



# Text Mining

Michael Granitzer  
mgrani@know-center.at



<http://www.know-center.at/swat>

# Inhalt

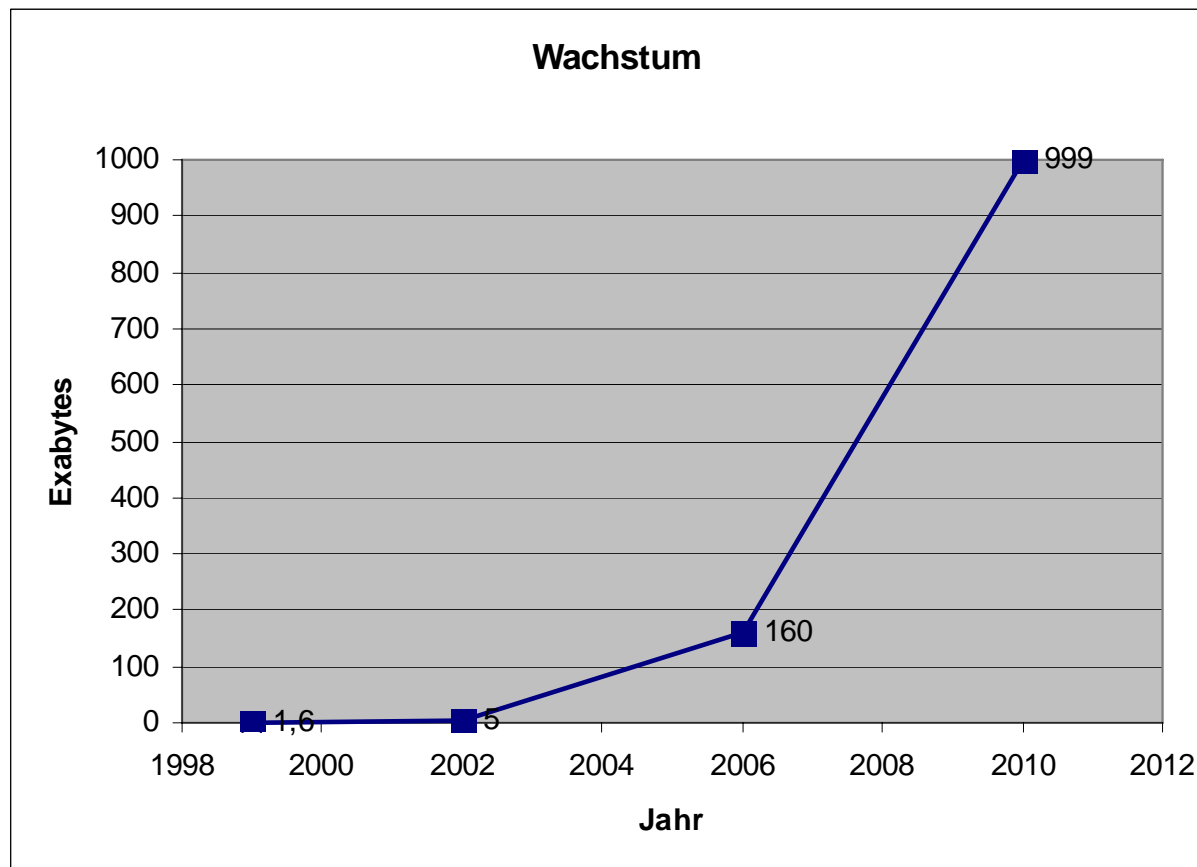
---

- Ein paar Zahlen zur Motivation
- Vorverarbeitung von Texten
- Statistische Analysen und Ähnlichkeit zwischen Dokumenten (VSM, LSI)
- Maschinelle Lernmethoden im Überblick
  - ◆ Textklassifikation (Rocchio, k-NN, SVM)
  - ◆ Clustering, Automatische Gruppierung von Texten (K-Means, HAC)
- Evaluierungskriterien

# Ausgangssituation

## Zahlen und Fakten I

- Wie viel Information umgibt uns?



<http://www.know-center.at>

# Ausgangssituation

## Zahlen und Fakten II

---

- 🌐 5 Exabytes: Alle Wörter die jemals von Menschen gesprochen wurden (2002)
- 🌐 Status 2006: 161 Exabytes an digitaler Information
  - ◆ 6 Tonnen an Bücher pro Einwohner der Erde
  - ◆  $\frac{3}{4}$  davon sind Kopien von Originalinhalten
  - ◆ Videos und Bilder werden immer wichtiger
  - ◆ Das Internet als Hauptursache für das Wachstum
  - ◆  $\frac{1}{4}$  der Information wird in Unternehmen erzeugt
  - ◆ 95% der Information ist unstrukturiert
  - ◆ In Unternehmen „nur“ 80% unstrukturiert

# Ausgangssituation

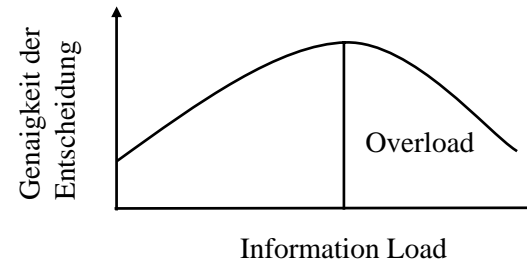
## Zahlen und Fakten II

---

**Aber:** Die Fähigkeit des Gehirns Information zu verarbeiten hat sich seit 500 Jahre nur geringfügig verbessert.

# Ausgangssituation

- Decision Making Performance (laut [Eppler 02])



- Änderung der Erwartungshaltung
  - ◆ Geringe Antwortzeit (e.g. E-Mails)
  - ◆ Entscheidungen beruhen auf ungenügender bzw. nicht relevanter Information
- Einfluss auf die physische und mentale Gesundheit

# Technische Lösungsansätze

---

- Verringerung der Informationsmenge
- Erhöhung der Informationsqualität
- Verbesserter und intelligenter Zugang zu Information

➤ Semantische Technologien

**Vision:** „Information at your fingertips“

**Problem:** unstrukturierter Informationsträger „Text“, Komplexes Problem des Sprachverstehens

**Fragestellung:**

- Erhöhung der semantischen Qualität von unstrukturierten Textdaten?
- Automatische Strukturierung von Information möglich?
- Schaffen von einfachen Zugängen zu komplexen Informationsräumen

# Definition Text Mining

---

*„Text Mining is the **discovery** by computer of **new, previously unknown information**, by automatically extracting information from different **written resources** „ [Hearst 1999]*

*→ Fokus liegt auf der Analyse von Inhalten (i.A. Text)*

# Von was gehen wir aus?

---

## ● Informationsobjekte mit textuellen Inhalten

- ◆ Dokumente
- ◆ Web-Seiten

## ● Informationsraum: Menge von Informationsobjekten

- ◆ Strukturiert z.B. mittels Taxonomien, Hyperlinks etc.
- ◆ Unstrukturiert

# Anwendungsgebiete

---

- Automatisches annotieren von Dokumenten
- Spam Filter
- Wartung von Klassifikationsschemata wie DMOZ
- Information Retrieval
- Ontology Learning from Text
- Visualisierung von Informationsräumen

# Inhalt

---

- Ein paar Zahlen zur Motivation
- **Vorverarbeitung von Texten**
- Statistische Analysen und Ähnlichkeit zwischen Dokumenten (VSM, LSI)
- Maschinelle Lernmethoden im Überblick
  - ◆ Textklassifikation (Rocchio, k-NN, SVM)
  - ◆ Clustering, Automatische Gruppierung von Texten (K-Means)
- Evaluierungskriterien

# Vorverarbeitung von Text

Inhalt eines Informationsobjektes/Dokumentes

---

## 🌐 Format des Objektes

- ◆ Text
- ◆ HTML/Word/PDF/PPT
- ◆ XML/SGML

## 🌐 Inhalt

- ◆ Folge von Zeichenketten

## 🌐 Metadaten

- ◆ Beschreibung des Informationsobjektes anhand unterschiedlicher Kriterien

## 🌐 Struktur/Aufbereitung des Inhalts

- ◆ Überschriften, Absätze, Kapitel

# Vorverarbeitung von Texten

## Beispiel Informationsobjekt

Format: PDF

Inhalt

Metadaten:

- Autor
- Schlüsselwörter
- Kategorie
- Erstellungsdatum
- Dateigröße

Struktur

- Überschrift
- Kapitel
- Literaturverzeichnis

**WebRat: Supporting Agile Knowledge Retrieval through Dynamic, Incremental Clustering and Automatic Labelling of Web Search Result Sets**

Michael Granitzer  
KnowCenter Graz  
mgrani@know-center.at

Vedran Sabol  
KnowCenter Graz  
vsabol@know-center.at

Wolfgang Kienreich  
KnowCenter Graz  
wkien@know-center.at

**Abstract**

*WebRat is an interactive system for visualizing and refining search result sets. Documents matching a query are dynamically clustered on the fly and visualised as a contour map of islands. Thematic clusters are built, analysed, and visualised in real time. Users can interactively explore the visualisation and refine queries by selecting from the keywords and clusters presented to them. WebRat does not rely on precalculated meta data. Instead, necessary information is directly extracted from query result representations provided by search engines, as for example ranked lists of document snippets. The system is language-independent, does not require a dedicated server machine and can be adapted to a number of data sources and visualisation modes easily. WebRat supports agile knowledge retrieval by transforming unstructured information input into a representation enriched with structure and meta information.*

**1. Introduction**

Today's standard web search interfaces display many similarities in user interface as well as in technical detail: Users type in one or more textual query terms and are then presented with a ranked list of matching documents in decreasing order of relevance, based on a full-text search of the query terms in a given data set. While easy to use, implement and maintain, such an approach features a number of drawbacks which renders it less useful in a

results if the result set is large, and the manifold topical dimensions of the result are hidden from the user.

Recently, many proposals have tried to address these issues, by enriching unstructured repositories with meta-data, or by introducing structures like topic maps and ontologies to represent topical interconnections and, in general, support search operations. While such approaches work well in clearly specified areas like the environmental domain, where rich meta-data is already available, they fail in situations where annotation or structuring of information entities is complicated or not possible at all.

The WebRat retrieval and visualisation system was designed to address the problem named. WebRat provides a framework capable of:

- querying various web data sources (in the fashion of a metasearch engine),
- merge results from data sources differing strongly in structure and content (i.e. web pages, email, newsgroups, databases)
- dynamic, incremental clustering of search results by topic;
- automatically extracting keywords describing topics and using these as cluster labels;
- interactive visualisation of results and topics in a number of ways.

The system does not require any precalculated information, as all necessary operations are done on the fly, based on search results as they arrive. All calculations can be performed on standard office machines.

# Vorverarbeitung von Text

## Überblick

---

Ziel: Überführung von Informationsobjekte in eine für Algorithmen verarbeitbare Form

- Sammeln von Dokumenten (Gatherer, Spider)
- Formatnormalisierung (e.g. PDF→Text, Word→Text)
- Lexikalische Analyse (Tokenization)
- Tokenanalyse (optional)
  - ◆ Lemmatisierung (Wortstammanalyse)
  - ◆ Linguistische Analyse (e.g. Nomen, Verben)
  - ◆ Strukturanalyse (e.g. Sätze, Absätze)
  - ◆ Informationsextraktion (IE, e.g. Personenerkennung)
- Merkmalsgenerierung und Gewichtung

Ergebnis:

- Eine Menge von Merkmalen/Features für jedes Dokument
- Merkmalsraum (Feature Space) für einen Informationsraum

# Vorverarbeitung von Text

## Crawling

---

Ziel: Sammeln einer Repräsentativen Dokumentmenge

- Crawler/Spider im Web
- Verfolgung von Hyperlinks ausgehend von einer Seed URL
- Ausnutzen von Graph Strukturen

# Vorverarbeitung von Text

## Formatnormalisierung

---

Ziel: Extraktion der relevanten Textteile aus gegebenen Informationsressourcen

- Trivial für bekannte „strukturierte“ Formate
- Nicht-Trivial im Web Kontext
  - ◆ Interpretation aktiver Inhalte (Java Script)
  - ◆ Was sind die relevanten Textteile?

Vc  
For

HOME

RESEARCH CENTERS

- + Java Standard Edition
- + Java Enterprise Edition
- + Java Micro Edition

Development Tools

- Application Management
- Data Access Tools
- Gaming Tools

Web Development Frameworks

- Security & Testing
- Java Application Servers
- Profiling and Monitoring
- Reporting

W

SITE RESOURCES

- Featured Tutorials
- News & Reviews
- Forums
- Podcasts
- Newsletters
- White Paper Library
- Web resources
- RSS Feeds

CAREERS

PARTNER SITES

- Demo.com
- LinuxWorld.com
- NetworkWorld.com

ABOUT US

SPONSORED LINKS

See your link here.

RS: Site Performance

Download the Application Delivery Performance Report to Learn More.

1. 5

T

simi

User

pres

decr

of th

impl

num

Report for IT Consultants: Learn How IT Consultants Can Profit From Clients' Innovation Needs.

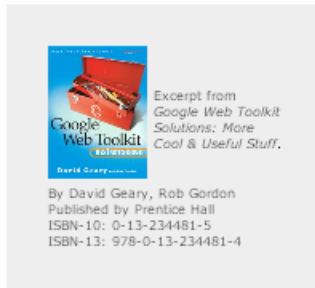
DEVELOPMENT TOOLS

## GWT solution #6: Drag and drop

Use the Google Web Toolkit to roll your own drag-and-drop module

By David Geary and Rob Gordon, JavaWorld.com, 11/06/07

You're not alone if you think drag and drop in Java Web applications is a drag. Fortunately, we found a solution in *Google Web Toolkit Solutions: More Cool & Useful Stuff*, forthcoming from Prentice Hall. In this excerpt, authors David Geary and Rob Gordon show you how to implement a drag-and-drop module that you can easily incorporate into your Java Web applications. Along the way you'll also learn about GWT modules, composite widgets, and widget event handling.



Excerpt from *Google Web Toolkit Solutions: More Cool & Useful Stuff*.

By David Geary, Rob Gordon  
Published by Prentice Hall  
ISBN-10: 0-13-234481-5  
ISBN-13: 978-0-13-234481-4

The ultimate in user interactivity, drag and drop is taken for granted in desktop applications but is a litmus test of sorts for Web applications: If you can easily implement drag and drop with your Web application framework, then you know you've got something special.

Until now, drag and drop for Web applications has, for the most part, been limited to specialized JavaScript frameworks such as *Script.aculo.us* and *Rico*. No more. With the advent of GWT, we have drag-and-drop capabilities in a Java-based Web application framework. Although GWT does not explicitly support drag and drop (drag and drop is an anticipated feature in the future), it provides us with all the necessary ingredients to make our own drag-and-drop module.

In this solution, we explore drag-and-drop implementation with GWT. We implement drag and drop in a module of its own so that you can easily incorporate drag and drop into your applications.

### Stuff you're going to learn

This solution explores the following aspects of GWT:

- Implementing composite widgets with the `Composite` class
- Removing widgets from panels
- Changing cursors for widgets with CSS styles
- Implementing a GWT module
- Adding multiple listeners to a widget
- Using the `AbstractPanel` class to place widgets by pixel location
- Capturing and releasing events for a specific widget
- Using an event preview to inhibit browser reactions to events

See *Google Web Toolkit Solutions: More Cool & Useful Stuff* Solution 1 and Solution 2 for more in-depth discussions of implementing GWT modules and implementing composite widgets, respectively.

[Continued](#)

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | [Next >](#)

- [Print](#)
- [Feedback](#)
- [Feedback](#)
- [Add to del.icio.us](#)

Best of JavaWorld

Editor's Choice

### From the Network World Data Center:

Inside three enterprise SOAs  
What SOA means for management  
10 best practices for your SOA  
SOA, meet SOI

center . graz  
**Know**

**TUG**

### FEATURED WHITEPAPERS

### NEWSLETTER SIGN-UP

Sign up for our technology specific newsletters.

Enterprise Java  
[View all newsletters](#)

Email Address:

### Sponsored Links

Get your **FREE 30 day VMware Workstation trial Now!**  
Virtualize your desktop today with VMware Workstation.

Related Article

# Vorverarbeitung von Text

## Lexikalische Analyse - Tokenization

**Ziel:** Zerlegen eines Textes in atomare, sinnvolle Einheiten welche weiter verarbeitet werden können.

Zeichenkette:

*"In diesem Seminar erhalten Sie wertvolle Tipps, wie das optimale Kosten-/Nutzenverhältnis durch gezielte Automatisierung der Metadaten-Extraktion erzielt werden kann."*

Beispiele für mögliche Tokens:

- Word-Grams:  
"In", "diesem", "Seminar", "erhalten", "Sie", "wertvolle", "Tipps", ",",  
"wie", "das", "optimale"....
- Word Gruppen (Word n-Grams):  
"In diesem", "diesem Seminar", "Seminar erhalten", "erhalten Sie"...
- Character n-Grams (hier Länge 3):  
"In ", "n d", " di", "die", "ies", "ese", "sem"...

# Vorverarbeitung von Text

## Tokenanalyse-Lemmatisierung

---

**Ziel:** Ermitteln von Eigenschaften und Bedeutungen eines Tokens

Lemmatisierung

- Reduktion eines Terms (i.e. Wort) auf gemeinsame Formen/Stämme
- Gleiche semantische Bedeutung, jedoch andere Syntax
- Suffix Stripping:
  - ◆ "Book" vs. "Books" → Book
  - ◆ "Manager", "Management", "managing" → "Manag"
  - ◆ "Relative" vs. "Relativity" → "Relativ"
- Root Stemming (morphologische Analyse):
  - ◆ "Haus" vs. "Häuser" → "Haus"
  - ◆ "gehen", "ging", "gegangen" → "gehen"
  - ◆ Komplizierter, benötigt Wörterbuch
- Phoneme

# Vorverarbeitung von Text

## Tokenanalyse-Satzgrenzenerkennung

---

**Ziel:** Ermitteln von Eigenschaften und Bedeutungen eines Tokens

Satzgrenzenerkennung:

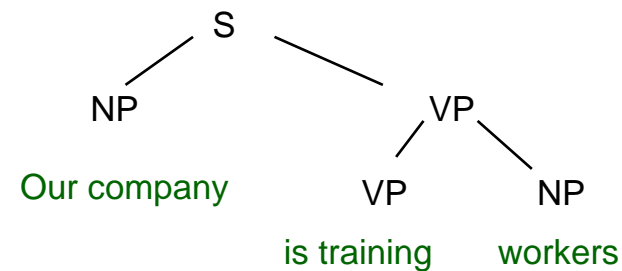
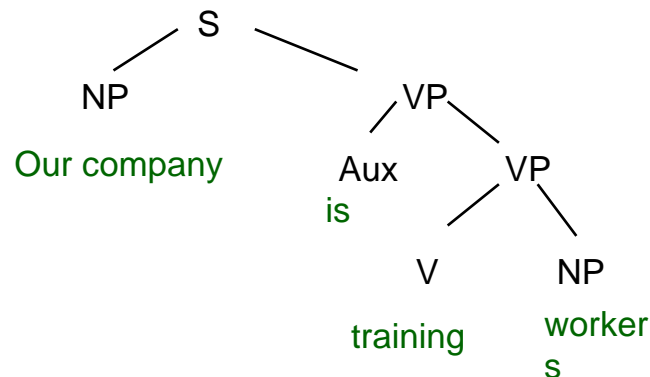
- "Im Rahmen des Vortrags am 18. November 2005 werden Themen wie z.B. Clustering behandelt."
- Welcher Punkt trennt einen Satz?
- Ansätze:
  - ◆ Manuelle Regeln bzw. regulären Ausdrücken
  - ◆ Über Klassifikationsverfahren
- Genauigkeit Domänenabhängig
- Für Zeitungstext über manuelle Regeln ca. 90%
- Über maschinelle Klassifikationsverfahren ca. 98%
  - ◆ Aber: Trainingsbeispiele nötig!

# Vorverarbeitung von Text

## Tokenanalyse- Part of Speech Tagging

Ziel: Identifikation von Eigenschaften von Wörtern

- Zuordnung von Wortformen (e.g. Nomen, Verben, etc.)
- Anzahl der Unterschiedlichen Wortformen definiert durch sogn. Tag Set
- Parsing-Bäumen: z.B. „Our company is training worker“



aus C. Manning, H. Schütze: Foundations of stat. NL Processing

<http://www.know-center.at>

# Vorverarbeitung von Text

## Parsing Bäume

---

“Our company is training workers”

### Mögliche Parsing Bäume

- [SUBJ Our company] [VP [VGROUPE is training] [OBJ workers]]
- [SUBJ Our company] [VP is [COMPL [V training] workers]]
- [SUBJ Our company] [VP is [COMPL [ADJ training] workers]]

### Äquivalente Syntax mit anderer Bedeutung

- Our problem is training workers (2. Parse)
- Our product is training wheels (3. Parse)

Siehe <http://l2r.cs.uiuc.edu/~danr/Teaching/CS598-05/Lectures/Lec2-intro.pdf>

# Vorverarbeitung von Text

## Tokenanalyse-Part of Speech Tagging

---

- Problem: Mehrdeutigkeit
- Laut Manning/Schütze 455 verschiedene Parse-Bäume für den Satz:  
"List the sales of the products produced in 1973 with the products produced in 1972,"
- Vollständiges Textverstehen ist ungelöstes Problem
- Für die meisten Anwendung reichen meist einfache Unterscheidungen (e.g. Nomen, Verben, Adjektive)  
→ Shallow Text Processing
- Bsp. Heuristik f. Deutsch  
großer Anfangsbuchstabe → Hauptwort

# Vorverarbeitung von Text

## Ein kurzes Beispiel

„Ein kurzes Beispielchen.“

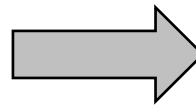
- Lexikalische Analyse:  
{„Ein“, „ „, „kurzes“, „ „, „Beispielchen“, „.“}
- Tokenanalyse
  - ◆ Lemmatisierung: {„Ein“, „kurz“, „Beispiel“, „.“}
  - ◆ Satzgrenzenerkennung:  
{„Ein“, „kurz“, „Beispiel“, [„.“;EOL]}
  - ◆ Part-of-Speech Tagging  
{[„Ein“;UART],[„kurz“;ADJ],[„Beispiel“;N],[„.“;EOL;PUNCTU  
ATION]}
- Zerlegung von Text in atomare Einheiten
- Grundlage für die weitere Verarbeitung

# Vorverarbeitung von Texten

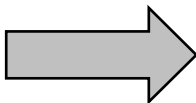
## Informationsextraktion

### Informationsextraktion (IE)

- „Füllen von vorgegebenen Tabellen“
- Überführung von unstrukturierten Text in strukturierte Vorlagen



Vorname	Nachname	Zugehörigk.	E-Mail
Michael	Granitzer	Know-Center	<a href="mailto:mgrani@know-center.at">mgrani@know-center.at</a>
Wolfgang	Kienreich	Know-Center	<a href="mailto:wkien@know-center.at">wkien@know-center.at</a>
Vedran	Sabol	Know-Center	<a href="mailto:vsabol@know-center.at">vsabol@know-center.at</a>



Firmenname	Firmenort	Rechtsform
Know-Center	Graz	?

# Vorverarbeitung von Texten

## Informationsextraktion

---

Zwei unterschiedliche Ansätze

- Grammatiken und reguläre Ausdrücke
- Maschinelles Lernen

Zusätzlich zur bisherigen Vorverarbeitung: Anwendung von Gazetteers, Thesauri und Ontologien

- Typisierung eines Tokens (e.g. Michael ist ein Vorname)
- Regeln unter Einbeziehung der Typisierung

Beispiel Strukturen:

- Personennamen:
  - Präsident John F. Kennedy = {Titel}{Vorname}{Nachname}
- Datum:
  - 4.11.2005, 4. November 2005 = {Zahl}{punkt}{Zahl}{punkt}{Zahl}, {Zahl}{punkt}{N:type=Monat}{Jahr}
- Problem Mehrdeutigkeit: „John F. Kennedy“, „Paris“

# Vorverarbeitung von Texten

## Informationsextraktion

---

Auflösen der Mehrdeutigkeit durch Wortkontext:

- [Person] arbeitet in [Firma]
- [Person] geboren am [Datum]

Nachteil bei Regeln/Lookup Listen:

- Nicht alles in Gazetteers abgebildet
- Mehrdeutigkeiten
- Wartung von Listen

Beispiel maschinelles Lernen:

- Vorgabe von Beispielen
- Training
- Vorhersagen von Namen auf Basis der trainierten Beispiele

Nachteil: Vorgabe von Trainingsbeispielen notwendig

# Vorverarbeitung von Texten

## Informationsextraktion

---

„Ein kurzes Beispiel von Michael Granitzer“

```
{[„Ein“;UART], [„kurz“;ADJ], [„Beispiel“;N], [von;ADV], [„Michael“,N] [Granitzer,N] [„.“;EOL;PUNCTUATION]}
```

- Anwenden einer Vornamen Lookup Liste

```
{[„Ein“;UART], [„kurz“;ADJ], [„Beispiel“;N], [von;ADV], [„Michael“,N,lookuptype=Vorname] [Granitzer,N] [„.“;EOL;PUNCTUATION]}
```

- Regel: if (Token==Vorname && Token+1.PartOfSpeech==N)  
Token+1.lookuptype=Nachname  
Token+1.entitytype = Person  
Token.entitytype = Person
- Aus anderer Perspektive: Person = [„Michael“,N] [Granitzer,N]

# Vorverarbeitung von Texten

## Informationsextraktion

---

Hauptaufgaben im IE:

- Named Entity Recognition (NE)
- Co-Reference Resolution (CO)

Eigenschaften:

- Sekundenbereich (je nach Länge/Komplexität)
- Genauigkeit abhängig von Domäne, Textkorpus, Sprache etc.
- NE ~ 95%, CO ~65%, TE ~80%, TR ~75%, ST ~60%

Anwendungen

- Metadatenextraktion
- Satzbewertungen
- Information Retrieval

# Vorverarbeitung von Texten

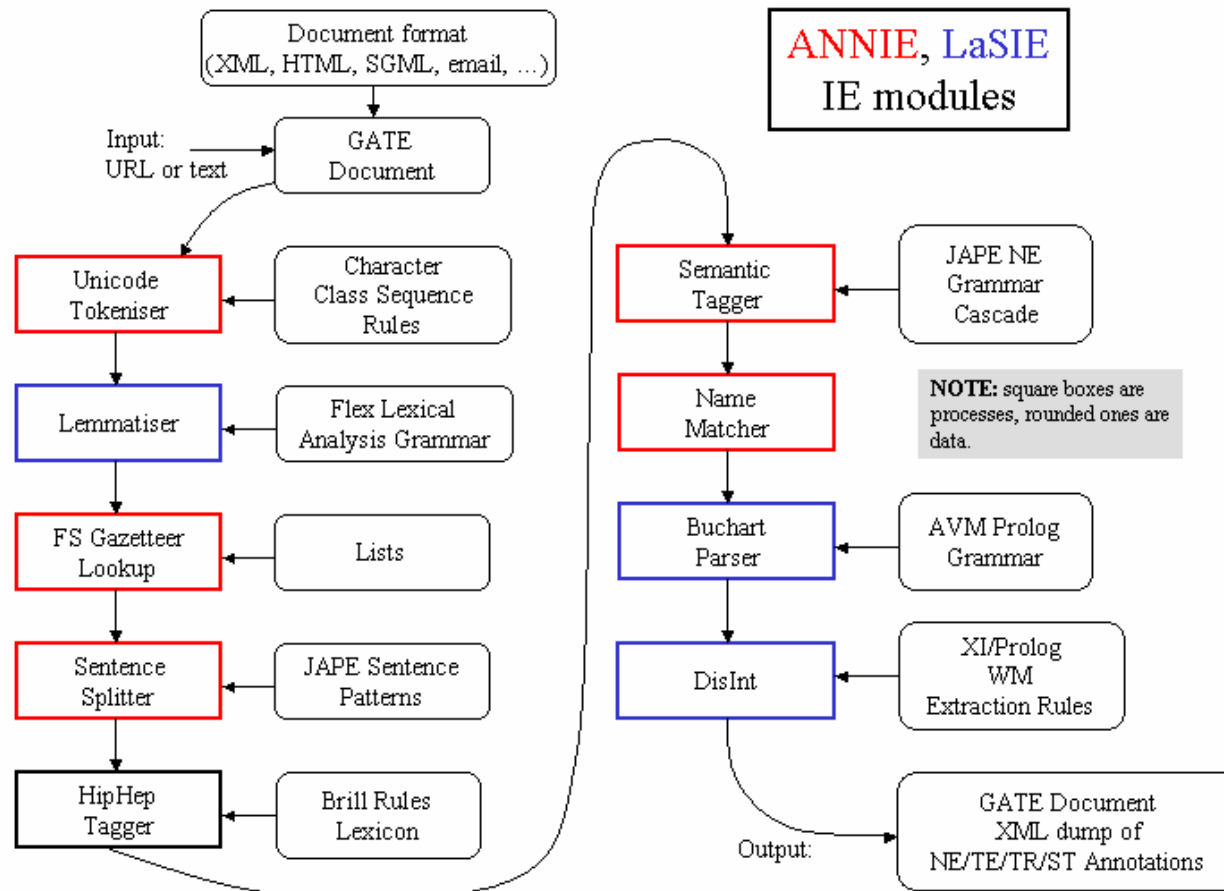
## Open Source Tools

---

- GATE, A General Architecture for Text Engineering (<http://gate.ac.uk>)
- Open Source Text Engineering Framework der Universität Sheffield
- Stanford NLP Toolkit
- ...

# Vorverarbeitung von Texten

## Beispiel: Gate ANNIE Architektur



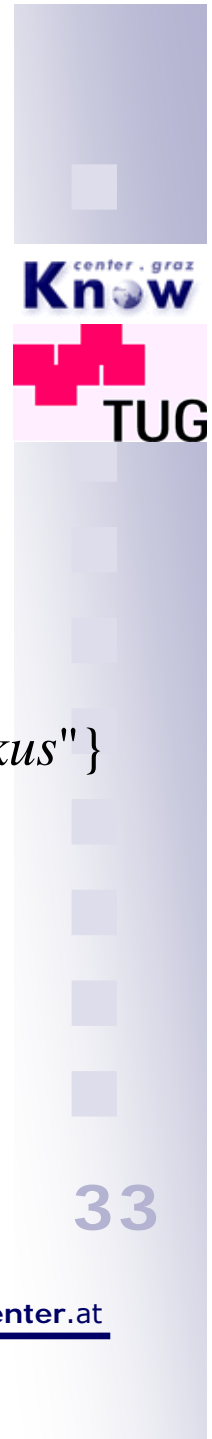
# Inhalt

---

- Ein paar Zahlen zur Motivation
- Vorverarbeitung von Texten
- **Statistische Analysen und Ähnlichkeit zwischen Dokumenten (VSM, LSI)**
- Maschinelle Lernmethoden im Überblick
  - ◆ Textklassifikation (Rocchio, k-NN, SVM)
  - ◆ Clustering, Automatische Gruppierung von Texten (K-Means)
- Evaluierungskriterien

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Motivation



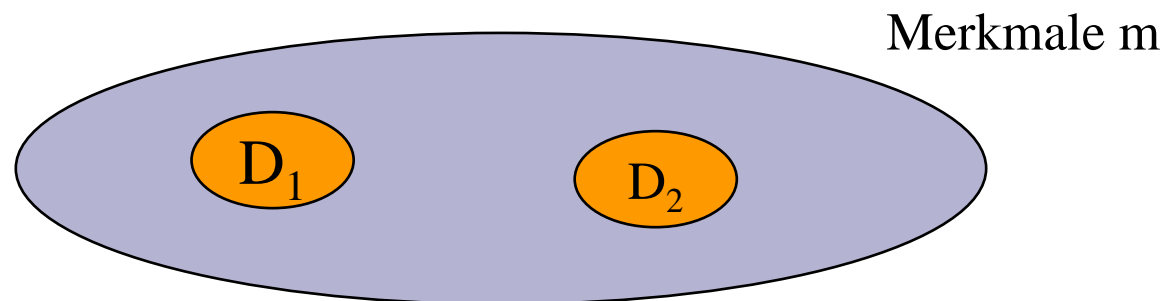
### Ergebnis d. Vorverarbeitung

- ◆ Pro Dokument eine Menge von  $n$  Merkmalen

$$D_j = \{m_1, m_2, m_3 \dots m_{k-1}, m_k\}$$

$$D_1 = \{ \text{"Michael"}, \text{"Granitzer"}, \text{"Vorlesung"} \dots \text{"Strohmaier Markus"} \}$$

- ◆ Alle Dokumente beschrieben in einem  $m$  dimensionalen Merkmalsraum



<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Motivation

### Ziel:

- Erzeugen von verschiedenen Merkmalen für Informationsobjekte
- Informationsraums  $\Leftrightarrow$  Merkmalsraum
- Mathematisches Modell f. Berechnungen
- Statistische Auswertung/Merkmalshäufigkeiten
- Ähnlichkeitsberechnung

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Merkmalshäufigkeit

---

### Zählen von Vorkommnissen von Merkmalen

- Welche statistischen Eigenschaften hat Sprache?
- Wie können diese für den Praxiseinsatz ausgenutzt werden?

### Anwendung:

- Erkennen von Zusammenhängen zwischen Merkmalen
- Selektion von relevanten Merkmalen
- Verbesserung der Qualität der Vorverarbeitung

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Merkmalshäufigkeit

Auswerten von Häufigkeiten

Beispiel: Häufigsten deutschen Wörter

Rang	Wort	Frequenz
1	UND	0.08427
2	DIE	0.05390
3	DER	0.05383
4	IN	0.02164
5	WIR	0.01676
6	ZU	0.01564
7	FÜR	0.01536
8	SIE	0.01306
9	VON	0.01285
10	DEN	0.01208
11	DES	0.01131
12	IST	0.01068

<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Zipf's Gesetz

Das Verhältnis der Häufigkeit des Auftretens eines Tokens ist invers proportional zu seiner Position in der Häufigkeitsliste ( $f \cdot r = \text{const}$ )

Das 30. Wort kommt 3x häufiger vor als das 90. Wort

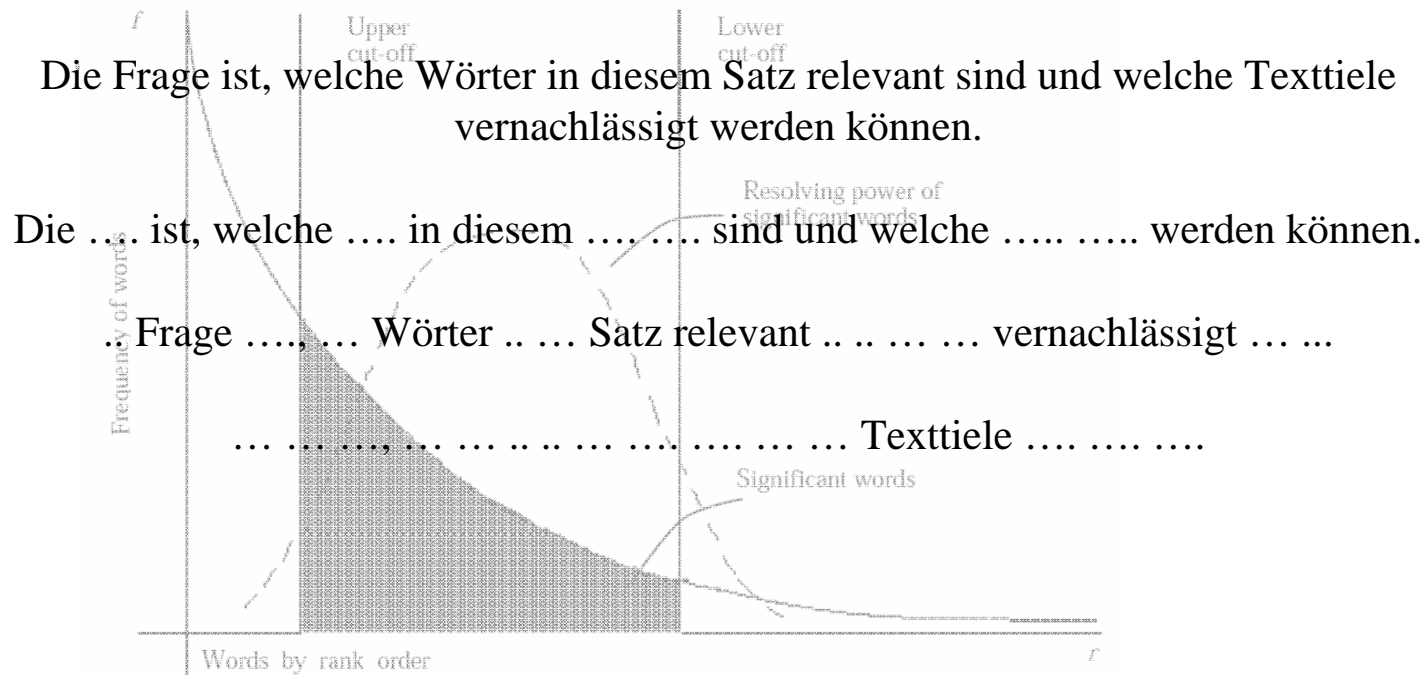
Rang	Wort	Frequenz	$f \cdot r$
1	UND	0.08427	$1 \cdot 0.08427 = 0.08427$
2	DIE	0.05390	$2 \cdot 0.05390 = 0.10780$
3	DER	0.05383	$3 \cdot 0.05383 = 1.6149$
4	IN	0.02164	$4 \cdot 0.02164 = 0.08656$
5	WIR	0.01676	$5 \cdot 0.01676 = 0.08380$
.	.	.	.
.	.	.	.
.	.	.	.
1000	Universität	0.00008427	$1000 \cdot 0.00008427 = 0.08427$

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Zipf's Gesetz

Das Verhältnis der Häufigkeit des Auftretens eines Tokens ist invers proportional zu seiner Position in der Häufigkeitsliste ( $f \cdot r = \text{const}$ )

Das 30. Wort kommt 3x häufiger vor als das 90. Wort



aus C. J. Rijsbergen, Information Retrieval

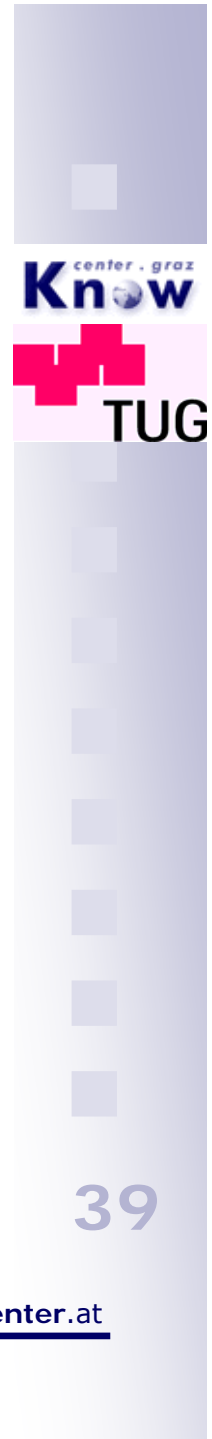
<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Merkmalshäufigkeit Anwendung

---

- „Language Guessing“ auf Zeichnungsebene
- Automatische Generierung von Stopwortlisten: Suche nach „und“ macht wenig Sinn (ca. 377.000.000 Ergebnisse bei Google)
- Reduktion der Merkmalsmenge auf Merkmale mit Informationsgehalt



# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Kollokationen

- Finden von sinnvollen Wortgruppen, so genannten Collocations
- Definition: *[A collocation is defined as] a sequence of two or more consecutive words, that has characteristics of a syntactic and semantic unit, and whose exact and unambiguous meaning or connotation cannot be derived directly from the meaning or connotation of its components. [Chouekra, 1988]*
- Großer Korpus notwendig
- Beispiel aus Manning/Schütze: Frequenz von Kollokationen  
Wort-Bigramme, einfache Tokenisierung  
Experiment mit 3 Monaten Text der "NewYork Times,,

Rang	Häufigkeit	1. Wort	2. Wort
1	80871	of	the
2	58841	in	the
3	26430	to	the
....			
15	11429	New	York
16	10007	he	said

<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Kollokationen

Verbesserung über Berücksichtigung von POS Verfahren [Justeson & Katz, 1995]

- Filtern nach vorgegebenen POS Mustern (z.B. NN, AN)

Rang	Häufigkeit	1. Wort	2. Wort	Muster
1	11487	New	York	NN
2	7261	United	States	NN
..				
4	3301	last	year	AN
..				
12	1942	Saddam	Hussein	NN
..				
17	1328	oil	prices	NN

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Kollokationen

- Verbesserung über Berücksichtigung von POS Verfahren [Justeson & Katz, 1995]
- Filtern nach vorgegebenen POS Mustern (z.B. NN, AN)

Rang	Häufigkeit	1. Wort	2. Wort	Muster
1	11487	New	York	NN
2	7261	United	States	NN
..				
4	3301	last	year	AN
..				
12	1942	Saddam	Hussein	NN
..				
17	1328	oil	prices	NN

<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Co-Occurrence Analyse

- Fragestellung: Welche Wörter/Merkmale kommen mit welchen  
Wörter/Merkmale oft vor?
- Erstellen einer Co-Occurrence Matrix
- Zelle enthält Häufigkeit des Vorkommnisses eines Merkmals
- Auswahl jener Pärchen, welche statistisch Signifikant oft  
vorkommen
- Anwendung: Automatisches Vorschlagen von Suchbegriffen  
Suche nach Windows liefert auch Treffer für Linux
- Probleme: Geschwindigkeit, Speicherverbrauch

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Co-Occurrence Analyse, Beispiel

„Linux als Open Source Betriebssystem“

„Das Betriebssystem Windows....“

Zusammenhang: Windows  $\Leftrightarrow$  Betriebssystem  $\Leftrightarrow$  Linux

Häufigkeiten	Windows	Linux	Betriebssystem	Open Source
Windows	-	0	10	2
Linux	0	-	9	11
Betriebssystem	10	9	-	3
Open Source	2	11	3	-

<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Gewichtung von Merkmalen

---

### Gewichtung von Merkmalen

Ziel: Bezogen auf die Anwendung sinnvolle Reduktion der Merkmale auf Basis statistischer Modelle

Wie aussagekräftig ist ein Merkmal?

Ist „Michael Granitzer“ ein aussagekräftiges Merkmal für einen Text?

- Abhängig von Anwendung
- Abhängig von der Domäne
- Abhängig vom Informationsraum

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Gewichtung von Merkmalen

---

Wichtigkeit eines Merkmales hängt i.A. ab von

- Wie oft kommt ein Merkmal/Wort in einem Dokument vor
- Wie oft kommt ein Merkmal in allen Dokumenten vor
- In wievielen Dokumenten kommt ein Merkmal/Wort vor
- Term Frequency Inverse Document Frequency (TFIDF)

Repräsentation eines Dokumentes als numerischer Vektor

Anwendung von Vektorrechnung

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Gewichtung von Merkmalen - Mathematisch

### Binäre Merkmalsgewichtung

$$w_{i,j} = \begin{cases} 1 & \text{merkmal}_i \in \text{Dokument}_j \\ 0 & \text{merkmal}_i \notin \text{Dokument}_j \end{cases} \quad \vec{d}_j = \langle 0,1,0 \dots 0,1,0 \rangle$$

### Term Frequency Inverse Document Frequency (TFIDF)

$$w_{i,j} = TF_{i,j} * f(DF_i)$$

$$TF_{i,j} = \frac{|\text{merkmal}_i \cap \text{Dokument}_j|}{|\text{Dokument}_j|}$$

$$DF_j = \frac{|\forall_j \text{merkmal}_i \in \text{Dokument}_j|}{|\text{Dokument}_j|}$$

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Vektorraum Modell - Mathematisch

- Überführung der Dokumente in Vektoren durch Gewichtung der Merkmale eines Dokumentes

$$\vec{d}_1 = \langle w_{1,1}, w_{1,2} \dots w_{1,n-1}, w_{1,n} \rangle$$

$$\vec{d}_2 = \langle w_{2,1}, w_{2,2} \dots w_{2,n-1}, w_{2,n} \rangle$$

⋮

$$\vec{d}_m = \langle w_{m,1}, w_{m,2} \dots w_{m,n-1}, w_{m,n} \rangle$$

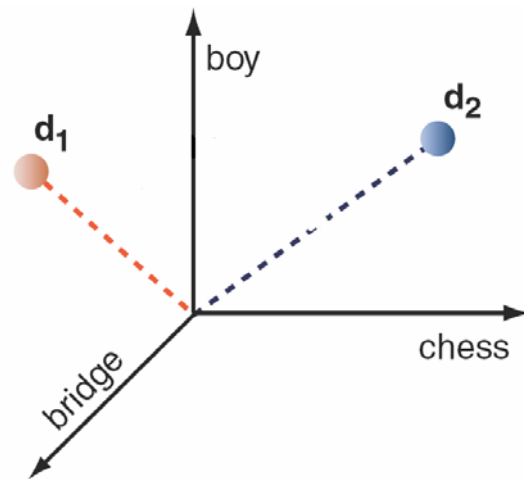
$$\vec{d}_1 = \langle 0.2, 0.4, 0.5, 0 \rangle$$

$$\vec{d}_2 = \langle 0.0, 0.4, 0.0, 0 \rangle$$

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Vektorraum Modell

- Darstellung im Vektorraummodell (Vector Space Model)



- In der Praxis umfasst der Termraum 100.000 Dimensionen und mehr

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Vektorraum Modell - Mathematisch

### Dokument Term Matrix

$$D_{m \times n} = \left\{ \begin{array}{cccccc} w_{1,1} & w_{1,2} & \cdots & w_{1,n-1} & w_{1,n} \\ w_{2,1} & w_{2,2} & & w_{2,n-1} & w_{2,n} \\ \vdots & & \ddots & & \vdots \\ w_{m-1,1} & w_{m-1,2} & & w_{m-1,n} & w_{m-1,n} \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n-1} & w_{m,n} \end{array} \right\}$$

	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$
$d_1$	8	0.4	0.3	0	0	0
$d_2$	0	0	0	0.1	0.1	0.1
$d_3$	0	0.1	0.1	0	0	0
$d_4$	1	0.2	0.4	3	4	

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Ähnlichkeiten

---

Ähnlichkeit ermöglicht das Ordnen von Dokumenten

- Jedes Dokument hat eine Menge von Merkmalen
- Je mehr Merkmale übereinstimmen, umso ähnlicher sind Dokumente

Problem: Ähnlichkeit wiederum abhängig von Anwendung/Erwartung des Benutzers

- Metadatenbezogene (e.g. Personen) Ähnlichkeit (z.B. Merkmale aus IE)
- Inhaltsbezogene Ähnlichkeit (z.B. Merkmale aus Nouns)

Ähnlichkeit zwischen Dokumenten ist entscheidend für viele Algorithmen

- Relevance Ranking (IR)
- Query by Example
- Klassifikation
- Clustering

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Ähnlichkeiten

Ähnlichkeit ist proportional der Menge der übereinstimmenden Merkmale

Hammingmetrik =  $\frac{\# \text{ gemeinsame Merkmale}}{\# \text{ Alle Merkmale}}$

Beispiel:

- D1 = {Semantic, Web}
- D2 = {Textanalyse, Semantic, Web}
- D3 = {RDF, Technologie, Web}
- Hamming(D1,D2)=2/3
- Hamming(D1,D3)=1/4
- Hamming(D2,D3)=1/5

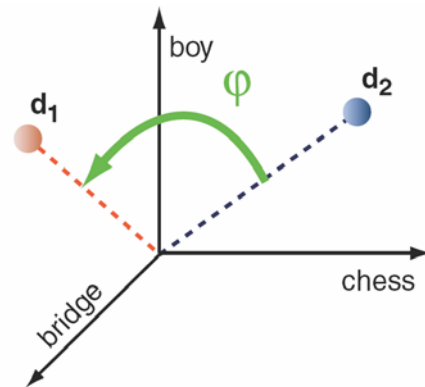
Wichtigkeit/Häufigkeit von Merkmalen wird nicht berücksichtigt

Vergleich von Schlüsselwörtern

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Cosinusähnlichkeit

- Vektorraum Modell: Winkel zwischen Vektoren entspricht Ähnlichkeit (Cosinusmaß)
- Häufig eingesetzt, einfach, liefert gute Ergebnisse
- Beispiel:



- Problem: Annahme, dass Merkmale voneinander unabhängig sind stimmt nicht

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

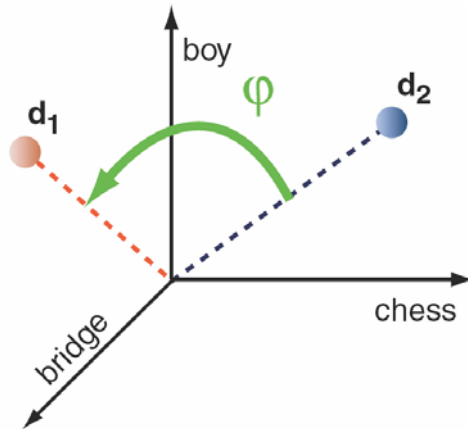
## Cosinusähnlichkeit - mathematisch

- Skalarprodukt (arithmetische Formel)

$$\vec{x} \cdot \vec{y} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

- Cosinusmaß = Winkel zwischen Query und Dokumentvektor

$$\text{sim}(d_m, q) = \frac{\vec{d}_m \cdot \vec{q}}{|\vec{d}_m| \times |\vec{q}|} = \frac{\sum_{i=1}^x w_{i,m} \times w_{i,q}}{\sqrt{\sum_{i=1}^x w_{i,m}^2} \times \sqrt{\sum_{i=1}^x w_{i,q}^2}}$$



# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Vector Space Model

---

### Vorteile:

- Schnell und einfach
- Erstellung des VSM erfolgt in  $O(n)$
- Auch „ähnliche“ Dokumente werden gefunden
- Sortierung nach Grad der Ähnlichkeit
- In der Regel bessere Ergebnisse als Hamming (wgn. Termgewichtung)

### Nachteile:

- Unabhängigkeitsannahme der Terme
- Relativ willkürliches Ähnlichkeitsmaß bezogen auf natürlichsprachliche Texte
- Berücksichtigung des Kontextes

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## Latent Semantic Indexing/Latent Semantic Analysis

---

### Ausgangspunkt

- Terme i.d.R. nicht unabhängig
- Co-Occurrence (gemeinsames auftreten) von Termen gibt Aufschluss über Synonyme/Polyseme Begriffe

### Ziel:

- Transformation der Dokumentvektoren vom hochdimensionalen Termvektorraum in einen **Konzeptvektorraum** niedrigerer Dimensionalität
- Ausnutzen von Korrelationen zwischen Termen zur Identifikation von Synonymen (z.B. „Web“ und „Internet“ häufig zusammen)
- Ausnutzen von Korrelationen zwischen Termen zur Identifikation von Polysemen (z.B. „Java“ mit „Library“ vs. „Java“ mit „Kona Blend“ vs. „Java“ mit „Borneo“)

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## LSI Mathematisch

Ausgangspunkt Dokument Term Matrix

$$D_{m \times n} = \left\{ \begin{array}{ccccc} w_{1,1} & w_{1,2} & \cdots & w_{1,n-1} & w_{1,n} \\ w_{2,1} & w_{2,2} & & w_{2,n-1} & w_{2,n} \\ \vdots & & \ddots & & \vdots \\ w_{m-1,1} & w_{m-1,2} & & w_{m-1,n-1} & w_{m-1,n} \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n-1} & w_{m,n} \end{array} \right\}$$

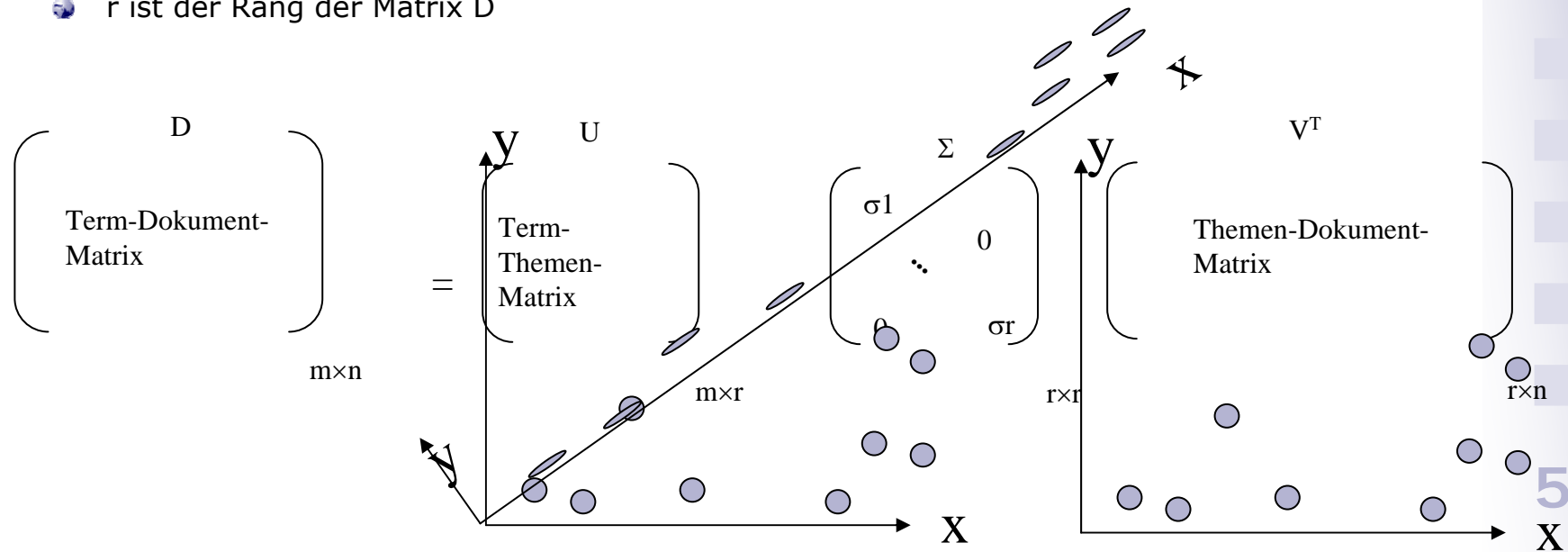
Frage: Wie kann diese zerlegt werden?

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## LSI Mathematisch

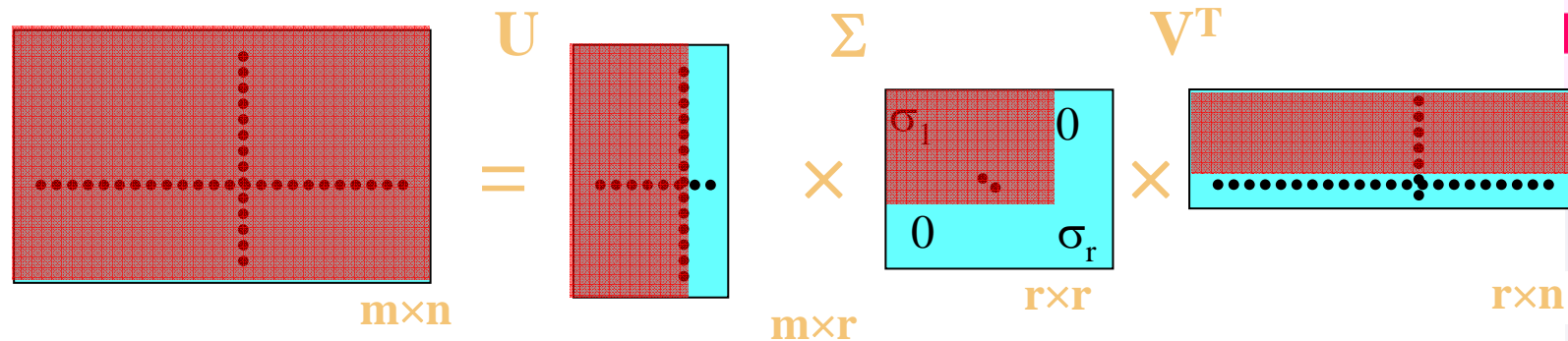
### Singulärwertzerlegung:

- Zerlegung der Dokument Term Matrix  $D = U \times \Sigma \times V^T$
- $m \times r$ -Matrix  $U$  mit orthonormalen Spaltenvektoren, entspricht den Eigenvektoren von  $DD^T$
- $r \times r$ -Diagonalmatrix  $\Sigma$  mit den Singulärwerten von  $D$
- $n \times r$ -Matrix  $V$  mit orthonormalen Spaltenvektoren, entspricht den Eigenvektoren von  $D^T D$
- $r$  ist der Rang der Matrix  $D$



<http://www.know-center.at>

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten Mathematisch



1. Reduktion der  $r$  Eigenwerte auf  $k$
2. Reduktion der Matrizen  $U$  &  $V$  auf  $m \times k$   
bzw.  $k \times n$
3. Berechnung der neuen Dokument Term  
Matrix

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## LSI Beispiel: Dokumente

---

- 
- $d_1$ : Indian government goes for open-source software
  - $d_2$ : Debian 3.0 Woody released
  - $d_3$ : Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
  - $d_4$ : gnuPOD released: iPod on Linux... with GPLed software
  - $d_5$ : Gentoo servers running an open-source mySQL database
  - $d_6$ : Dolly the sheep not totally identical clone
  - $d_7$ : DNA news: introduced low-cost human genome DNA chip
  - $d_8$ : Malaria-parasite genome database on the Web
  - $d_9$ : UK sets up genome bank to protect rare sheep breeds
  - $d_{10}$ : Dolly's DNA Damaged
- 

Aus „Modelling the Internet and the Web – Probabilistic Methods and Algorithms“, P. Baldi, P. Frasconi, P. Smyth, Wiley, 2003“

<http://www.know-center.at>

# Statistische Analysen zwischen Dokumente LSI Beispiel: Term Doku

	$d_1$	$d_2$	$d_3$							
open-source	1	0	0							
software	1	0	0							
Linux	1	0	0							
released	0	1	1							
Debian	0	1	1	0	0	0	0	0	0	0
Gentoo	0	0	1	0	1	0	0	0	0	0
database	0	0	0	0	1	0	0	1	0	0
Dolly	0	0	0	0	0	1	0	0	0	1
sheep	0	0	0	0	0	1	0	0	1	0
genome	0	0	0	0	0	0	1	1	1	0
DNA	0	0	0	0	0	0	2	0	0	1

- $d_1$ : Indian government goes for open-source software
- $d_2$ : Debian 3.0 Woody released
- $d_3$ : Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
- $d_4$ : gnuPOD released: iPod on Linux... with GPLed software
- $d_5$ : Gentoo servers running an open-source mysql database
- $d_6$ : Dolly the sheep not totally identical clone
- $d_7$ : DNA news: introduced low-cost human genome DNA chip
- $d_8$ : Malaria-parasite genome database on the Web
- $d_9$ : UK sets up genome bank to protect rare sheep breeds
- $d_{10}$ : Dolly's DNA Damaged

Aus „Modelling the Internet and the Web – Probabilistic Methods and Algorithms“, P. Baldi, P. Frasconi, P. Smyth, Wiley, 2003“

# Statistische Analysen und zwischen Dokumenten LSI Transformierte Matrix

- $d_1$ : Indian government goes for open-source software
- $d_2$ : Debian 3.0 Woody released
- $d_3$ : Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
- $d_4$ : gnuPOD released: iPod on Linux... with GPLed software
- $d_5$ : Gentoo servers running an open-source mysql database
- $d_6$ : Dolly the sheep not totally identical clone
- $d_7$ : DNA news: introduced low-cost human genome DNA chip
- $d_8$ : Malaria-parasite genome database on the Web
- $d_9$ : UK sets up genome bank to protect rare sheep breeds
- $d_{10}$ : Dolly's DNA Damaged

	$d_1$	$d_2$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_9$	$d_{10}$
open-source software	0.34	0.28	0.38	0.42	0.24	0.00	0.04	0.07	0.02	0.01
Linux	0.44	0.37	0.50	0.55	0.31	-0.01	-0.03	0.06	0.00	-0.02
released	0.63	0.53	0.72	0.79	0.45	-0.01	-0.05	0.09	-0.00	-0.04
Debian	0.39	0.33	0.44	0.48	0.28	-0.01	-0.03	0.06	0.00	-0.02
Gentoo	0.36	0.30	0.41	0.45	0.26	0.00	0.03	0.07	0.02	0.01
database	0.17	0.14	0.19	0.21	0.14	0.04	0.25	0.11	0.09	0.12
Dolly	-0.01	-0.01	-0.01	-0.02	0.03	0.08	0.45	0.13	0.14	0.21
sheep	-0.00	-0.00	-0.00	-0.01	0.03	0.06	0.34	0.10	0.11	0.16
genome	0.02	0.01	0.02	0.01	0.10	0.19	1.11	0.34	0.36	0.53
DNA	-0.03	-0.04	-0.04	-0.06	0.11	0.30	1.70	0.51	0.55	0.81

Aus „Modelling the Internet and the Web – Probabilistic Methods and Algorithms“, P. Baldi, P. Frasconi, P. Smyth, Wiley, 2003“

# Statistische Analysen und Ähnlichkeit zwischen Dokumenten

## LSI

---

### Vorteile

- Höhere Genauigkeit
- Auflösen von polysemen und homonymen Begriffen

### Nachteile

- SVD ist Rechenintensiv in der Berechnung
- Aktualisierung mit neuen Dokumenten komplex
- Keine Sparse Repräsentation mehr
- Einschränkung der Suchoperatoren (AND, OR, NEAR etc.)

# Inhalt

---

- Ein paar Zahlen zur Motivation
- Vorverarbeitung von Texten
- Statistische Analysen und Ähnlichkeit zwischen Dokumenten (VSM, LSI)
- **Maschinelle Lernmethoden im Überblick**
  - ◆ Textklassifikation (Rocchio, k-NN, SVM)
  - ◆ Clustering, Automatische Gruppierung von Texten (K-Means)
- Evaluierungskriterien

# Maschinelles Lernen

## Definitionen

---

**Definition:** The ability of a program to learn from experience — that is, to modify its output on the basis of newly acquired information (Nature).

- Induktiv: Vom Speziellen zum Allgemeinen (Beispielbasiert)
- Deduktiv: Vom Allgemeinen zum Speziellen (Logik)
- Der Fokus im ML liegt auf der Induktion

Relevante Disziplinen:

- Künstlichen Intelligenz (vorwiegend deduktive Ansätze)
- Wahrscheinlichkeitstheorie & Statistik
- Komplexitätstheorie & Informationstheorie
- Philosophie, Psychologie & Neurobiologie

# Maschinelles Lernen

## Wichtigsten Lernarten

---

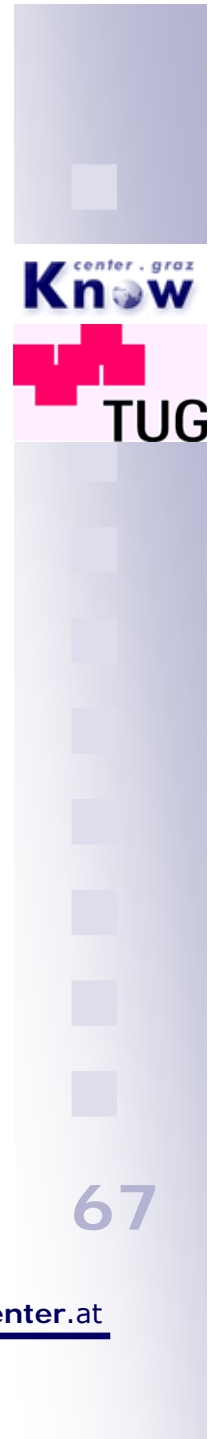
- **Supervised Learning (Klassifikation)**  
Lernen von vorgegebenen Zuordnungen
- **Unsupervised Learning (Clustering)**  
Zuordnung eines Modells zu Datenpunkten
- **Semi-Supervised Learning**  
Mischung aus Supervised & Unsupervised
- **Reinforcement Learning**  
Lernen von Aktionsmustern durch Belohnung

# Maschinelles Lernen

Auszug an populären Techniken

---

- Bayes Klassifikation
- Neuronale Netzwerke
- Lineare Klassifikatoren
- Entscheidungsbäume
- Genetische Algorithmen
- ....



# Maschinelles Lernen

## Definition: Supervised Learning

---

Gegeben: Menge von Datenpunkten ( $x$ ) mit gewünschten Zuordnungen ( $y$ )

$$X = \{ \langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_{n-1}, y_{n-1} \rangle, \langle x_n, y_n \rangle \}$$

Ziel: Lernen einer Funktion, welche Zuordnungen automatisch auf neuen, noch nicht gesehenen Datenpunkten trifft.

$$\Theta : X \times Y \rightarrow \{T, F\}$$

$$\Theta : X \times Y \rightarrow \mathbb{R}$$

Lösung (Brute-Force): Suche in allen möglichen Funktionen nach der „besten“ Lösung

Wichtig: Generalisierungsfähigkeit & Overfitting

# Maschinelles Lernen

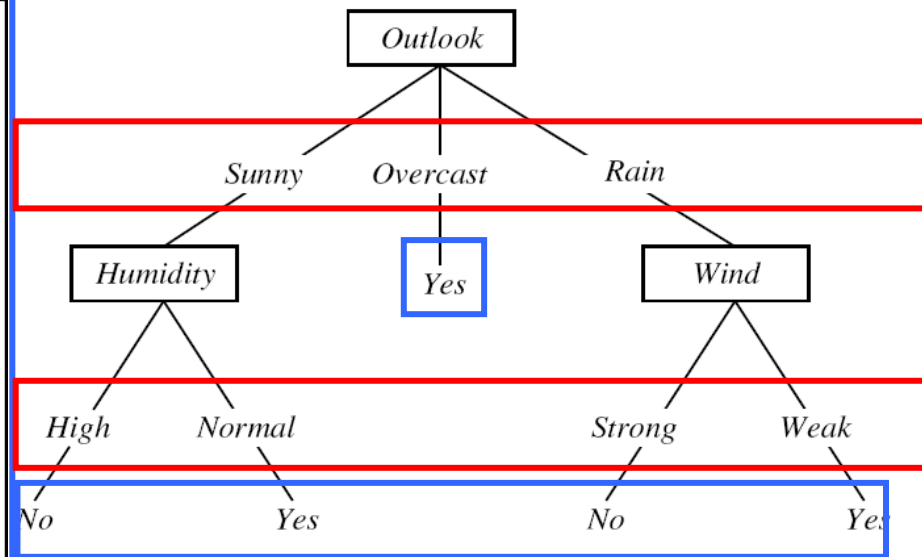
## Supervised Learning - Beispiel



Wann spielt man Tennis?

$x_i$						$y_i$
Day	Outlook	Temperature	Humidity	Wind	PlayTennis	
D1	Sunny	Hot	High	Weak	No	
D2	Sunny	Hot	High	Strong	No	
D3	Overcast	Hot	High	Weak	Yes	
D4	Rain	Mild	High	Weak	Yes	
D5	Rain	Cool	Normal	Weak	Yes	
D6	Rain	Cool	Normal	Strong	No	
D7	Overcast	Cool	Normal	Strong	Yes	
D8	Sunny	Mild	High	Weak	No	
D9	Sunny	Cool	Normal	Weak	Yes	
D10	Rain	Mild	Normal	Weak	Yes	
D11	Sunny	Mild	Normal	Strong	Yes	
D12	Overcast	Mild	High	Strong	Yes	
D13	Overcast	Hot	Normal	Weak	Yes	
D14	Rain	Mild	High	Strong	No	

Hypothesenklasse ist Entscheidungsbaum



# Maschinelles Lernen

## Anwendungsbereich Supervised Learning

---

- Textklassifikation
- Kontexterkenkung
- Ranking von Suchergebnissen
- Gen Daten Analyse
- Bildanalyse
- Spracherkennung
- Robotik
- Quantenphysik („Charming Quants“)

# Maschinelles Lernen

## Überblick hinsichtlich Datenart

---

### 🌐 Inputdaten (X)

- ◆ Nominale Wert: Temperature = {Hot, Mild, Cold}
- ◆ Numerische Werte: Temperature = 32°
- ◆ Vektoren  $X = \langle 0.4, 0.5 \dots 0.1 \rangle$

### 🌐 Klassifizierung y

- ◆ Binäre Klassifikation  $y = \{0, 1\}$
- ◆ Mehrere Klassen:  $y = \{\text{PlayTennis}, \text{PlayGolf}, \text{PlayJazz}\}$
- ◆ Regression:  $y \in \mathcal{R}$

# Maschinelles Lernen

## Textklassifikation

Supervised ML: Automatisches Zuordnen von Dokumenten zu Klassen basierend auf deren Merkmalen

- Input: hochdimensionale Vektoren

$$\vec{d}_1 = \langle w_{1,1}, w_{1,2} \dots w_{1,n-1}, w_{1,n} \rangle$$

⋮

$$\vec{d}_n = \langle w_{n,1}, w_{n,2} \dots w_{n,m-1}, w_{n,m} \rangle$$

- Im Allgemeinen mehrere Klassen (Multi-Class)
- Im Allgemeinen mehrere Zuordnungen eines Dokumentes zu Klassen (Multi-Label)

$$\langle \vec{d}_1, c_1 \rangle, \langle \vec{d}_1, c_2 \rangle \dots \langle \vec{d}_n, c_1 \rangle$$

# Textklassifikation

## Anwendungsmöglichkeiten

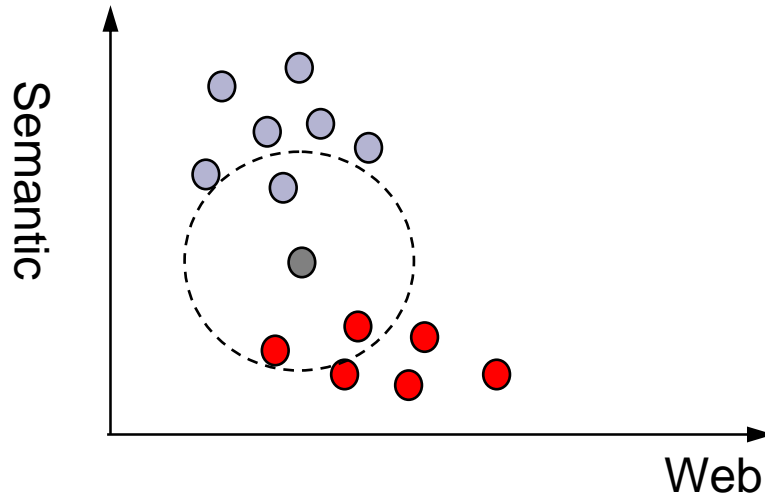
---

- Spam Filter
  - Positiven Beispiele: Erlaubte Mails
  - Negative Beispiele: Spam Mails
- Zuordnung von Dokumenten zu Klassifikationsschemata
  - z.B. IPTC, ACM, DMOZ, YAHOO
- Named Entity Extraction
- Part of Speech Tagging
- Helpdesk

# Textklassifikation

## k-Neares Neighbour Classifier

Dokument = Merkmalsvektor



Training: Lazy Learner, d.h. "kein" Training

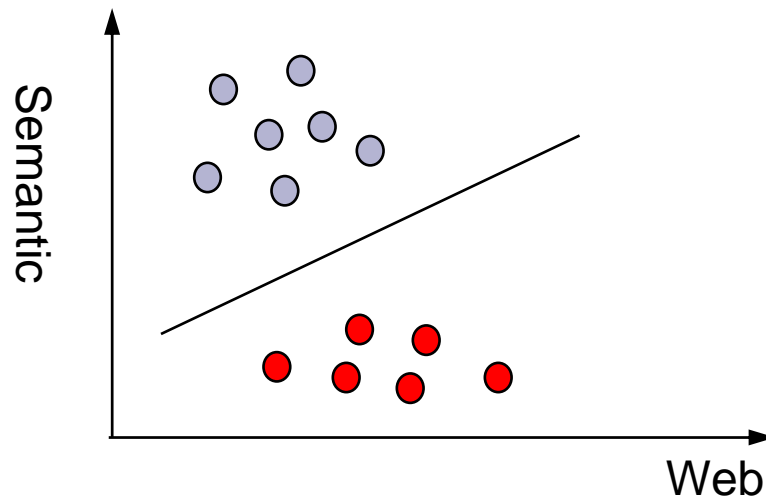
Klassifikation: Welche Nachbarn gehören zu welcher Klasse?

Majority Vote über die Anzahl der Klassen

# Textklassifikation

## Lineare Klassifikatoren

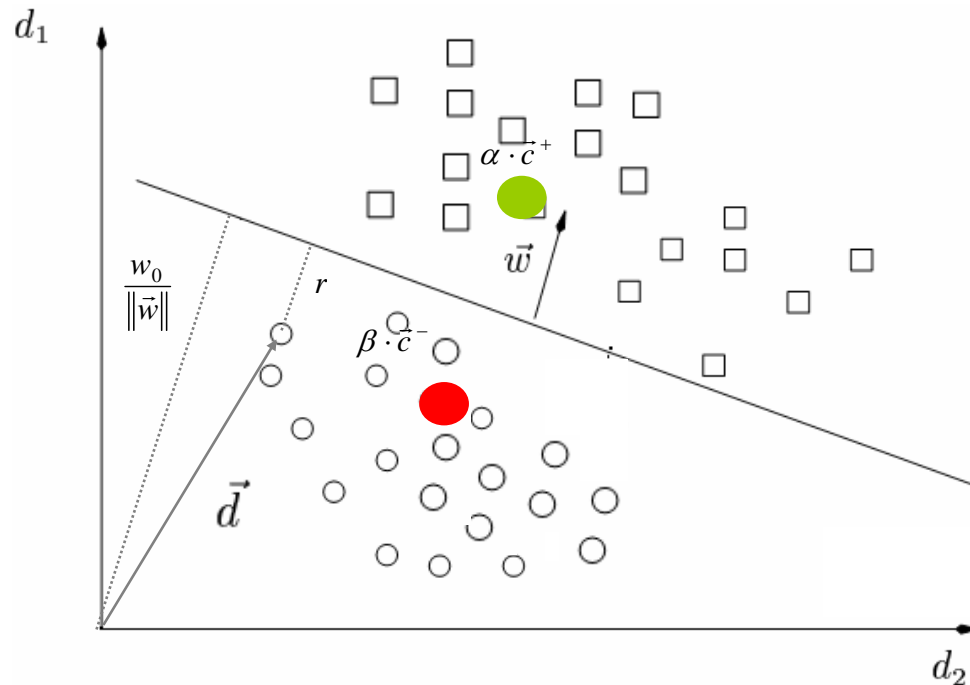
- Dokument = Merkmalsvektor



- Training: Finde trennende Ebene (Hyperebene)
- Algorithmen: Rocchio, Perceptron, Support Vector Machines
- XOR-Problem

# Textklassifikation

## Linearer Rocchio Klassifikator



$$h(\vec{d}) = \vec{w} \cdot \vec{d} + w_0$$

$$h(\vec{d}) = \begin{cases} > 0 & \text{positiv} \\ < 0 & \text{negativ} \\ = 0 & \text{on hyperplane} \end{cases}$$

$$r = \frac{h(\vec{d})}{\|\vec{w}\|}$$

$$y_i = \text{sgn}(h(\vec{d}))$$

$$\vec{w} = \alpha \cdot \frac{1}{|\text{positiv}|} \sum_{d_i \in \text{positiv}} \vec{d}_i - \beta \cdot \frac{1}{|\text{negativ}|} \sum_{d_i \in \text{negativ}} \vec{d}_i + w_0$$

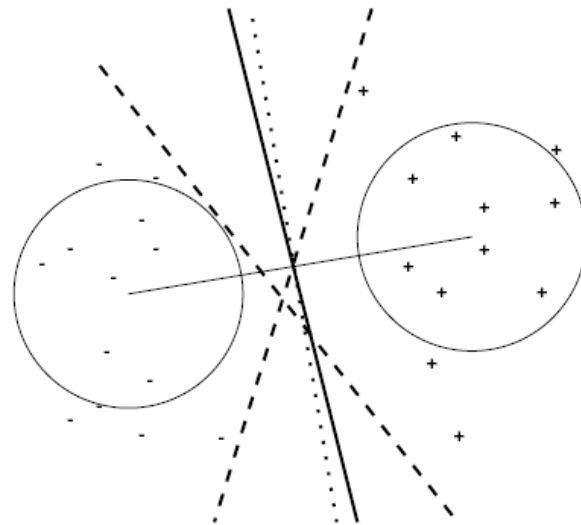
$$\vec{w} = \alpha \cdot \vec{c}^+ - \beta \cdot \vec{c}^- + w_0$$

# Textklassifikation

## Support Vector Maschinen

---

Problem: i.A. existieren eine Vielzahl von möglichen Trennenden (Hyper)Ebenen



Welche davon liefert die besten Vorhersagen?

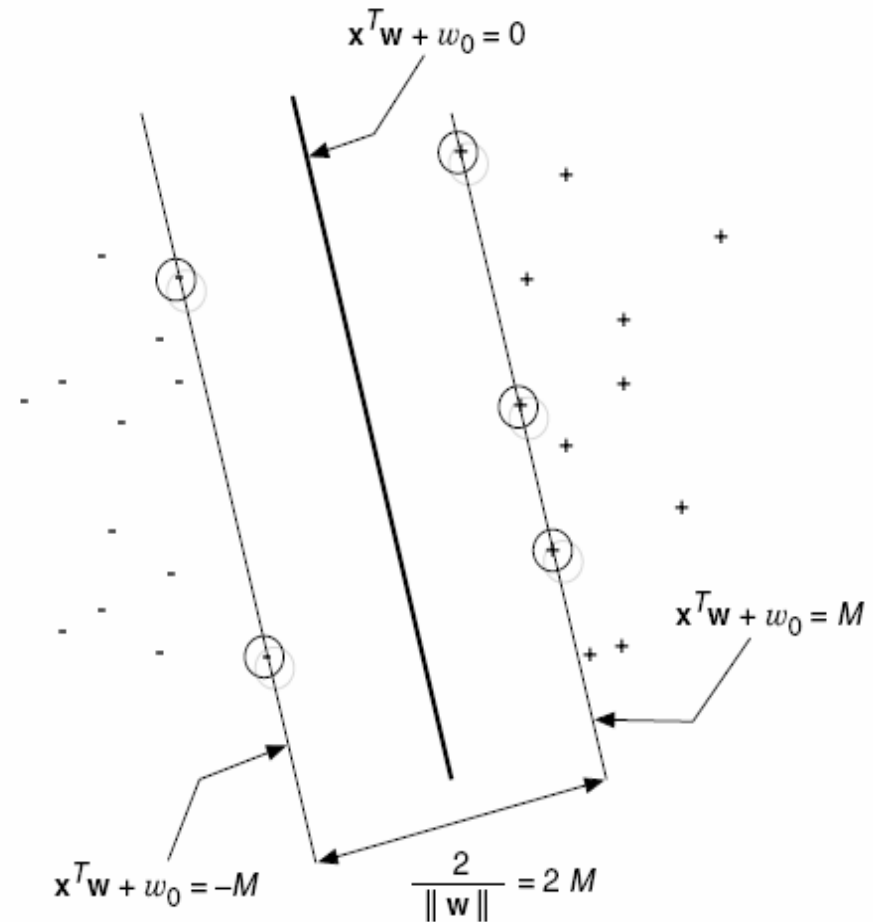
# Textklassifikation

## Support Vector Maschinen

Large Margin Classifier:  
(Hyper)Ebene, mit  
maximalen Abstand zu  
den Beispielen

Support Vektoren:

Vektoren mit minimalen  
Abstand zur Hyperebene



# Textklassifikation

## Support Vector Maschinen

Eigenschaft der Beispiele

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad \text{for } y_i = +1$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad \text{for } y_i = -1$$

→ n-Ungleichungen

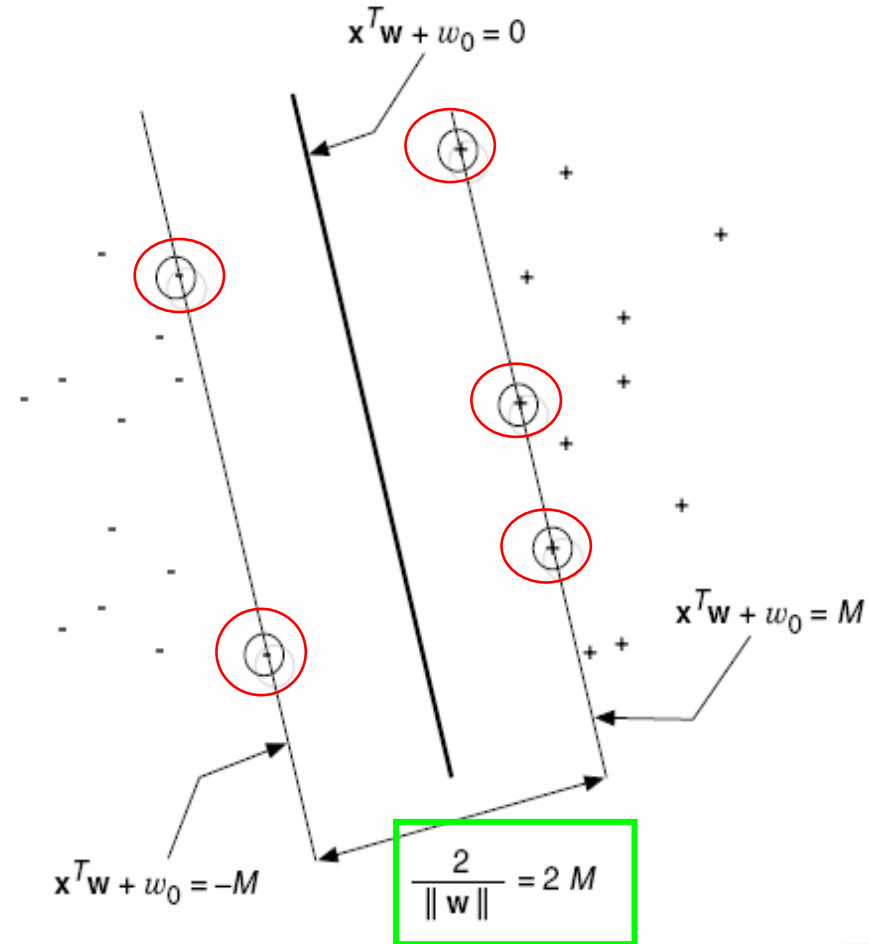
$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i$$

Minimieren nach  $w$  und  $b$  liefert  
maximale Margin

→ Lagranges Optimierungsproblem

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

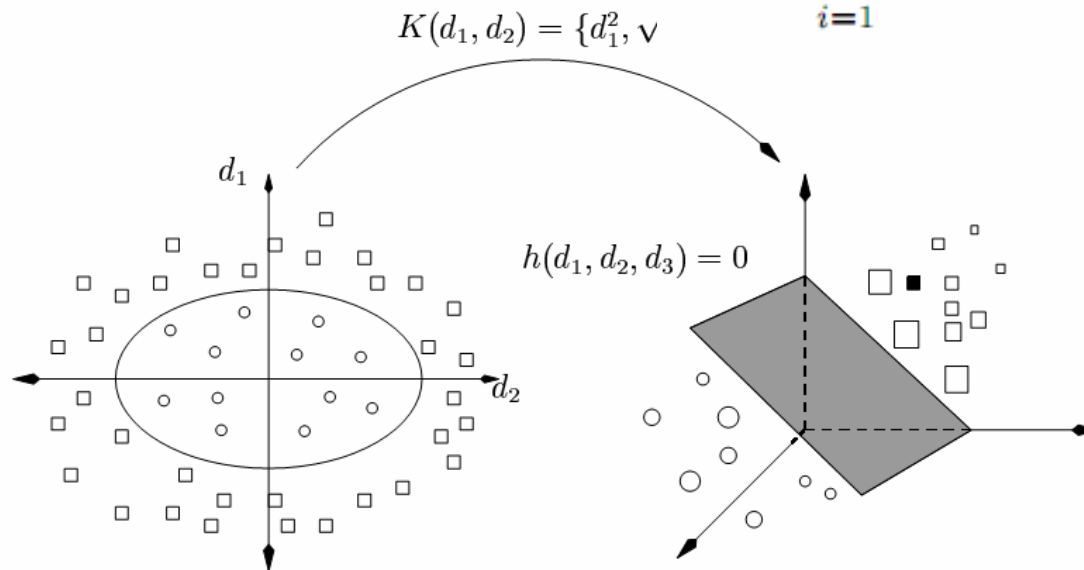


# Textklassifikation

## Kerneleigenschaft (simplifiziert)

- Supportvektoren sind Beispiele mit  $\alpha_i \neq 0$
- Klassifikationshypothese als inneres Produkt der Supportvektoren

- Kerneleigenschaft  $f(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i y_i \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b$



# Textklassifikation

## SVM's

---

### Vorteile

- Hohe Genauigkeit
- Hochdimensionale Daten (d.h. gut für Text)
- Trennende Hyperebene bestimmt durch Beispiele (Supportvektoren)
- Kerneigenschaft
- Fundierte Theorie: vorhersage der Generalisierungseigenschaften mit gewisser Wahrscheinlichkeit möglich

### Nachteile

- Hohe Trainingszeit (ohne Heuristiken)
- Gute Heuristiken existieren

# Textklassifikation

## Support Vector Maschinen

---

Einführung in SVM's:

<http://www.kernel-machines.org/>

<http://mlg.anu.edu.au/~raetsch/ps/review.pdf>



# Textklassifikation

## Genauigkeit

---

- Weitere Techniken
  - Neuronale Netzwerke
  - Bayes Klassifikatoren
  - Entscheidungsbäume
- Anzahl der Trainingsbeispiele entscheidend
- Inter Indexer Inkonsistenz
- Qualität abhängig von der Anzahl der Klassen, Domäne etc. (z.B. [Liu et. al 05]: Tests mit Yahoo: F1 Wert von 0.24 bei 800.000 Dokumenten und 300.000 Klassen)
- Ausnutzen von Strukturen von Informationsräumen (e.g. Taxonomien)
- Problem: Reihung von Klassifikationsergebnissen

# Maschinelles Lernen

## Unsupervised Learning

---

Gegeben: Menge von Datenpunkten ( $x$ ) ohne Zuordnung ( $y$ )

$$X = \{ \langle x_1 \rangle, \langle x_2 \rangle, \dots, \langle x_{n-1} \rangle, \langle x_n \rangle \}$$

Ziel: Approximation eines vorgegebenen Modells

- Clustering
- Reinforcement-Learning
- Dimensionalitätsreduktion
- Wahrscheinlichkeitsfunktion

# Clustering

## Definition

---

- Gegeben eine Menge an Datenpunkte

$$X = \{ \langle x_1 \rangle, \langle x_2 \rangle, \dots, \langle x_{n-1} \rangle, \langle x_n \rangle \}$$

- Finde jene Gruppen  $C$  von Datenpunkten, welche ein gegebenes Kriterium (z.B. Ähnlichkeitsfunktion) optimieren

$$C = \{ C_i \mid f_{\text{int ra}}(C_i) \rightarrow \max \wedge \forall_{j \neq i} f_{\text{int er}}(C_i, C_j) \rightarrow \min \}$$

$$C_i \subseteq X; X = \bigcup_i C_i$$

- Hartes Clustering vs. Fuzzy Clustering

# Clustering

## Definition

---

- Intra-Cluster Kriterium:
  - Maximiere die Ähnlichkeit aller Datenpunkte in einem Cluster
  - Minimiere die Distanz der Datenpunkte in einem Cluster
- Inter-Cluster Kriterium:
  - Minimiere die Ähnlichkeit der Datenpunkte aus unterschiedlichen Cluster
  - Maximiere die Distanz der Datenpunkte aus unterschiedlichen Cluster
- Formulierung auch in Form einer Funktion möglich

# Clustering

---

- Repräsentation eines Clusters
  - Centroid: Summe der Datenpunkte
  - Medoid: „Bester“ Datenpunkt im Cluster
- Problem: Lesbare Beschreibung eines Cluster
- Finden von abstrakten Begriffen zur Beschreibung eines Clusters ist problematisch (e.g. Fußball vs. "Rapid, Austria")

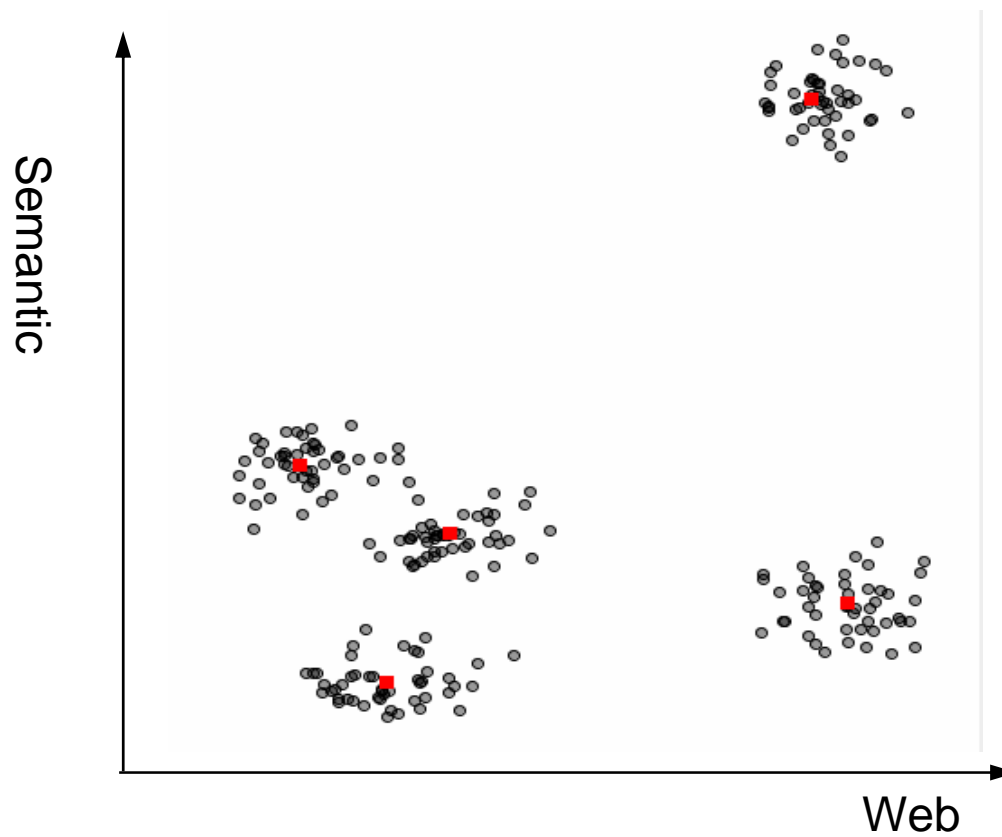
# Clustering

## Methoden

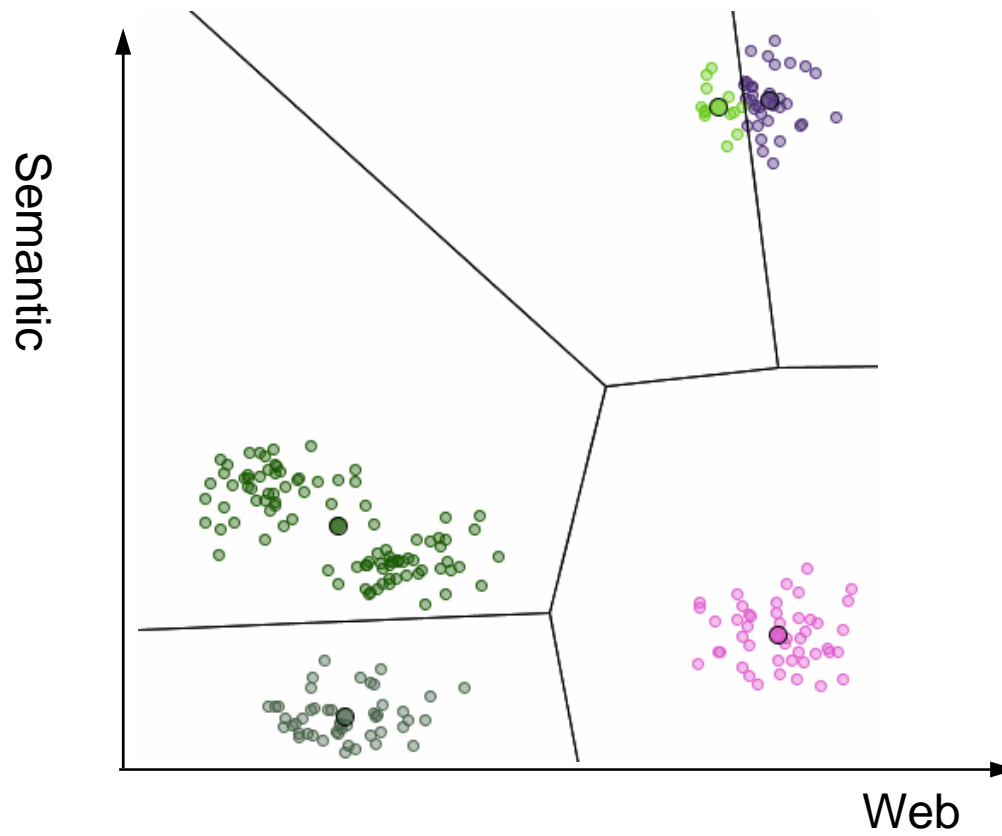
---

- Hierarchisches Clustering
  - Agglomerativ (Bottom-Up)
  - Divisive (Top-Down)
- Partitionierendes Clustering
  - K-Means/K-Medoid
  - Fuzzy K-Means
  - Probabilistische Methoden
  - Dichtebasierte Verfahren

# Clustering Beispiel



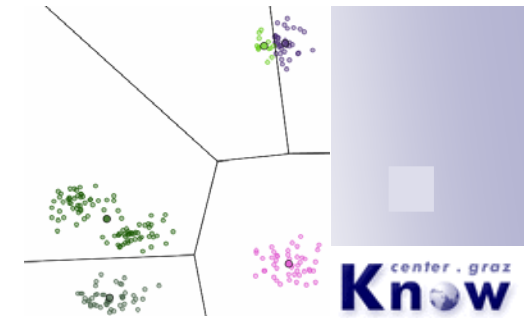
# Clustering Beispiel



<http://www.know-center.at>

# Clustering

## Beispiel K-Means



- Gegeben: Dokumentvektoren, Anzahl der Cluster  $c$

- Sketch d. Algorithmus

1. Wähle zufällig  $c$  Dokumente und setze diese als Clustercentroid

$$\vec{c}_k = \vec{d}_j$$

2. Berechne die Ähnlichkeit aller Dokumente zu den Clustercentroiden

$$sim_{k,j} = c_k \cdot d_j$$

3. Addiere die Dokumente zu dem Centroiden mit maximaler Ähnlichkeit

$$c_k = \frac{1}{N} \sum_{d_j \in D_{\max,k}} d_j$$

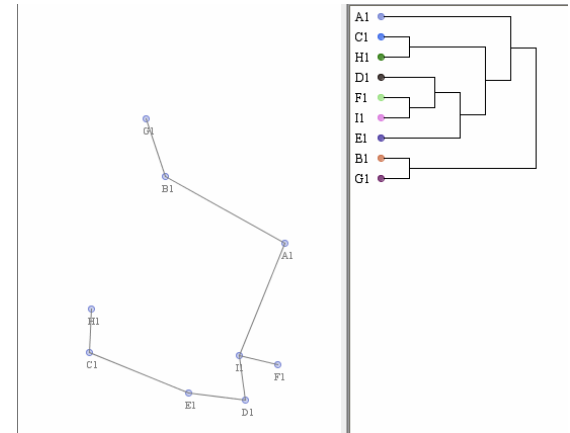
4. Gehe zu 2 solange bis sich die Centroiden nicht mehr ändern

- Continuous K-Means vs. Batch K-Mean

# Clustering

## Beispiel Hierarchical Agglomerative Clustering

- Bottom up
- Gegeben Dokumentvektoren, Anzahl Cluster  $c$ 
  1. Suche Dokument mit maximaler Ähnlichkeit zu allen anderen Dokumenten und allen bekannten Clustern
  2. Vereinige das gefundene Dokument mit dem ähnlichsten Cluster/Dokument
  3. Gehe zu 1
- Verschiedene Linkage Strategien
  - ◆ Single Link: minimale Distanz zwischen Cluster
  - ◆ Complete Link: maximale Distanz zwischen Cluster
  - ◆ Average Link: Durchschnittliche Distanz zwischen Cluster



# Clustering

---

- Anwendung:
  - Automatische Extraktion von Themen einer Suche
  - Finden von Plagiaten/Mutationen von Texten
  - Extraktion von Konzepten in einem Informationsraum
  - Zusammenfassung von Texten

# Clustering

## Anwendungsbeispiel

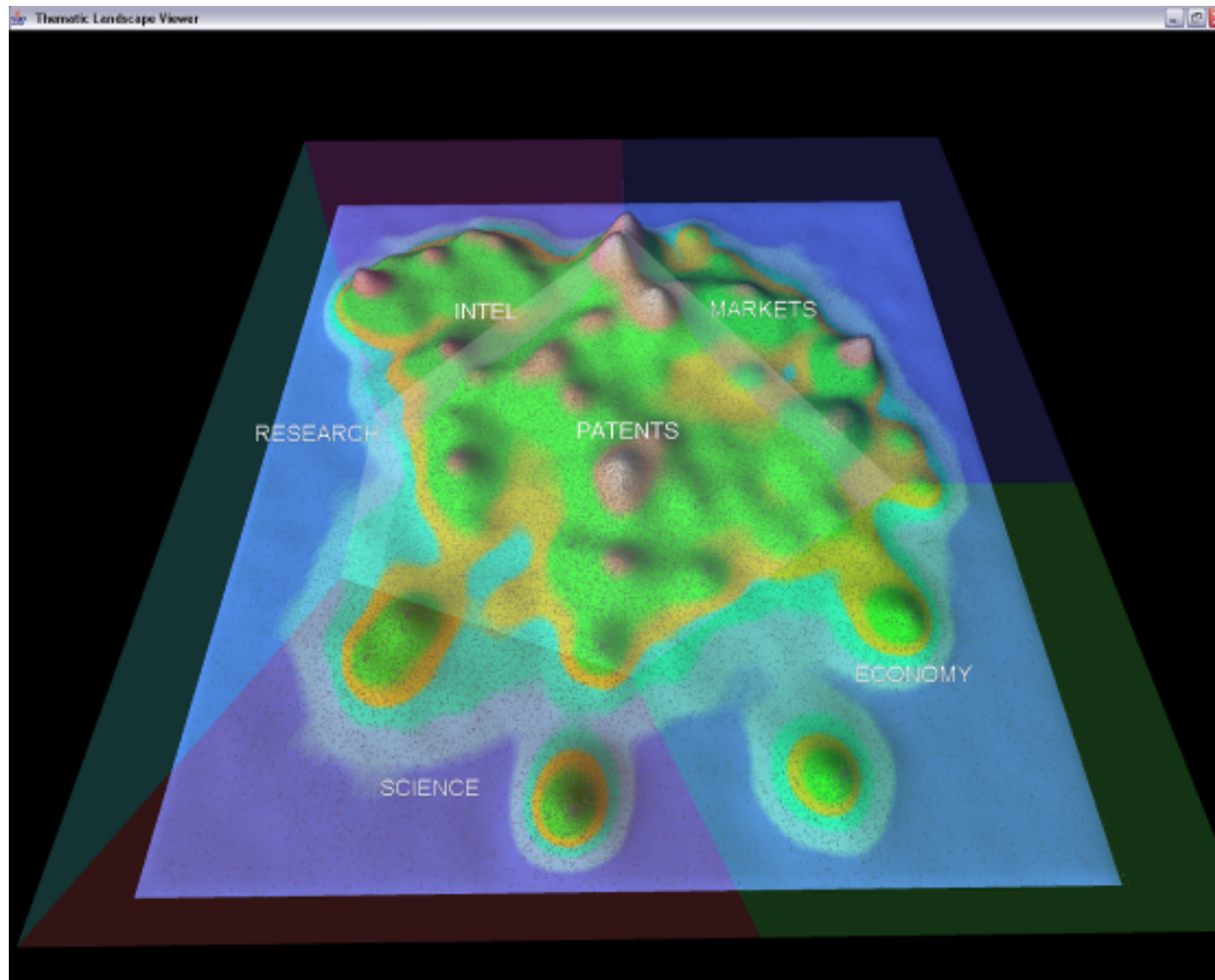
The screenshot shows the Clusty search engine interface. At the top, there's a navigation bar with categories like 'Web+', 'News', 'Images', 'Shopping', 'Wikipedia', 'Blogs', 'Jobs', and 'Customize!'. A search bar contains the text 'semantic web' and a 'Cluster' button. Below the search bar, there's a 'Cluster by:' dropdown menu set to 'Topics'. The main content area displays 'Top 205 results of at least 1,054,037 retrieved for the query semantic web (Details)'. On the left, a sidebar lists various clusters: 'All Results (205)', 'W3C, Activity (21)', 'Ontology (29)', 'Web Services (23)', 'Berners-Lee (17)', 'Semantic Web Conference (14)', 'World Wide Web (13)', 'Semantic Web Technologies (13)', 'Science (8)', 'Developers (10)', and 'Resource (11)'. The main results list includes:

- Semantic Web** (Sponsored Result): Development Tools, SDKs & Resource Docs to Create Content/Applications. [www.Forum.Nokia.com](http://www.Forum.Nokia.com)
- New Semantic Web Tool** (Sponsored Result): Visually design RDF, RDFS and OWL docs & ontologies. Free Trial. [www.Altova.com/SemanticWorks](http://www.Altova.com/SemanticWorks)
- 1. W3C Semantic Web** (Search Result): **Semantic Web The Semantic Web** provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is a ... [www.w3.org/2001/sw](http://www.w3.org/2001/sw) - [cache] - GigaBlast, MSN, Wisenut, MSN Search, Ask Jeeves, Looksmart
- 2. SemanticWeb.org - The Semantic Web community portal** (Search Result): semanticweb.org is going to be relaunched old site preview site project documentation site [www.semanticweb.org](http://www.semanticweb.org) - [cache] - Wisenut, Looksmart, MSN, Ask Jeeves, Open Directory, MSN Search, GigaBlast
- 3. The Semantic Web: An Introduction** (Search Result): The **Semantic Web: An Introduction** This document is designed as being a simple but comprehensive introductory publication for anybody trying to get into the **Semantic Web**: from beginners through to ... [www.infomesh.net/2001/swintro](http://www.infomesh.net/2001/swintro) - [cache] - Ask Jeeves, MSN, GigaBlast, Looksmart, MSN Search, Wisenut, Open Directory
- 4. Shirky: The Semantic Web, Syllogism, and Worldview** (Search Result)

At the bottom left of the Clusty interface, there is a contact form: 'Tell us what you think. clusty@clusty.com'.

# Clustering

## Automatisches Gruppieren von Patenten



# Clustering

## Herausforderungen

---

- Ähnlichkeitsmaß ist essentiell
  - Gruppierung nach Datum
  - Gruppierung nach Personen
  - Gruppierung nach Inhalt
- Laufzeit vs. Qualität
- Wie viele Gruppen?
- Clustering von 4000 Dokumenten in "Echtzeit" möglich
- Clustering des WWW's:
  - Nur Approximativ
  - 30 Millionen Dokumente ~ 2 Tage

# Evaluierungsmethoden

---

- Was bedeutet Genauigkeit?
- Wie ist diese Messbar
- Unterschied Supervised vs. Unsupervised
  - Supervised: Messung, wie gut die Zuordnung gelernt wurde
  - Unsupervised: Durchschnittliche Inter- bzw. Intra Cluster Similarity

# Evaluierungsmethoden

---

- Was bedeutet Genauigkeit?
- Wie ist diese Messbar
- Unterschied Supervised vs. Unsupervised
  - 🌐 Supervised: Messung, wie gut die Zuordnung gelernt wurde
  - 🌐 Unsupervised:
    - ◆ Durchschnittliche Inter- bzw. Intra Cluster Similarity
    - ◆ Vergleich mit vorgegebener Klassifikation

# Evaluierungsmethoden

## Supervised

- Kontingenztabelle
- Für jede Klasse:

Klassenzugehörigkeit  
(Ground Truth)

Klassifikations- entscheidung	Klasse $C_i$	True	False
	Positive	True Positives	False Positives
	Negative	False Negatives	True Negatives

# Evaluierungsmethoden

## Supervised

### Accuracy/Error Rate

$$\hat{A} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\hat{E} = \frac{FP+FN}{TP+TN+FP+FN} = 1 - \hat{A}$$

### Precision (Genauigkeit)

$$prec_i = \frac{TP_i}{TP_i + FP_i}$$

### Recall (Vollständigkeit)

$$rec_i = \frac{TP_i}{TP_i + FN_i}$$

### F<sub>β</sub>-Measure

$$F_\beta = \frac{(\beta^2 + 1) * prec * rec}{\beta^2 * prec + rec}$$

Klasse C <sub>i</sub>	True	False
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

# Evaluierungsmethoden

## Supervised

### Macro-Averaging vs. Micro-Averaging

$$prec_i^M = \frac{\sum_{i=1}^{|C|} prec_i}{C}$$

$$rec_i^M = \frac{\sum_{i=1}^{|C|} rec_i}{C}$$

$$prec_i^\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)}$$

$$rec_i^\mu = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

### Erstellen von Testsamples

- ◆ Split in Training- & Testdaten
- ◆ Random Sampling
- ◆ Cross-Validation

# Inhalt

---

- ◆ Ein paar Zahlen zur Motivation
- ◆ Vorverarbeitung von Texten
- ◆ Statistische Analysen und Ähnlichkeit zwischen Dokumenten
- ◆ Clustering, Automatische Gruppierung von Texten
- ◆ Textklassifikation
- ◆ Ontology Learning from Text

# Ontology Learning from Text

## Motivation und Fragestellung

---

- Reduzierung des Aufwandes bei
  - Erzeugung von Ontologien
  - Erweitern bzw. aktualisieren von Ontologien
  - Zuordnen von Instanzen
- Multidisziplinäres Feld: NLP, Data und Web Mining, maschinelles Lernen, Wissensrepräsentation

# Ontology Learning from Text

## Motivation und Fragestellung

---

- Wie können die hier beschriebenen Techniken angewendet werden?
  - Identifikation von Konzepten
  - Identifikation von Relationen zwischen Konzepten
  - Zuordnung von Instanzen zu Konzepten
- Startpunkt:
  - relevante Menge an Dokumenten/Texten
  - Lexikalische Ressourcen (e.g. Dictionaries, Glossar)
  - Ontologien (e.g. WordNet)

# Ontology Learning from Text

## Extraktion von Konzepten

---

- Vorverarbeitung
  - POS Tagging essentiell (z.B. Identifikation von Noun Phrases)
  - Struktur/Syntax wird mit berücksichtigt
- Extraktion von Konzepten
  - Statistische Analysen
    - Finden von Schlüsselbegriffen
    - Finden von Schlüsselphrasen (n-Word Grams)
  - Clustering
    - Finden von Konzepten in Datenbeständen
    - Definition des Konetxtes
      - Syntaktisch
      - Co-Occurrence
- Information Extraction
  - Identifikation von Personen, Orten etc.
  - Zusätzliche Analyse der Struktur (e.g. Autor: [Person])

# Ontology Learning from Text

## Zuordnen von Instanzen

---

- Klassifikation
  - Zuordnen von Dokumenten
  - Zuordnen von Sätzen
  - Zuordnen von Wortgruppen
- Information Extraction
  - Identifikation von Personen, Orten etc.
  - Zusätzliche Analyse der Struktur (e.g. Autor: [Person])

# Ontology Learning from Text

## Extraktion von Relationen

---

### Symbolisch:

- Regulärer Ausdruck (Lexico Syntactic Patterns aka. Hearst Patterns):
  - ◆ Musikinstrumente, wie die Gitarre, das Schlagzeug...
  - ◆ [Substantiv], wie [Art] [Substantiv], [Art] [Substantiv] \*
- Lernen von Ausdrücken mittel Klassifikation
- Automatisches finden von Gruppen solcher Regeln
- Part-of-Speech Tagging
- Ein richtiges Beispiel genügt

### Statistisch:

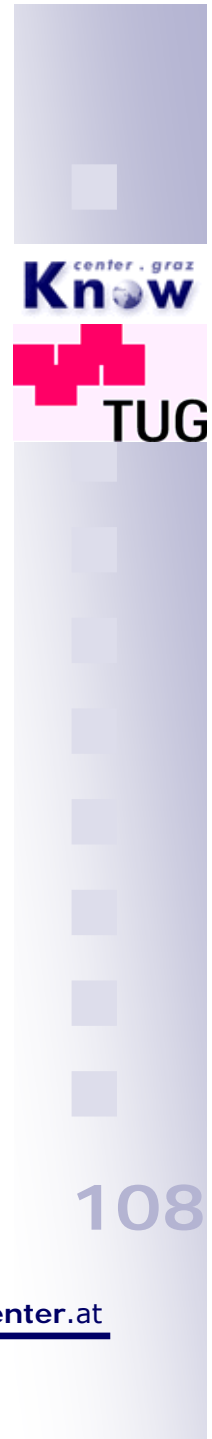
- Co-Occurrences
- Hierarchical Clustering: hierarchische Beziehung zu Sub-Cluster
- Extraktion aus großen Datenmengen
- Klassifikation von Instanzen

# Ontology Learning from Text

Beispiel: Text2Onto, AIFB Karlsruhe

---

- Ontology Learning Framework vom Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Universität Karlsruhe (TH)
- <http://ontoware.org/projects/text2onto/>



# Literatur zum Thema

---

Chris Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press.

<http://nlp.stanford.edu/fsnlp/>

D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria, 2001.

<http://gate.ac.uk/sale/ranlp2001/maynard-etal.pdf>.

Gate User Guide: <http://gate.ac.uk/sale/tao/index.html>

Text Mining: Predictive Methods for Analyzing Unstructured Information (Hardcover), Sholom Weiss, Nitin Indurkha, Tong Zhang, Fred Damerau

<http://www.know-center.at>

# Zusammenfassung

---

- Linguistische Analysen zur Merkmalsgenerierung aus Text
- Überführung in Vektorform zur Berechnung
  - Gewichtung des Vektorraums
  - Selektion von Merkmalen
  - Transformation des Vektorraums
- Supervised Machine Learning
  - Lineare Klassifikatoren
  - Support Vektor Maschinen
- Unsupervised Machine Learning
  - K-Means
  - HAC
- Evaluierung: Precision & Recall

---

Danke für die Aufmerksamkeit

Michael Granitzer

mgrani@know-center.at

<http://www.know-center.tugraz.at/forschung/wissenserschliessung>

111

<http://www.know-center.at>

# Zusammenfassung

---

Zum Nachlesen: "Modelling the Internet and the Web – Probabilistic Methods and Algorithms", P. Baldi, P. Frasconi, P. Smyth, Wiley, 2003

Kapitel 4: Text Analysis, verfügbar unter:

[http://media.wiley.com/product\\_data/excerpt/61/04708490/0470849061.pdf](http://media.wiley.com/product_data/excerpt/61/04708490/0470849061.pdf)

# Literatur zum Thema

---

- C. van Rijsbergen. Information Retrieval, 1979
- D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, and D. Maynard.  
Experiments with geographic knowledge for information extraction. In Workshop on Analysis of Geographic References, HLT/NAACL'03, Edmonton, Canada, 2003.  
<http://gate.ac.uk/sale/hlt03/paper03.pdf>
- Mladenic, D., "Text-learning and related intelligent agents: a survey,"  
*Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]* , vol.14, no.4pp.44-54, Jul/Aug1999
- Text Categorization (2005) Fabrizio Sebastiani  
<http://citeseer.ist.psu.edu/sebastiani05text.html>
- Xu, R. & Wunsch, D. (2005), 'Survey of clustering algorithms', Neural Networks, IEEE Transactions on 16(3), 645--678.

# Literatur zum Thema

---

- [Hearst 1999] Hearst, M.A. (1999), Untangling text data mining, in 'Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics', Association for Computational Linguistics, Morristown, NJ, USA, pp. 3--10.
- [Lyman 2003] Lyman, Varian, How Much Information 2003  
<http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/>
- [Breivik 1998] Patricia Senn Breivik, Student Learning in the Information Age (1998)
- [Delphi 2002] Delphi Group, Taxonomy & Content Classification Market Milestone Report, Delphi Group White Paper, 2002. See <http://delphigroup.com>.
- [Berners-Lee 2001]
- [Boehm 86] Boehm, B. (1986), 'A spiral model of software development and enhancement', SIGSOFT Softw. Eng. Notes 11, 14--24
- [Wurman 1989] Richard Saul Wurman, Information Anxiety (1989)