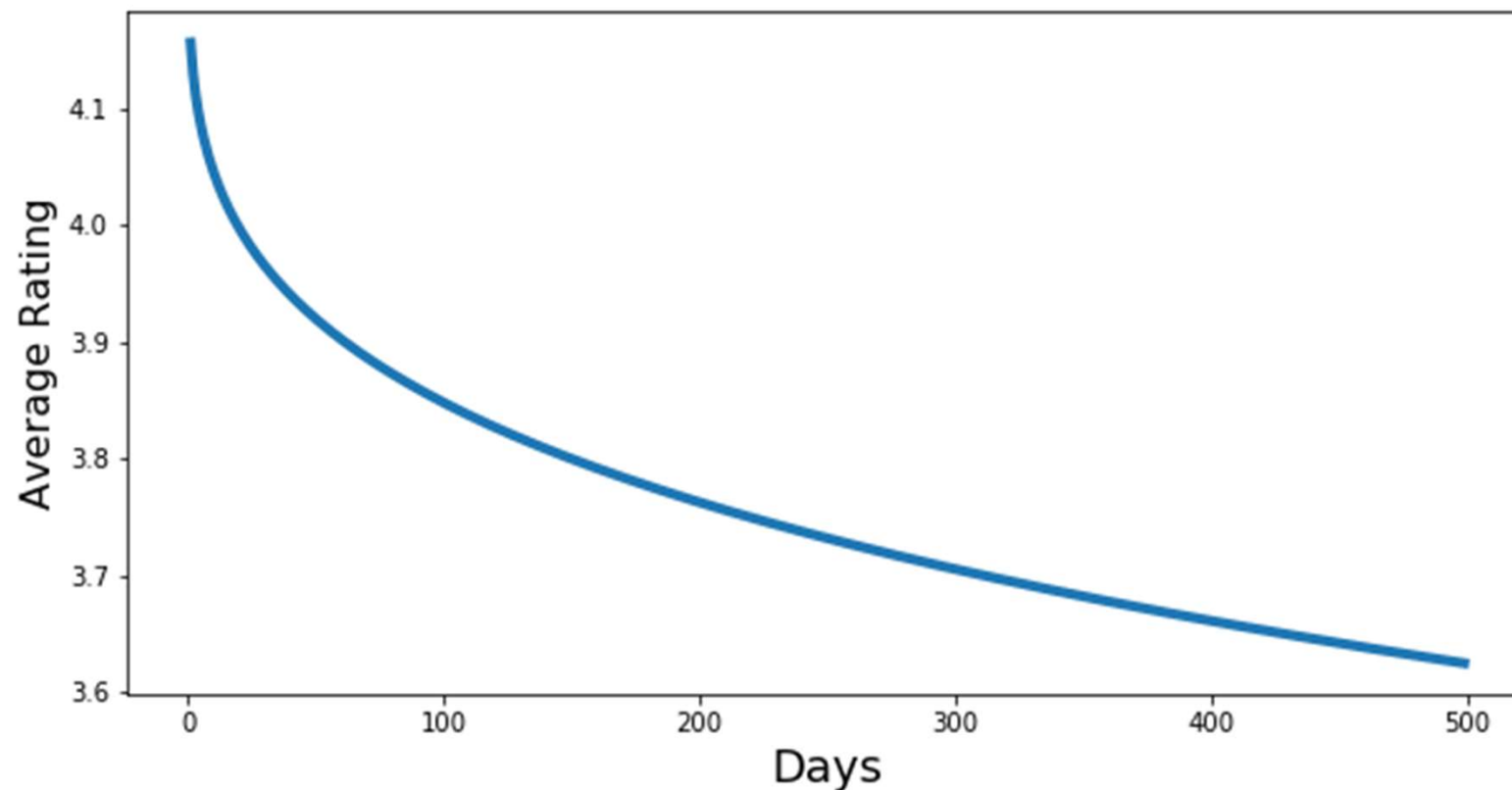
**Final Score:**  $4.1 \times \star$

Wenn wir diese drei einfach bestimmbaren Faktoren entfernen, können wir den spannenden Teil isolieren: die konkrete Präferenz eines bestimmten Nutzers in Hinsicht auf diesen spezifischen Film – hier 0.6 Sterne höher als erwartet



# Nutzer Biases

- Weitere Effekte müssen berücksichtigt werden: je mehr Filme ein Nutzer/in schaut, desto kritischer werden die Bewertungen



# Nutzer Biases

Wenn wir einen Tag isoliert betrachten, dann gilt für einen spezifischen Nutzer/in, der/die mehrere Filme (nicht unbedingt an diesem Tag geschaut) bewertet:

- Die Tagesstimmung beeinflusst die Bewertungen. Alle Filme werden entweder besser oder schlechter bewertet
- Nur Filme, welche “Ausreisser” sind, werden bewertet, d.h, die besten und schlechtesten
- Wir erhalten tendenziell keine Bewertungen für mittelmässige Filme

# Nutzer Biases

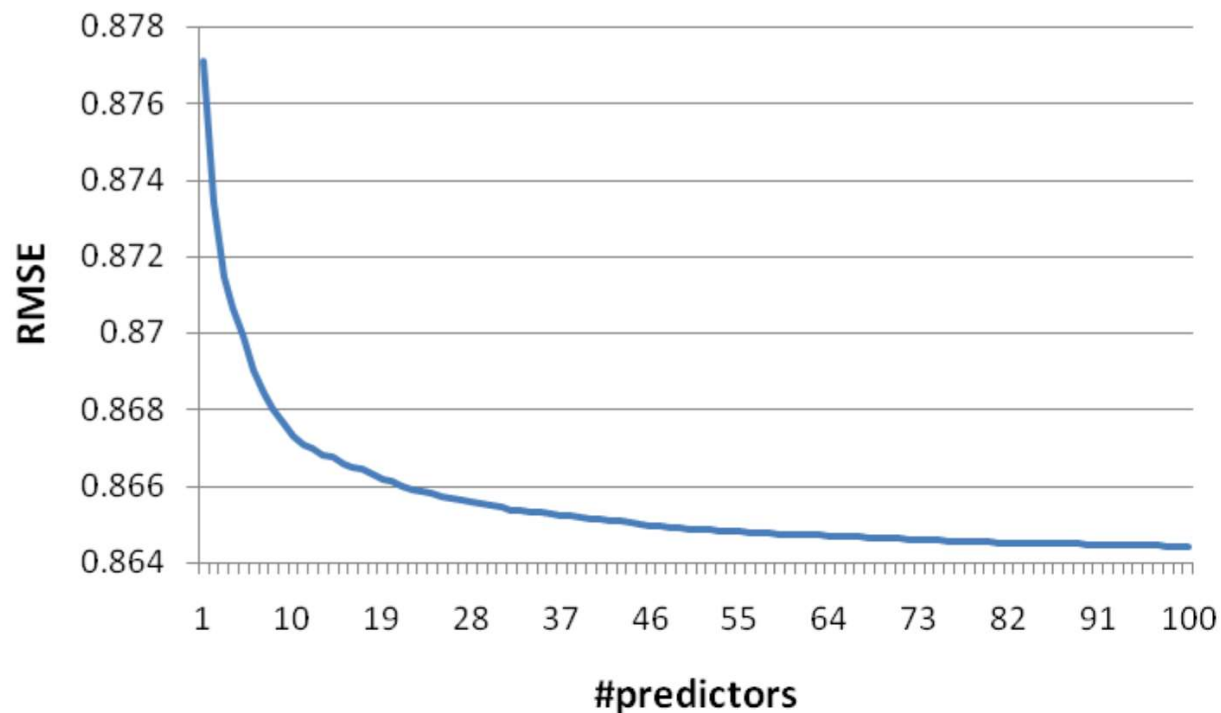
- Die Berücksichtigung solcher spezifischen Biases ist essentiell (Datenaufbereitung, Data Engineering)
- Der Einfluss ist grosser als der Algorithmus an sich. Bereits die Baseline funktioniert auf bereinigten Daten bedeutend (signifikant) besser)

# Netflix-Prize: Sieger

- Das Siegersystem war eine Linearkombination von **mehr als 200 verschiedenen Algorithmen** (“Kitchen-sink approach”)
- Unter anderem verwendet wurden:
  - Nearest Neighbours (kNN)
  - Matrix Factorisation (SVD)
  - Neural Networks
  - Decision Trees
- Dies wurde kombiniert mit einer Datenbereinigung (siehe oben)

# Analyse Lösung Sieger

Aber: der “Return on investment” bei einer solchen Kombination sinkt rapide.  
Schon 2 Methoden bringen 75% der Verbesserung (und sind viel besser skalierbar!)



# Reflexion Collaborative Filtering

- Unsere bestehenden Ansätze (z.B. kNN) können auch für Collaboratives Filing angepasst werden
- Möglichkeit, implizite “Signale” in den Daten zu nutzen
- Interessant in Sachen Diversität: nicht nur “Quasi-Doubletten”, sondern inhaltlich andere Objekte vorschlagen
- Probleme mit “Kaltstart”
- Sehr interessant: Hybrid aus content-based und collaborative

# Anwendung Kategorisierung: Informationsstrukturen

- Bei Informationsstrukturen geht es immer darum Dokumente/Informationsobjekte in Verbindung zueinander setzen.
- Was sind typische Informationsstrukturen:
  - Thesauri
  - Klassifikationen
  - Wörterbücher, Data Dictionaries, Konkordanzlisten,...
  - ER-Modelle
  - OO-Modelle
  - Prozess-Modelle (Geschäftsprozesse)
  - UML
  - Inferenzmodelle aus der künstliche Intelligenz
  - Navigationsstrukturen (Intranet)
- Kategorien helfen solche Informationsstrukturen zu bauen.
- Die Informationsstrukturen, wie z.B. Thesauri, können fürs Retrieval genutzt werden (z.B. Synonym/Homonym-Problematik)

# Granularität in Informationsobjekten

- Granularität von Informationsobjekten in Informationsstrukturen?  
Welche Einheiten von Informationsobjekten?
- Dokumente, Kapitel, oder Paragraphen?
- Folgendes sollte man beachten:
  - Je kleiner das Informationsobjekt, desto höher der Verwaltungsaufwand
  - (je kleiner das Informationsobjekt, desto schwieriger mit statistischen Verfahren zu bearbeiten)



# Vordefinierte Informationsstrukturen

- Thesauri und Klassifikationen sind traditionelle Verfahren um Beziehungen zwischen Konzepten/Kategorien, Worten und Phrasen zu definieren.
- Thesauri werden oftmals benutzt für:
  - Indexierungszwecke (manuell oder automatisch)
  - Anfrageformulierung, verhindert dadurch ein „mismatch“ zwischen Anfrage- und Dokumentmerkmalen (auch automatisch, Anfrageerweiterungen)
- Klassifikationen sind in Bibliotheksumfeld populär:
  - Einordnen von Büchern und anderen Bibliotheksobjekten
  - Sehr bekannt von der Dezimalklassifikation (z.B Dewey)

# Thesaurus

## ■ Ein Thesaurus besteht aus:

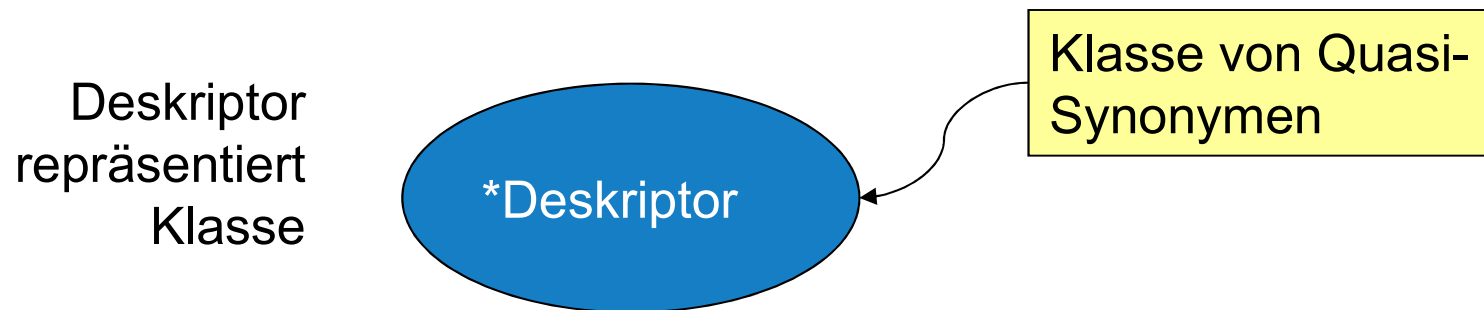
- kontrolliertem Vokabular (z.B Kategorien!)
- Beziehung zwischen den einzelnen Merkmalen des Vokabulars
- Ausserdem: Anmerkungen zu Merkmalen und deren Beziehungen

## ■ Wiederholung Synonym / Homonym:

- Synonym:
  - Zwei oder mehrere Wörter sind (von der Bedeutung) äquivalent zueinander
  - Mit anderen Worten: verschiedene Bezeichnungen für das gleiche Konzept
- Homonym:
  - Zwei oder mehr Wörter (im linguistischen Sinn!) sind Homonyme, falls sie gleich ausgesprochen oder geschrieben werden, aber unterschiedliche Bedeutung haben.
  - Mit anderen Worten: die gleiche Bezeichnung wird für unterschiedliche Konzepte gebraucht

# Deskriptoren vs. Nicht-Deskriptoren

- Bei Synonym-/Homonymproblemen muss eines der Merkmale als Deskriptor deklariert werden. Mit anderen Worten:



- Nicht-Deskriptoren: Rest der Klasse, die bei der Indexierung **nicht verwendet** werden darf.
- Indexierungssprache: Menge von Deskriptoren → kontrolliertes Vokabular

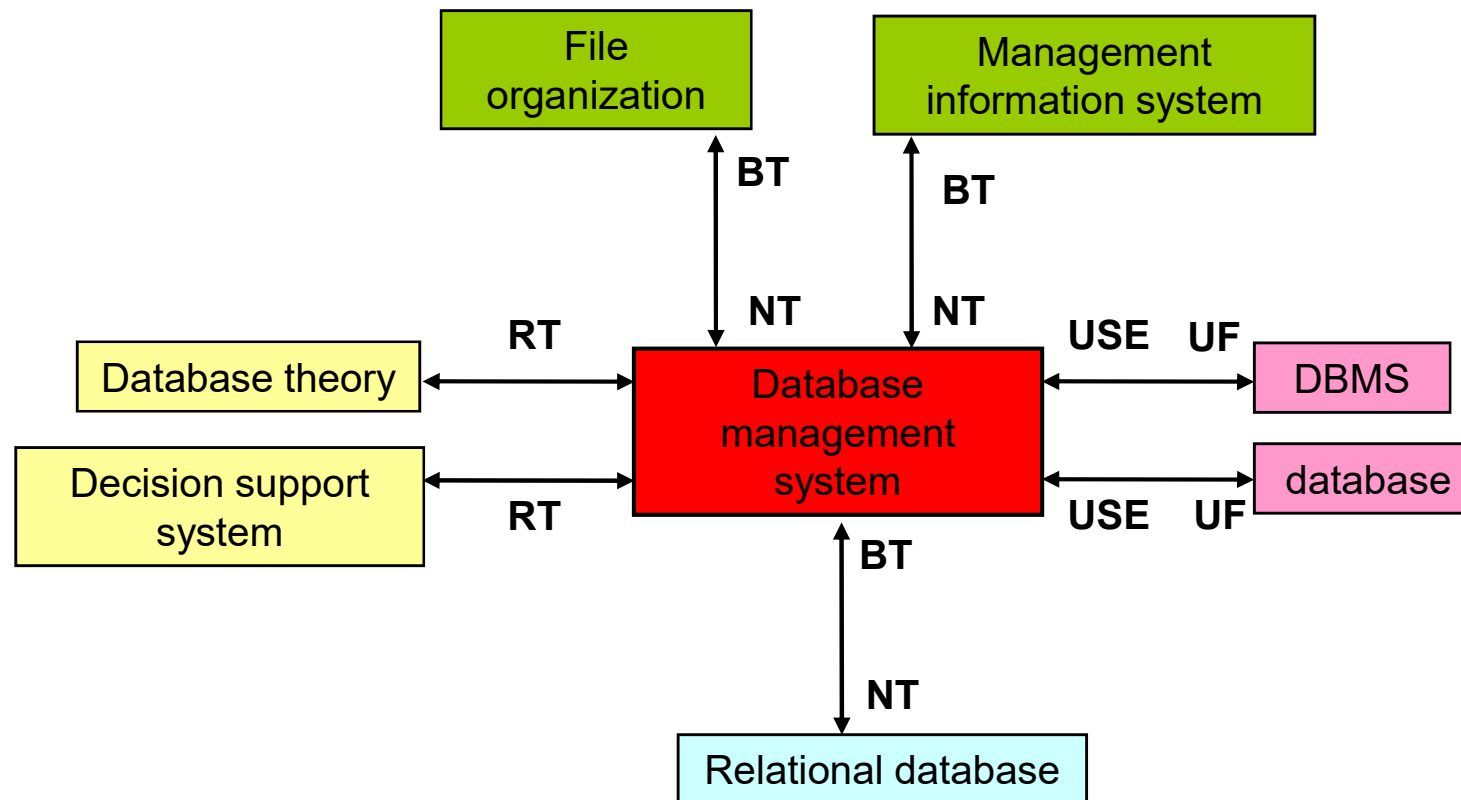
# Beziehungen in Thesauren

## ■ Die wichtigsten Beziehungen zwischen Deskriptoren sind:

- USE synonym use another term as descriptor
- UF used for use this term instead
- NT narrower term restricted concept
- BT broader term umbrella concept
- TT top term concept at head of hierarchy
- RT related term similar concept

## ■ Diese und andere Beziehungen werden gebraucht, um eine Diskursdomäne zu modellieren. Sie können ebenfalls gebraucht werden, um Suchmerkmale zu identifizieren, wenn Information gefunden werden soll.

# Beispiel aus einem Thesaurus (INSPEC)



# Herausforderung betreffs Thesauri

- Aufwand
- Konsistenz (manuell konstruierte Thesauri sind selten zyklensfrei)
- Unschärfe? (→ Transitivität?)
- Vollständigkeit?
- Bedeutungen sind abhängig von Zeit, Betrachter, thematischen Kontext. Wie global kann ein Thesaurus sein?
- Ein Thesaurus kann helfen, Begriffe konsistent zu verwenden.

# Beispiel Wordnet

- WordNet ist eine lexikalische Datenbank der Englischen Sprache
- Schauen Sie sich WordNet-Online an

<http://wordnetweb.princeton.edu/perl/webwn>

- Was für eine Struktur hat WordNet?
- Was erfährt man über das Wort „Virus“? Über "Bank"?
- In was für Szenarien kann ein Thesaurus beim Information Retrieval hilfreich sein?

# Thesauri und Unschärfe → Ähnlichkeitsthesaurus

- D: Sammlung von Informationsobjekten, Domain
- $\phi_1, \phi_2$ : Indexierungsmerkmale
- $\text{sim}(\phi_1, \phi_2)$  = Mass für die “Ähnlichkeit”
- Wie sieht die Informationsstruktur aus, die einen Ähnlichkeitsthesaurus darstellt?
- Welche Parameter würden Sie in die Ähnlichkeitsberechnung einfließen lassen und wie?
- Anwendung eines Ähnlichkeitsthesaurus?

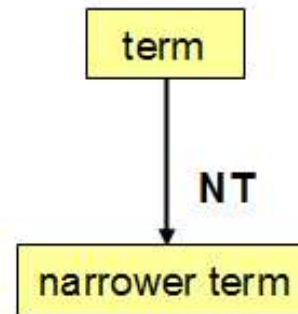


# Beispiel Ähnlichkeitsthesaurus

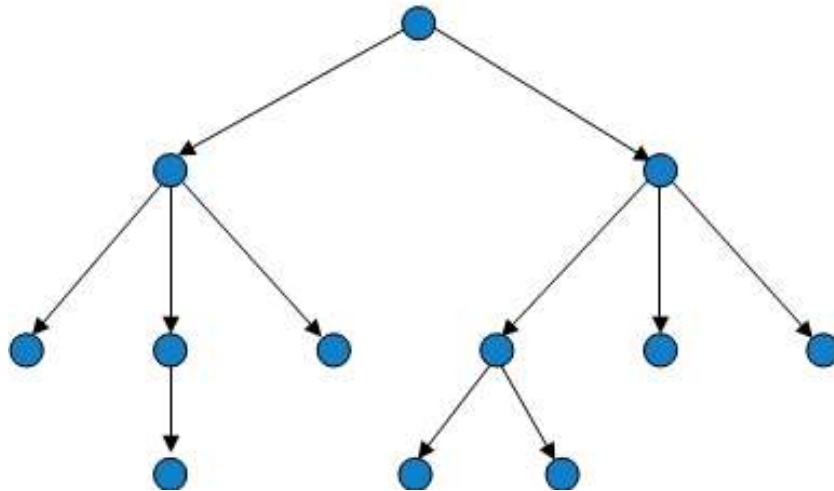
English → French: offer	French → German: incendie ( <i>fire</i> )
4.7494 offre ( <i>offer</i> )	0.6321 brand ( <i>fire</i> )
4.5737 offert ( <i>offered</i> )	0.5130 feuer ( <i>fire</i> )
4.2255 comptant ( <i>in cash</i> )	0.4223 feuerwehr ( <i>fire department</i> )
4.0475 pret ( <i>loan</i> )	0.4160 brandstiftung ( <i>arson</i> )
3.8256 opa ( <i>tender offer</i> )	0.3865 flammen ( <i>flames</i> )
3.7980 faite ( <i>done</i> )	0.3703 brandursache ( <i>cause of fire</i> )
3.7215 prevoit ( <i>calculates</i> )	0.3686 sachschaden ( <i>material damage</i> )
3.5656 echec ( <i>failure</i> )	0.3623 braende ( <i>fires</i> )
3.5602 intention ( <i>intention</i> )	0.2779 brannte ( <i>burnt</i> )
3.5171 engage ( <i>hire</i> )	0.2581 ausgebrochen ( <i>broke out</i> )

Beispiel aus Peters et al. (2012)

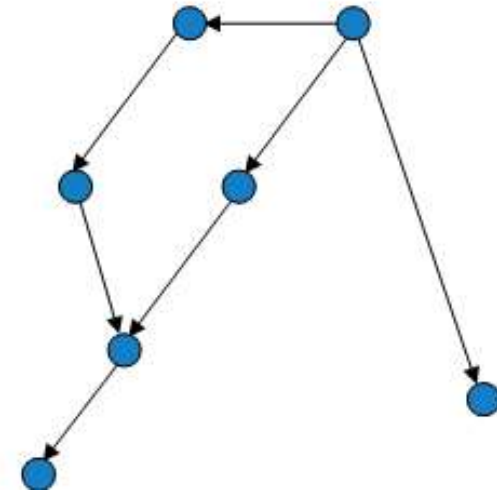
# Hierarchie (Thesaurus)



**Monohierarchy**

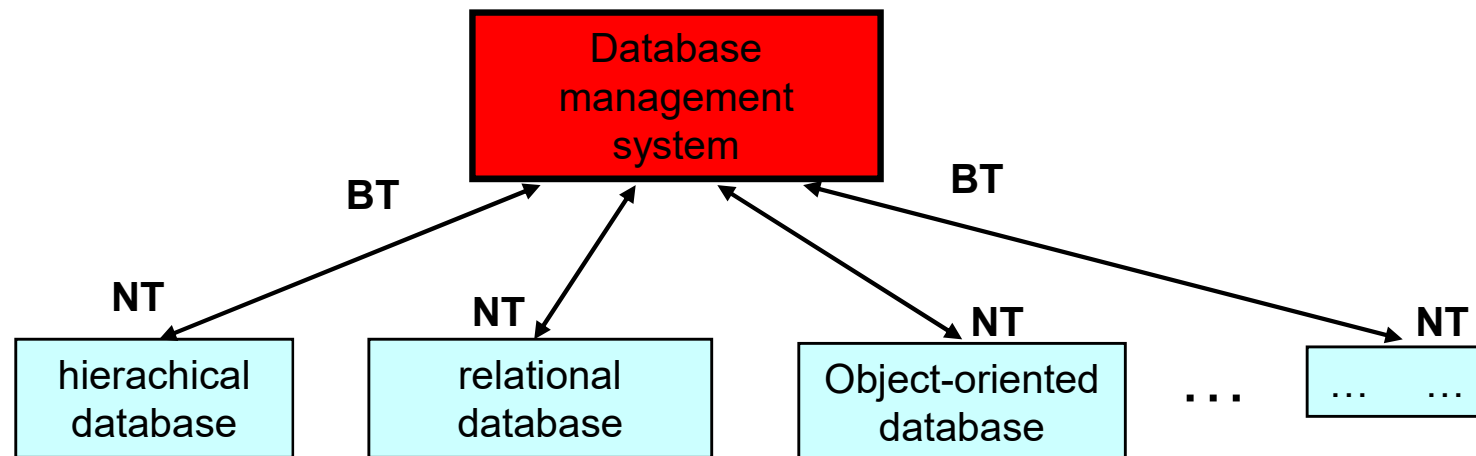


**Polyhierarchy**



# Klassifikation

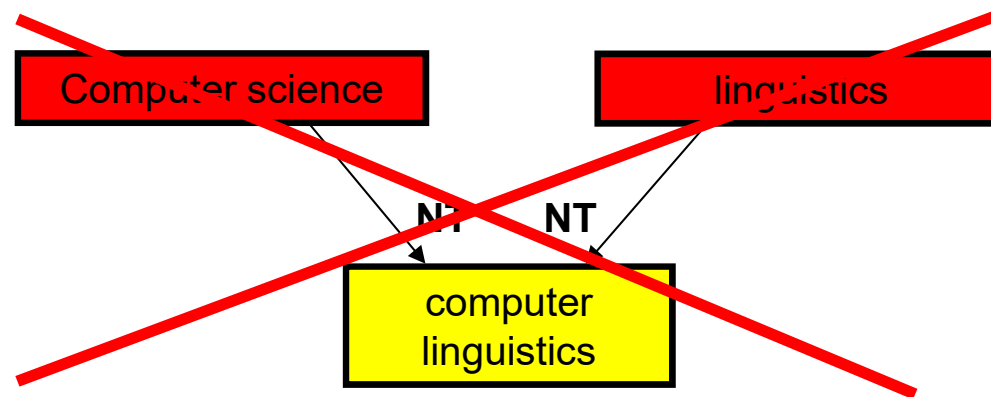
- Eine Klassifikation ist ein Thesaurus der nur NT (und BT) Beziehungen hat. Es handelt sich hier um eine Monohierarchie.
- Eine Klassifikation deckt eine Dokumentenkollektion im Allg. vollständig ab.



- Beachten Sie, dass es schwierig sein kann, zu entscheiden, zu welchem BT (broader term) ein NT (narrower term) zuzuordnen ist.

# Struktur einer Klassifikation

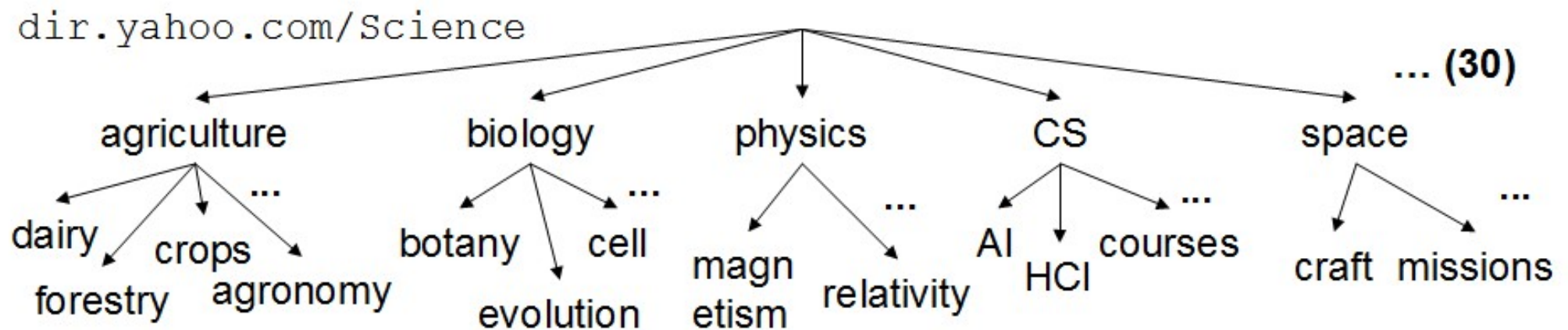
- Es gibt genau einen NT-Pfad der von einem „top term“ (TT) zu untergeordneten Termen zeigt.
- Das ist der Grund für die Schwierigkeiten z.B. bei der Darstellung von interdisziplinären Fachgebieten:





# Browsing - Webverzeichnis

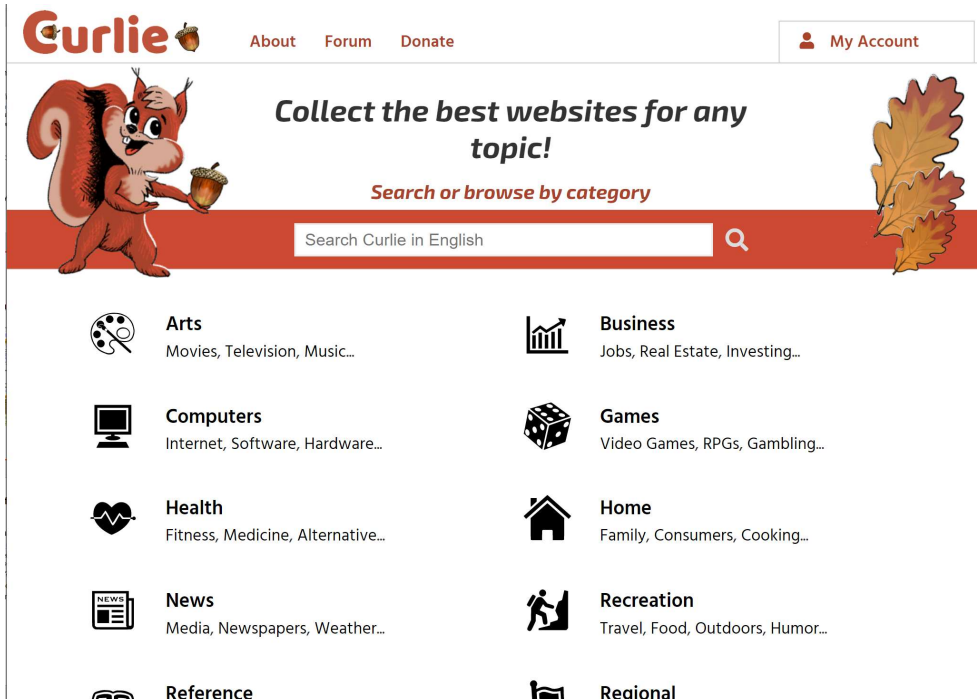
- Suchparadigma, bei dem man sich zum Resultat „durchhangelt“.
- Das Browsing beginnt mit einer Klassifikation.
  - Bei einem Webverzeichnis wird der Baum **statisch** aufgebaut und wird **manuell gepflegt** (→ hoher Verwaltungsaufwand).
  - Die Knoten sind meist **hierarchisch** angeordnet und besitzen entsprechende Unterknoten.



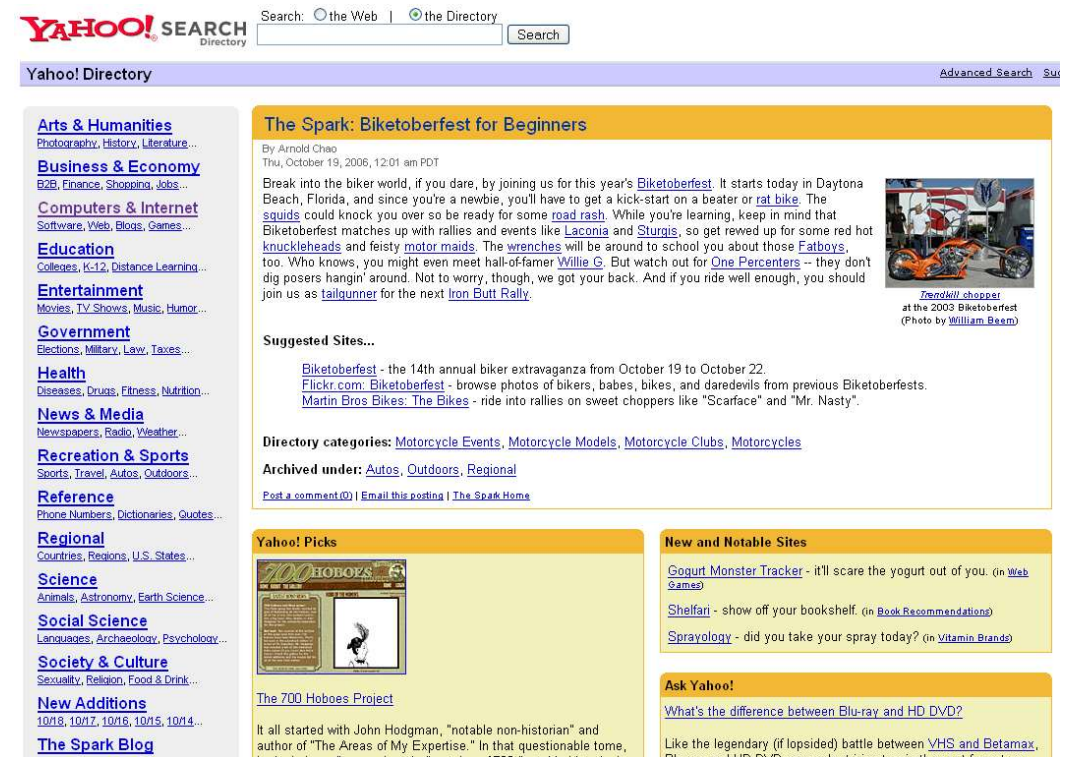
(Quelle: <http://www.stanford.edu/class/cs276a/handouts/lecture13.ppt>)

# Beispiele: Browsing

## ■ Curlie.org (ex DMOZ)



## ■ Yahoo! – einer der ganz grossen Player in der Frühzeit des Web





# Exkurs: Dewey Decimal Classification

- Schauen Sie sich ein Beispiel des Einsatzes von DDC an!

<http://edoc.ub.uni-muenchen.de/view/ddc/dewey.html>

- Beobachten Sie die Verteilung der Dokumente in den -Klassen!
  - Würden Sie die Themen ähnlich anordnen?
  - Wie könnten Sie den DDC für IR einsetzen?
- Versuchen Sie zu beschreiben, in welchen Situationen sie eher horizontal suchen sollten, und in welchen Sie hierarchische Strukturen zum Browsen verwenden sollten?

# Herausforderungen bzgl. Klassifikationen

- Aufwand für Verwaltung, wie erfasst man Unschärfe
- Vollständigkeit? (Wir treffen immer wieder den Eintrag “other” an)
- Bedeutungen ändern sich. → Eine Klassifikation muss sich ändern können, das hat grossen Einfluss auf die Verwaltung von Klassifikationen.
- Menschen klassifizieren unterschiedlich (auch “Profis”).
- Ein Mensch klassifiziert zu verschiedenen Zeiten unterschiedlich. (Bemerkung: “Beschlagwörter” werden manchmal auf ihre Konsistenz getestet und beurteilt, was halten Sie davon?)



# Clustering

## ■ Definition Clustering

- Clustering ist ein **Gruppierungsprozess**, bei dem eine Menge von Objekten in "Cluster" von ähnlichen Objekten geordnet wird

## ■ Anwendungsbeispiel: Resultat einer Suchanfrage wird durch Clustering-Algorithmen geordnet

- Diese Cluster sollen dem Benutzer helfen, effizienter in den gefunden Resultaten weiterzusuchen (d.h., die Information effizienter zu "verdauen")
- Navigations- und Visualisierungshilfe für den Benutzer
- Cluster können beliebige Untercluster besitzen
- Die Cluster werden durch die Algorithmen automatisch aufgebaut
- Benutzer/innen sehen «tiefer» in die Resultatliste hinein

# Beispiele: Clustering

## ■ Clusty, Yippy: RIP

The screenshot shows the Clusty search engine interface. At the top, there's a search bar with 'clustering' entered. Below the search bar, there are tabs for 'clusters', 'sources', and 'sites'. The 'clusters' tab is selected, showing a list of search results. The first result is 'Clustering' with a description: 'Performance, availability & ROI load balance without sticker shock. - www.coyotepoint.com'. Below this, there are several other results, including 'Software: Clustering and Segmentation (Data Mining)', 'Online Software for Clustering', and 'Neural Clustering Segmentation Software - SOM and Lift Chart'. The interface is blue and white, with a sidebar on the left containing various categories like 'Engine, Search', 'Computing', 'Linux', 'High Availability', 'Windows', 'Hierarchical', 'Definition', 'Load Balancing', and 'Database'.

## ■ Kartoo: RIP



Heute vor allem noch in Enterprise Search

# Strukturierungsstufen von Informationsstrukturen

- Implizit: Intuition, Bilder, Video, Audio, “da liegt was in der Luft, ich kann es nicht benennen” (keine IS)
- Explizite Konzepte: Wortlisten, Phrasen, Texte, Glossar, kontrollierte Vokabulare
- Explizite Konzepte in expliziter Verbindung: Hypertext, netzartige Strukturen, Objekte und Eigenschaften/Attribute, (Hypertexte, Navigationsstrukturen, Ähnlichkeitsthesauri, Dokumentencluster,...)
- Explizite Konzepte, explizite Verbindungen, und Inferenzen (Konsistenzbedingungen und Regeln die neue Konzepte oder neue Verbindungen generieren): Thesauri, Inferenznetzwerke, “Ontologien”, OO-Datenmodelle mit Vererbung, Datenstrukturen+Code, UML
- Explizite Konzepte, explizite Verbindungen, Inferenzen/Regeln und “Wahrheit”, d.h. Bewertungen von Aussagen in einer Logik.

# Schlussfolgerungen

## Kategorisierung/Thesauri

- Kategorien sind (kontrollierte) Mengen von Objekten, die u.a. das Retrieval von Information, das Routing, die Indexierung und die Sortierung vereinfachen können.
- Kategorien können anhand von Regeln, statistischen Modellen oder vordefinierten Poly/Mono-Hierarchien gebildet werden.
- Thesauri reflektieren die Beziehungen zwischen den Konzepten/Deskriptoren. Manuelle Erstellung ist extrem arbeitsintensiv und teuer.
- Wir unterscheiden zwischen Deskriptoren und Nicht-Deskriptoren, so genannte „lead-in terms“.
- Klassifikationen bestehen nur aus NT-Relationen. Diese sind eher statisch und hauptsächlich im Bibliotheksumfeld zu finden.

# Social Tagging

- Social Tagging beschreibt den Prozess, bei welchem Benutzer Metadaten, in Form von Schlüsselwörtern, hinzufügen und diese Daten mit anderen Nutzern teilen.
- Definition Tag:
  - "Aufkleber" (vergleiche: Deskriptor), mit dem ein Informationsobjekt näher (inhaltlich) beschrieben wird
- Definition Tagging:
  - Benutzer fügt Tags zu einem Informationsobjekt hinzu und teilt diese ggf. mit anderen Benutzern

# Social Tagging

- Wird auch „*collaborative tagging*“ genannt (wenn auf Communities aufgeteilt)
- „*Folksonomy*“ (von "folk taxonomy") wird das entstehende Vokabular aus häufigen Tags genannt
- Wurde geprägt durch Flickr und Del.icio.us
- Hat sich heute breit durchgesetzt, vor allem in Form von «Hashtags»: Twitter, Instagram etc.

# Social Tagging - Eigenschaften

- **Spontane** und offene Gemeinschaften
- Keine fixen Indexierungsregeln
- Kein kontrolliertes Vokabular
- Keine Hierarchie, keine Beziehung zwischen Tags
- Eine „Underground“-Alternative zur Kategorisierung
- Vergleiche Streben nach Konsistenz bei Kategorisierung
- Analog zum Verhältnis Verschlagwortung vs. Volltextindexierung
- Keine Klassifizierung durch Experten (→jeder kann Taggen, sowohl Benutzer als auch Ersteller)

# Social Tagging - Modell

## ■ Ressourcen

- Die Verlinkung zwischen Ressourcen ist ein gut untersuchtes Fachgebiet (PageRank/HITS).

## ■ Tags

- Verknüpfungen zwischen Benutzern und Ressourcen

## ■ Benutzer

- Analyse von Beziehungen zwischen Benutzern in sozialen Netzwerken.

- → In unserem Fall sind die Ressourcen oft Informationsobjekte



# Social Tagging - Dimensionen

Dimension	Hauptkategorie	Zusammenfassung potentieller Verbindungen
Tagging-Rechte	Selbst-Tagging, Genehmigungs-Tagging, Frei-für-Alle-Tagging	Art und Typ der resultierenden Tags; Rolle der Tags im System
Tagging-Support	Blind, Vorgeschlagen, Sichtbar	Konvergenz in Folksonomies oder Übergewichtung von Tags
Aggregationsmodell	Bag, Menge	Verfügbarkeit von Aggregationsstatistiken
Objekttypen	Text, Nicht-Text	Art und Typ von resultierenden Tags
Quellen der Materialien	Benutzerbeitrag, System, global	Verschiedene Anreize, Art und Typ von resultierenden Tags
Verbindung von Informationsobjekten	Links, Gruppen, keine	Konvergenz bei ähnlichen Tags für verlinkte Informationsobjekte
Soziales Netzwerk	Links, Gruppen, keine	Konvergenz in lokalen Folksonomies

Quelle: Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead, <http://www.danah.org/papers/WWW2006.pdf>

# Social Tagging - Anwendungsbeispiele

## ■ Beispiele:

- Hashtags: Instagram, Twitter
- Flickr
- Youtube
- Etc.

## ■ «#» war «Wort des Jahres 2014» in der Schweiz

## ■ Anwendungsbeispiele findet man praktisch ausschliesslich im Web. Dies hat folgende Gründe:

- Das Web ist ein guter "Brutplatz" für spontane Benutzergruppen
- Es gibt keine Instanz, die für die Verschlagwortung verantwortlich ist
- Es hat schlicht zu viel Content für eine Kategorisierungsinstanz
- Die Beziehungen werden erst mit einer gewissen kritischen Masse stabil und sinnvoll

## ■ Beliebte visuelle Darstellung von Tags



# Frage



## ■ Was fällt ihnen dabei auf?

.net advertising **ajax** apple architecture **art** article audio **blog** blogging **blogs** book **books** **business** cms comics community  
computer cool **css** culture daily **design** **development** diy download downloads **education** entertainment fashion fic film  
firefox **flash** flickr  fonts **food** forum **free** **freeware** **fun** funny game **games** geek **google** **graphics** gtd hardware  
health history **howto** html **humor** illustration images **imported** **inspiration** **internet** japan japanese **java** **javascript**  
jobs language library lifehacks links **linux** literature **mac** magazine maps marketing math media microsoft mobile movies mp3  
**music** **news** online **opensource** **osx** **photo** **photography** photos photoshop **php** podcast **politics** productivity  
**programming** python radio rails recipes **reference** religion **research** resources rss **ruby** **science** **search** **security**  
seo sga shop **shopping** slash **social** **software** sports tech **technology** **tips** tool **tools** toread **travel** **tutorial**  
tutorials tv ubuntu usability **video** videos **web** **web2.0** **webdesign** **webdev** wiki **windows** wordpress work writing xml  
youtube



# Frage



- Tag Cloud nicht mit klassischen Informationsstrukturen verwechseln!



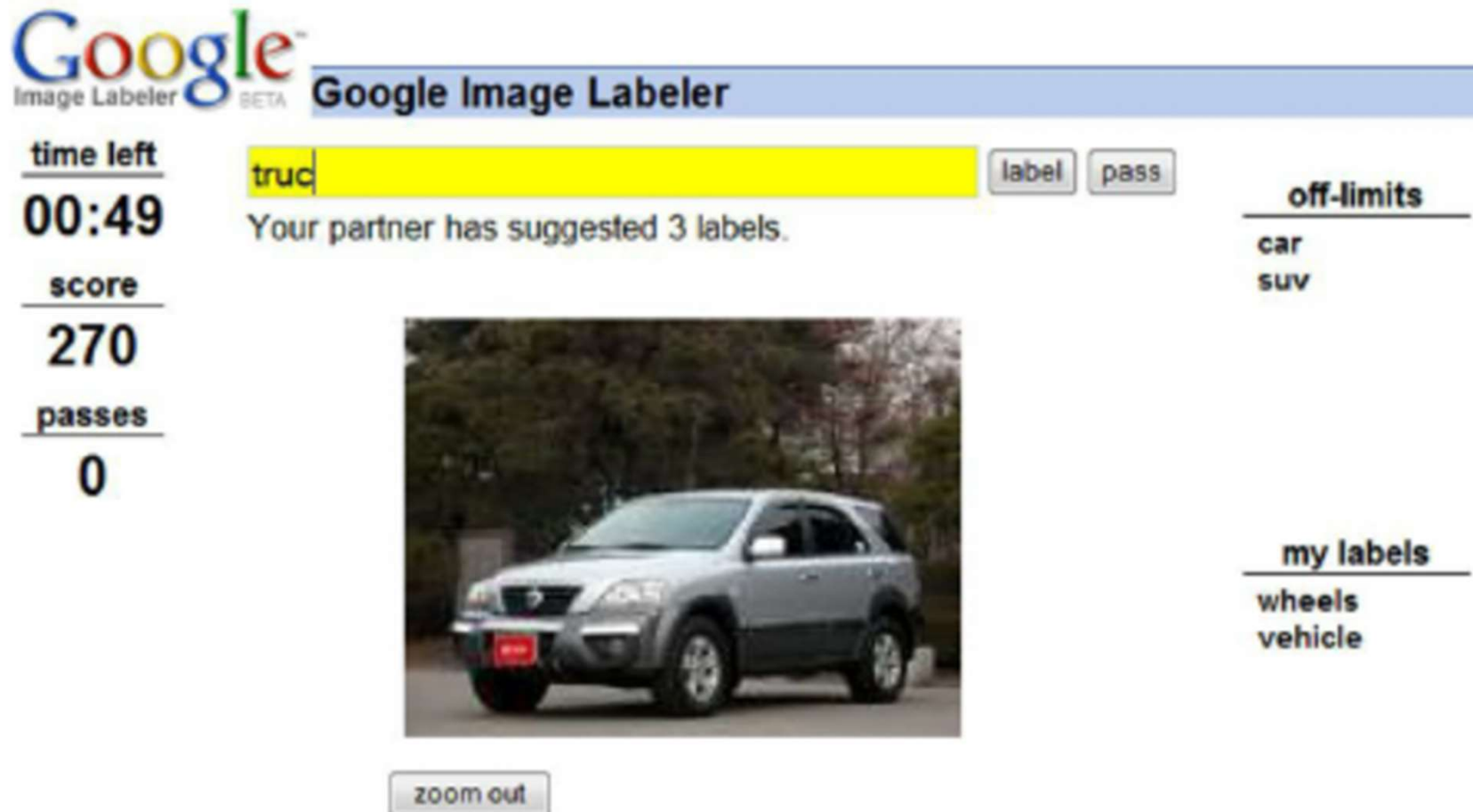
# Social Tagging – Vorteile für Betreiber

- Nutzer kategorisieren selber (nehmen aktiv an der Klassifizierung teil), Tags reflektieren das Benutzervokabular
- Benutzer hat Einfluss auf andere Benutzer
- Betreiber braucht keine Experten für die Auswahl der Metadaten (Tags)  
→ Benutzer erstellen Metadaten selbst
- Flexibel auf Veränderungen (kann auch Nachteil sein)
- Bessere Abdeckung als mit kontrolliertem Vokabular
- Extrem kostengünstig im Vergleich zur herkömmlichen Kategorisierung
- Stärkere Bindung des Benutzers an das Angebot (mehr involviert)

# Social Tagging – Vorteile für Benutzer

- Leichteres Auffinden der eigenen Informationsobjekte
- Leichteres Auffinden von gemeinsamen Informationsobjekten, die mehrere Benutzer teilen (z.B. Konzertfotos/Urlaubsortfotos bei Instagram)
- Anreiz schaffen für andere Benutzer, die eigenen Informationsobjekte finden
- Spiel und Wettbewerb
  - <http://www.gwap.com> (ehemals <http://www.espgame.org/>) – nicht mehr online
  - <http://images.google.com/imagelabeler> - nicht mehr online
- Selbstdarstellung
- Ausdruck der eigenen Meinung

# Google Image Labeler



Quelle: Robertson, Vojnovic, Weber, CHI 2009



# Social Tagging – Nachteile

- Zersplitterung der Kategorien durch unkontrolliertes Vokabular, Singular/Plural, Schreibfehler
- Homonymieproblematik durch unkontrolliertes Vokabular (Mehrdeutigkeit)
- Synonymproblematik durch unkontrolliertes Vokabular (Mehrdeutigkeit)
- Mangelnde Struktur der Metadaten
- Meist nur Einworttags möglich (kein Kompositaunterstützung)
- Mangelnde Präzision (und Ausbeute?)
- Ungenaue Deskriptoren
- Spam

# Ausblick Social Tagging

- Tag Clustering / Tag Bündelung
- Metadaten auf Tags
- Visualisierung von Tags (Tagmaps / Geotagging)
- Gemeinsame Tags in einer Community werden zu einem Thesaurus