Text Mining

Michael Granitzer
mgrani@know-center.at

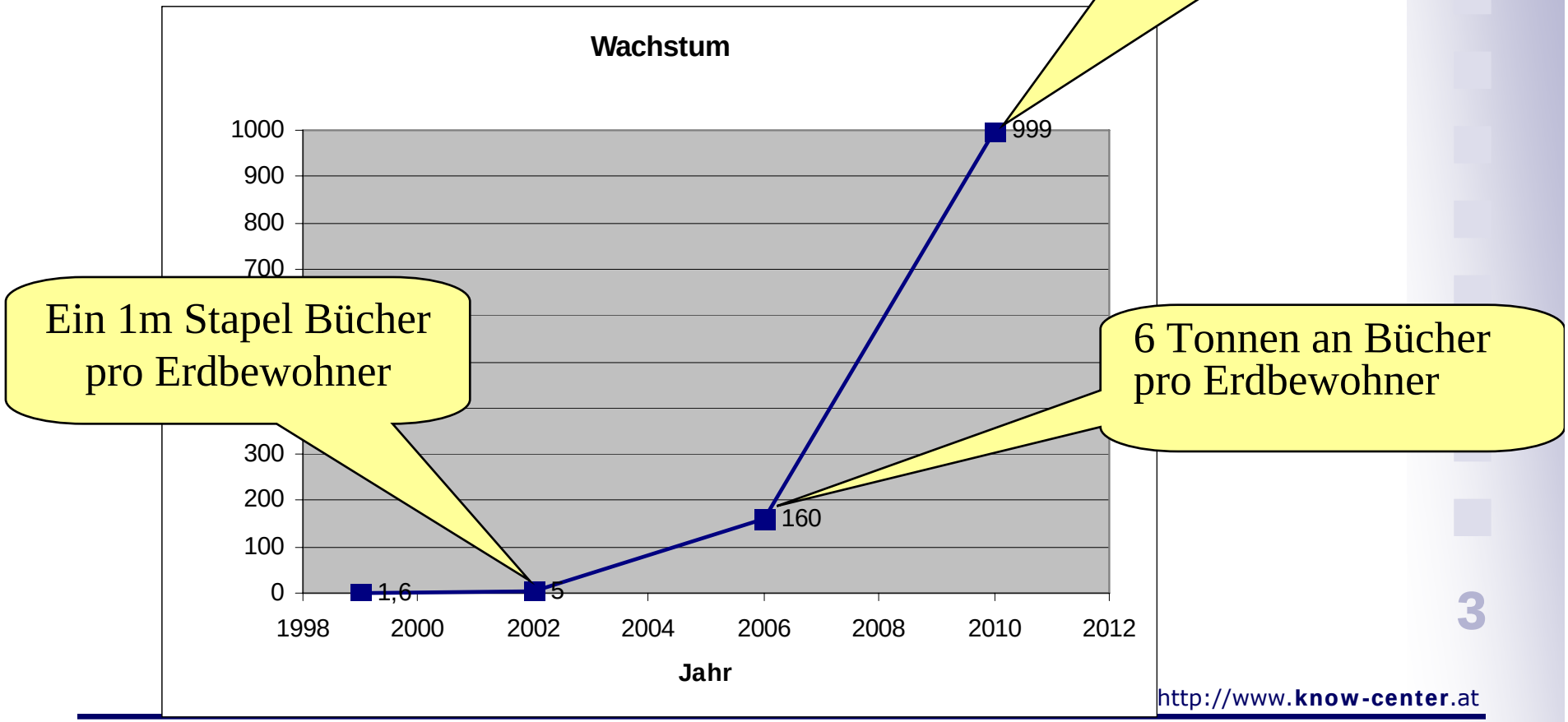
http://www.**know-center**.at/swat

Inhalt

- **Ein paar Zahlen zur Motivation**
- Vorverarbeitung von Texten
- Vektorraummodell
- Maschinelle Lernmethoden im Überblick
  - Supervised
  - Unsupervised

Ausgangssituation
Zahlen und Fakten I

- Wie viel Information umgibt uns?



Technischer Lösungsansatz
Knowledge Discovery & Text Mining

- Derzeit Text als Hauptinformationsträger
- Hoher Grad an redundanter Information
- „Noise" überdeckt relevante Information

➔(Automatische) Unterstützung beim Umgang mit textueller Information

➔ Extraktion semantischer Beziehungen aus Texten

Knowledge Discovery

Knowledge Discovery and Data Mining: Towards a Unifying Framework (1996)
Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth
Knowledge Discovery and Data Mining

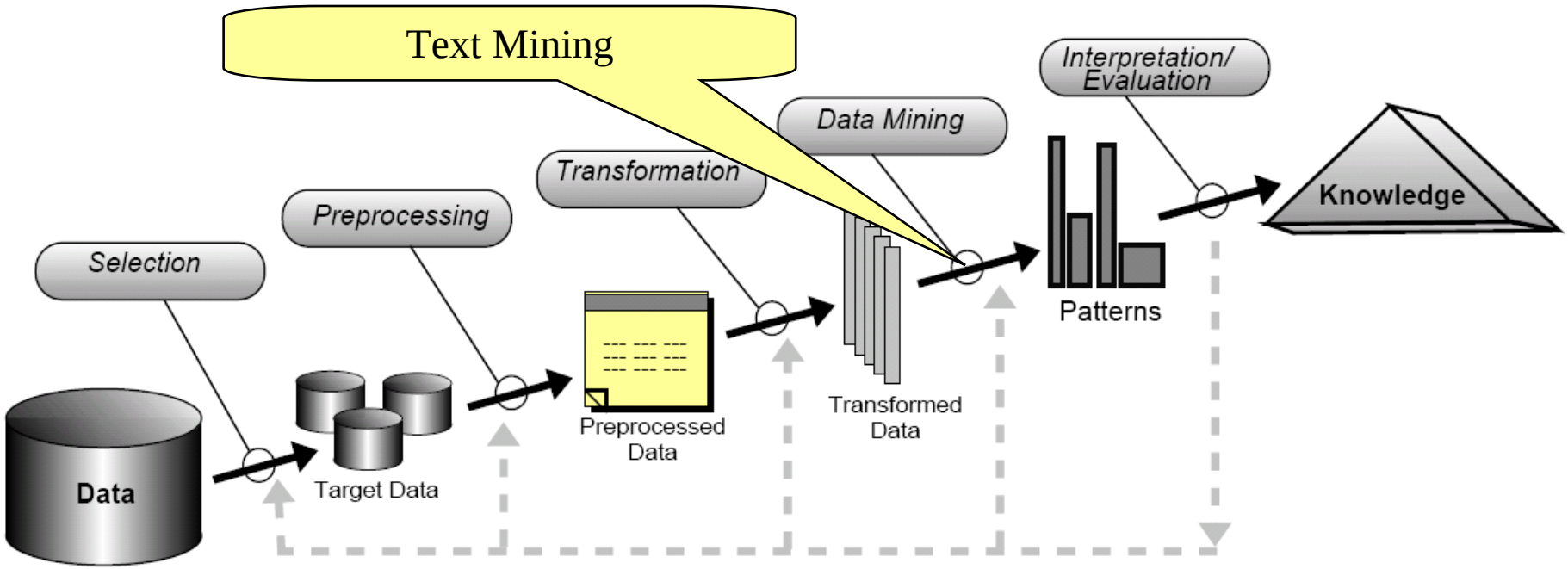


Figure 1: An overview of the steps comprising the KDD process.

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)*

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**
- Vektorraummodell
- Maschinelle Lernmethoden im Überblick
  - Supervised
  - Unsupervised

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

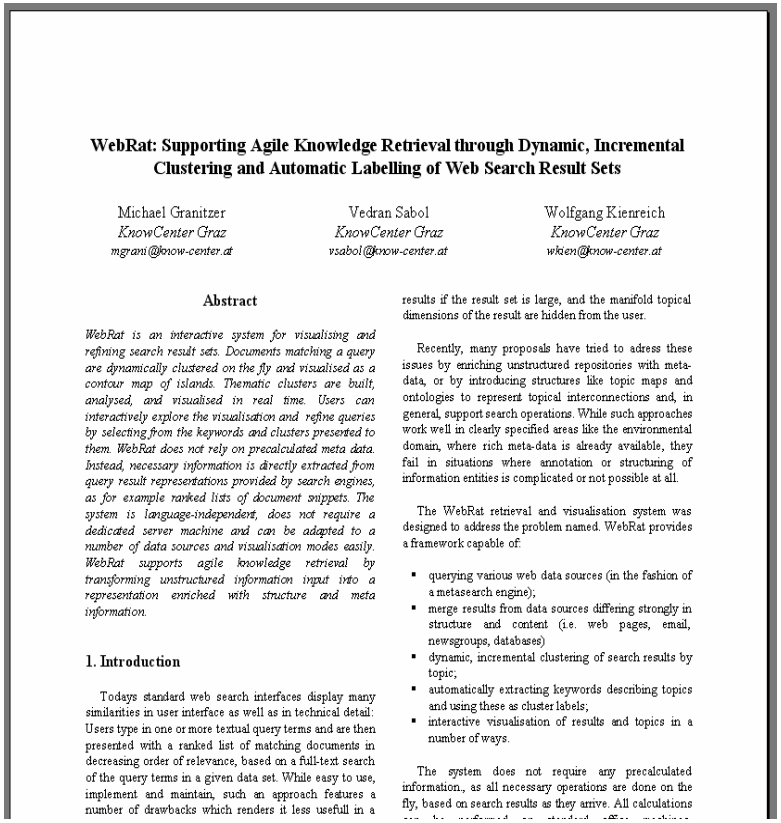
Inhalt

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 visualising 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

Todays 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 usefull 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 adress 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

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
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?

Vorverarbeitung von Text

Lexikalische Analyse - Tokenization

Ziel: Zerlegen eines Textes in atomare, sinnvolle Einheiten welche weiter verarbeiteten 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.)
  - Wortform = Nomen
  - Der = Artikel
  - …
- Anzahl der unterschiedlichen Wortformen definiert durch sogn. Tag Set
- Sprachabhängig

Vorverarbeitung von Text

Ein kurzes Beispiel

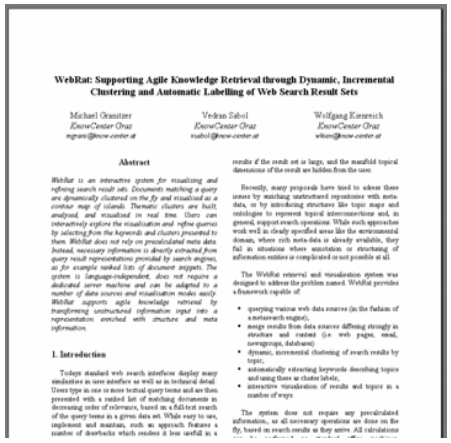
„Ein kurzes Beispielchen.“

- 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;PUNCTUATION]}
- 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



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 matching unstructured repositories with meta data, or by introducing structure 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 visualization system was designed to address the problem named. WebRat provides a framework capable of

- querying various web data sources (in the fashion of a meta-search 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 visualization 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

Vorname	Nachname	Zugehörigk.	E-Mail
Michael	Granitzer	Know-Center	mgrani@know-center.at
Wolfgang	Kienreich	Know-Center	wkien@know-center.at
Vedran	Sabol	Know-Center	vsabol@know-center.at

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 Gazeteers, Thesauri und Ontologien

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

Vorverarbeitung von Texten
Informationsextraktion

Beispiel f. Regeln:

- 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

„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

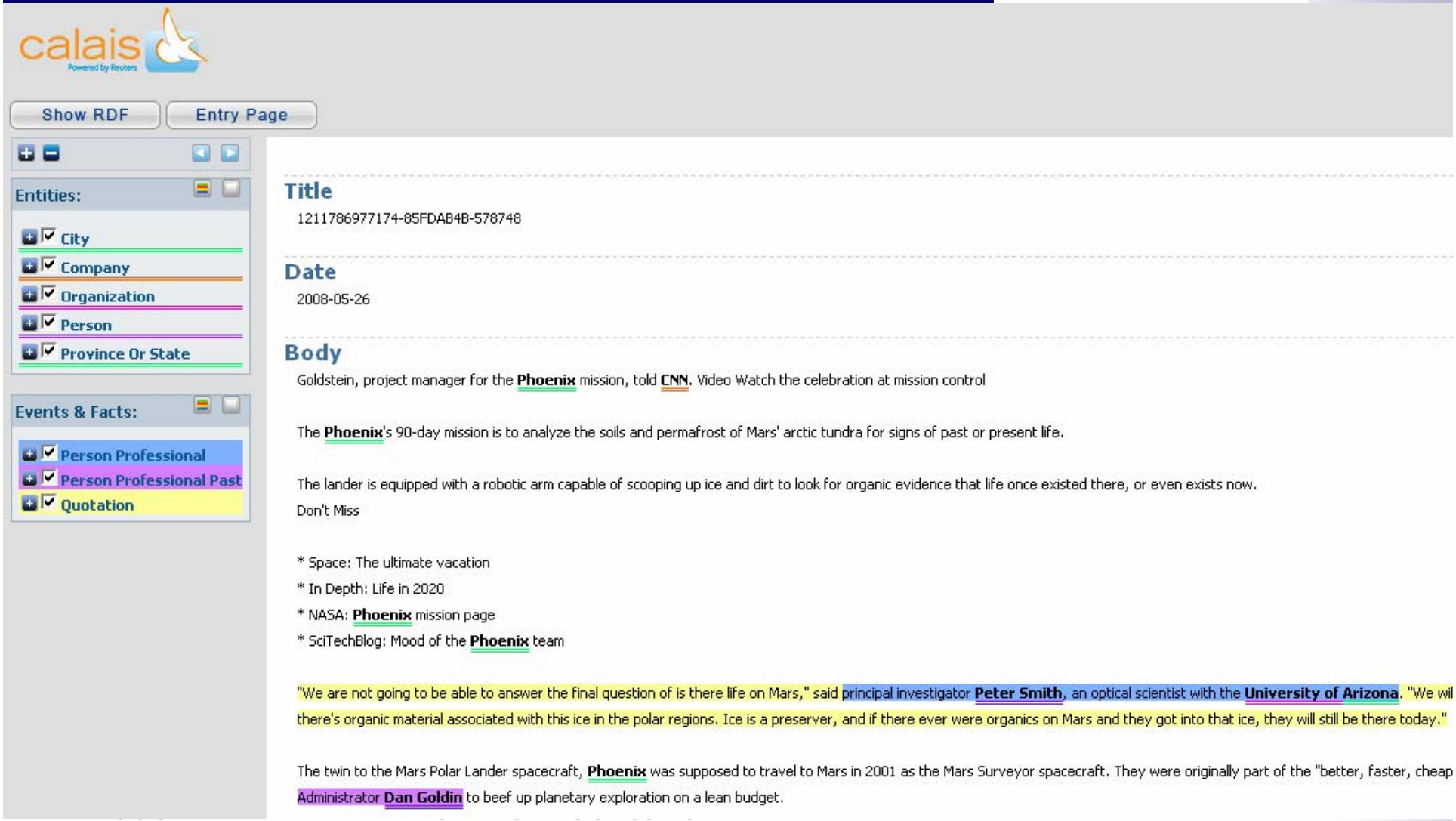
- Aus anderer Perspektive: Person = [„Michael",N] [Granitzer,N]

Vorverarbeitung von Texten
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
- Web Service: Open Calais (http://www.opencalais.com/calaisAPI)
- …

Vorverarbeitung von Texten

Open Calais Document Viewer (http://sws.clearforest.com/calaisviewer/)



Vorverarbeitung von Texten
Beispiel Linked Facts

Inhalt

- Ein paar Zahlen zur Motivation
- Vorverarbeitung von Texten
- **Vektorraummodell**
- Maschinelle Lernmethoden im Überblick
  - Supervised
  - Unsupervised

Vektorraummodell
Ausgangsbasis

- Gegeben: Vorverarbeitung von Dokumenten
  - Tokenization, POS Tagging, Named Entity Extraction
  - Menge von Merkmalen pro Dokument
- Gesucht: Mathematisches Modell für Berechnungen
  - Ähnlichkeits- und Distanzberechnung
  - Addition, Subtraktion von Dokumenten
  - Transformation von Dokumenten

Vektorraummodell
Mathematisches Modell

- Vektorraummodel = mathematisches Modell auf Basis Vektoralgebra/linearer Algebra
- Erzeugung über statistische Auswertung der Merkmale eines Dokumentes
  1. Analyse der Merkmalstypen
  2. Analyse der Merkmale pro Dokument
  3. Merkmalsgewichtung
  4. Merkmalsreduktion
     1. Selektion
     2. Transformation
  5. Aufspannen des Vektorraums

Vektorraummodell
Merkmale im Vektorraum

$D_1 =$"Die Vorlesung gehalten v on Markus Strohmaier . Vorlesung SS 08"

$D_2 =$"Behandelt wird das Vector Space Model, kurz VSM. Vector Definition :"

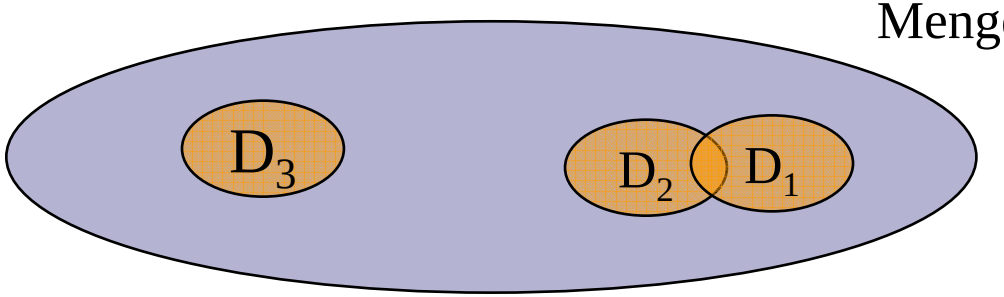
$D_j = \{m_1, m_2, m_3 \ldots m_{k-1}, m_k\}$

$D_1 = \{"Vorlesung" \ldots "Strohmaier", "Vorlesung"\}$

$D_2 = \{"Vector", \ldots, "VSM", "Vorlesung", "VSM"\}$

$FeatureTypes = \{"Vorlesung", "Strohmaier", "Vector", "VSM\}$

Menge der Merkmalstypen



Vektorraummodell
Merkmale im Vektorraum

- Wörterbuch: Die Menge der *m* unterschiedlichen Merkmalstypen
- Dokument
  - Besteht aus einer Menge von Merkmalen
  - Jedes Merkmal ist i.A. eine Zeichenkette im Dokument
  - Jedes Merkmal hat einen Merkmalstyp

Vektorraummodell
Merkmalstyp vs. Merkmalsinstanz

- Merkmalstyp/Dimension/Merkmalsklasse
  - Die Vereinigungsmenge aller unterschiedlichen Merkmale (i.e. Zeichenketten) in Dokumente
  - Unabhängig vom Dokument
- Merkmalsinstanz/Merkmalsvorkommniss
  - Vorkommnis eines Merkmalstyp in einem Dokument
  - Instanz der Zeichenketten in einem Dokument
- Achtung: Merkmal als Term oft mehrdeutig in der Literatur

Vektorraummodell
Dokument ➔ Vektor

- Ziel:

$$\vec{d}_1 =< w_{1,1}, w_{1,2} \ldots w_{1,n-1}, w_{1,n} > \qquad \vec{d}_1 =< 0.2, \quad 0.4, \quad 0.5, \quad 0 >$$

$$\vec{d}_2 =< w_{2,1}, w_{2,2} \ldots w_{2,n-1}, w_{2,n} > \qquad \vec{d}_2 =< 0.0, \quad 0.4, \quad 0.0, \quad 0 >$$

$$\vdots$$

$$\vec{d}_m =< w_{m,1}, w_{m,2} \ldots w_{m,n-1}, w_{m,n} > \qquad \vec{d}_j =< 0,1,0 \ldots 0,1,0 >$$

- Jeder Eintrag im Vektor entspricht einem Merkmalstyp
- Gewichtung des Merkmalstyps nach Wichtigkeit für das Dokument

Merkmalsgewichtung
Binär

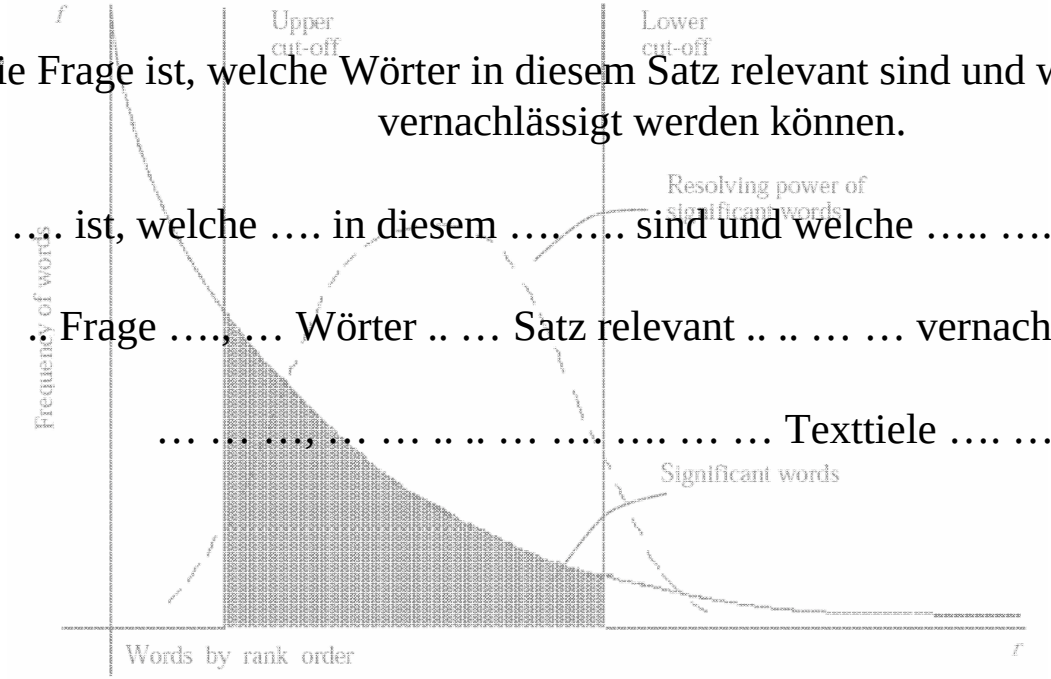
- Führen zu binäre Merkmalsvektoren
- Anwendung von Mengenoperationen als mathematisches Modell
- Nachteil das Stoppwörter wie „und", „oder" etc. gleich wichtig sind wie „sinnvolle" Wörter

$$w_{i,j} = \begin{cases} 1 & merkmal_i \in Dokument_j \\ 0 & merkmal_i \notin Dokument_j \end{cases} \qquad \vec{d}_j = < 0,1,0 \ldots 0,1,0 >$$

Merkmalsgewichtung
Merkmalsverteilung – Zipf's Gesetz

Das Verhältnis der Häufigkeit des Auftretens eines Tokens ist invers proportional zu seiner Position in der Häufigkeitsliste (f*r=const)

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



Die Frage ist, welche Wörter in diesem Satz relevant sind und welche Texttiele vernachlässigt werden können.

Die …. ist, welche …. in diesem …. …. sind und welche ….. ….. werden können.

.. Frage …., … Wörter .. … Satz relevant .. .. … … vernachlässigt … ...

… … …, … … .. .. … …. …. … … Texttiele …. …. ….

aus C. J. Rijsbergen, Information Retrieval

Merkmalsgewichtung
TF-IDF

Berücksichtigung der vorangegangenen Analysen zur Bestimmung der Wichtigkeit eines Merkmals

- TF: Term Frequenz resp. Merkmalsfrequenz: Wie oft kommt ein Merkmal in einem Dokument (bezogen auf dessen Länge) vor
- IDF: Inverse Dokumentfrequenz = 1 /Dokumentfrequenz: In je mehr Dokumenten ein Merkmal vorkommt um so unwichtiger wird es
- Kombination von TF und IDF:

$$w_{i,j} = TF_{i,j} * \log(IDF_i)$$

$$TF_{i,j} = \frac{\left|merkmal_i \cap Dokument_j\right|}{\left|Dokument_j\right|}$$

$$IDF_j = \frac{\left|Dokument_j\right|}{\left|\forall_j merkmal_i \in Dokument_j\right|}$$

Merkmalsgewichtung
TF-IDF

- Repräsentation eines Dokumentes als numerischer Vektor
- Anwendung von Vektorrechnung und Vektoralgebra
- Unterschiedliche Möglichkeiten der TF/IDF Berechnung

$$w_{i,j} = \sqrt{TF_{i,j}} * \sqrt{IDF_i}$$

$$w_{i,j} = \log(TF_{i,j}) * \log(IDF_i)$$

Aufspannen des Vektorraums
Vektorraum Modell - Mathematisch

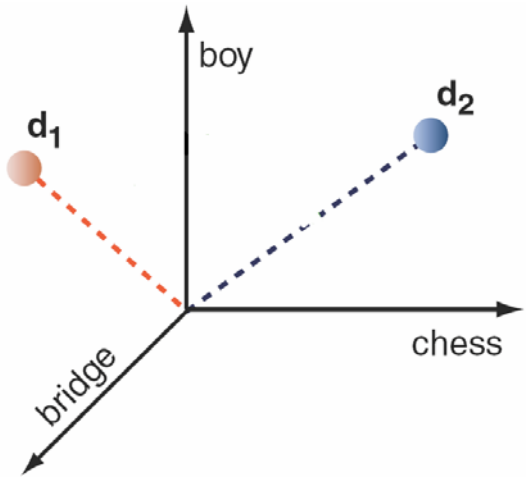
Dokument Term Matrix

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

	m_1	m_2	m_3	m_4	m_5	m_6
d_1	0.2	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	

Aufspannen des Vektorraums
Vektorraum Modell

- Visuelle Interpretation



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

$$D_1 = \{"boy","bridge"\}$$

$$D_2 = \{"boy","chess"\}$$

$$\vec{d}_j = < w_{boy}, w_{chess}, w_{bridge} >$$

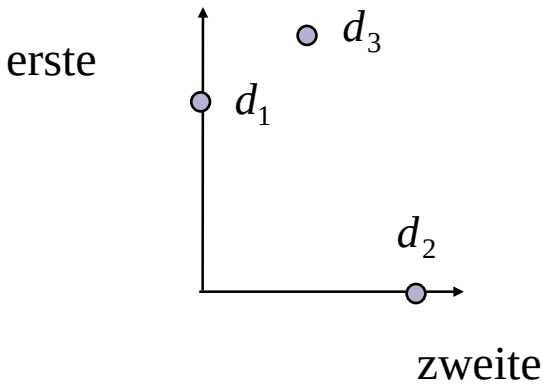
$$\vec{d}_1 = < 0.5,0,0.5 >$$

$$\vec{d}_2 = < 0.5,0.5,0 >$$

- In der Praxis umfasst der Termraum 100.000 Dimensionen und mehr

Aufspannen des Vektorraums
Beispiel

- Dokument 1: „Dies ist der erste Satz“
- Dokument 2: „Dies ist der zweite Satz“
- Dokument 3: „Dies ist der erste erste/zweite Satz“



	$m_1 = erste$	$m_2 = zweite$
d_1	0.035	0
d_2	0	0.035
d_3	0.050	0.025

Operationen im Vektorraum

- Addition von Dokumenten
- Subtraktion von Dokumenten Textdokumente
- Transformation des Vektorraums in einen neuen Vektorraum (Merkmalsselektion & Projektion)
- Ähnlichkeitsberechnung im Vektorraum
- Distanzberechnungen im Vektorraum
- Multidimensionale Skalierung der Vektoren zur Visualisierung)

Operationen im Vektorraum
Ähnlichkeit/Distanz

- Ähnlichkeit/Distanz ermöglicht das Ordnen von Dokumenten
- Relevant in unterschiedlichen Algorithmen
  - Relevance Ranking (IR)
  - Query by Example
  - Klassifikation
  - Clustering
  - Transformation & Projektion

Operationen im Vektorraum
Ähnlichkeiten Binär

- Binärer Vektorraum: Ähnlichkeit ist proportional der Menge der übereinstimmenden Merkmale
- Jaccard Koeffizient
  - Ähnlichkeit = # gemeinsame Merkmale/# vereinigten Merkmale

$$sim(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_j|}$$

Operationen im Vektorraum
Beispiel Jaccard Koeffizient

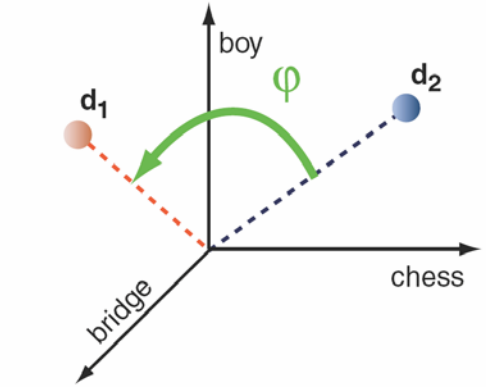
- Beispiel:
  - Dokument 1: „Dies ist der erste Satz“ ➔[1111101]
  - Dokument 2: „Dies ist der zweite Satz“ ➔[1111011]
  - Binäre Gewichtung

Merkmal	Dokument 1	Dokument 2
Dies	1	1
Ist	1	1
Der	1	1
Erste	1	0
Zweite	0	1
Satz	1	1

- Jaccard(Document 1, Document 2) = 4/6

Operationen im Vektorraum
Kosinusähnlichkeit

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



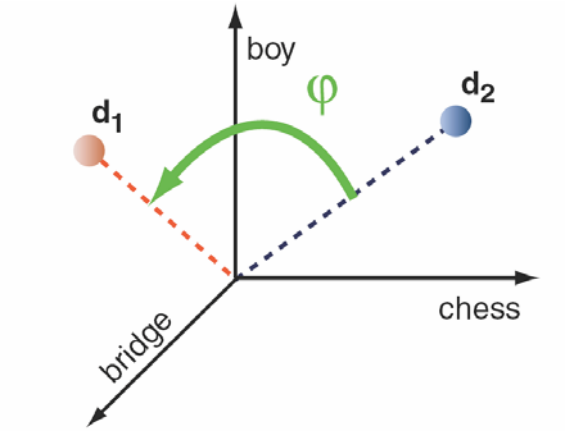
- Problem: Annahme, das Merkmale voneinander unabhängig sind stimmt nicht

Operationen im Vektorraum
Kosinusähnlichkeit - mathematisch

- Skalarprodukt (arithmetische Formel)

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

- Cosinusmaß= Cosinus des Winkels zwischen Query und Dokumentvektor

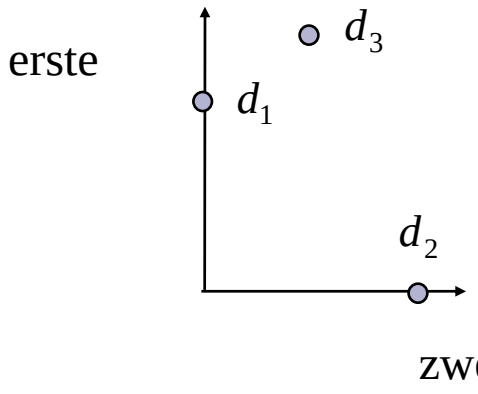


$$sim(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_j \cdot \vec{d}_i}{\left|\vec{d}_j\right| \cdot \left|\vec{d}_i\right|} = \frac{\sum_{k=1}^{|D|} w_{j,k} \cdot w_{i,k}}{\sqrt{\sum_{k=1}^{|D|} w_{j,k}^2} \cdot \sqrt{\sum_{k=1}^{|D|} w_{i,k}^2}}$$

Operationen im Vektorraum
Kosinusähnlichkeit - Beispiel

- Dokument 1: „Dies ist der erste Satz“
- Dokument 2: „Dies ist der zweite Satz“
- Dokument 3: „Dies ist der erste erste/zweite Satz“

	$m_1 = erste$	$m_2 = zweite$
d_1	0.035	0
d_2	0	0.035
d_3	0.050	0.025

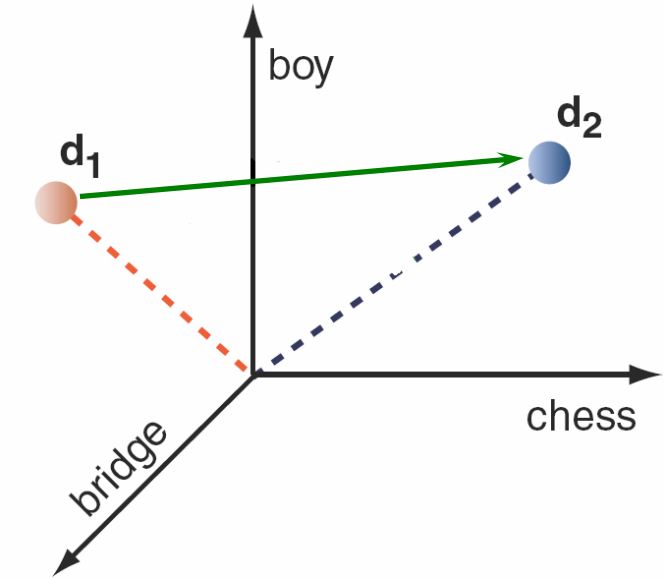


$$sim(d_1, d_2) = \frac{0.035 \cdot 0 + 0 \cdot 0.035}{\sqrt{0.035^2 + 0^2} * \sqrt{0^2 + 0.035^2}} = 0$$

$$sim(d_1, d_3) = \frac{0.035 \cdot 0.05 + 0 \cdot 0.025}{\sqrt{0{,}035^2 + 0^2} * \sqrt{0.05^2 + 0{,}025^2}} = 0{,}89$$

$$sim(d_2, d_3) = \frac{0 \cdot 0.05 + 0.035 \cdot 0.025}{\sqrt{0{,}035^2 + 0^2} * \sqrt{0.05^2 + 0{,}025^2}} = 0{,}44$$

Operationen im Vektorraum
Euklidsche Distanz



$$dist_{euclidean}(\vec{d}_i, \vec{d}_j) = \sqrt{\sum_{\forall m_k} \left(w_{i,k} - w_{j,k}\right)^2}$$

Vektorraummodell
Zusammenfassung

Vorteile:

- Schnell und einfach
- Erstellung des VSM erfolgt in O(n)
- Ähnlichkeitskriterium zwischen Dokumenten
- TFIDF stellt eine bewährte Heurisitk dar (seit 1968)

Nachteile:

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

Inhalt

- Ein paar Zahlen zur Motivation
- Vorverarbeitung von Texten
- Vektorraummodell
- **Maschinelle Lernmethoden im Überblick**
  - Supervised
  - Unsupervised

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
Textklassifikation

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

- Input: hochdimensionale Vektoren

$$\vec{d}_1 =< w_{1,1}, w_{1,2} \ldots w_{1,n-1}, w_{1,n} >$$

$$\vdots$$

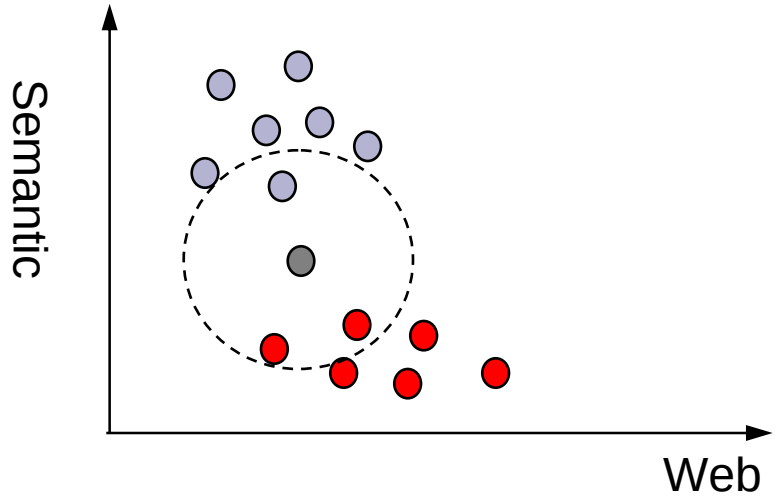
$$\vec{d}_n =< w_{n,1}, w_{n,2} \ldots w_{n,m-1}, w_{n,m} >$$

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

$$< \vec{d}_1, c_1 >, < \vec{d}_1, c_2 > \ldots < \vec{d}_n, c_1 >$$

Textklassifikation
k-Neares Neighbour Classifier

Dokument = Merkmalsvektor



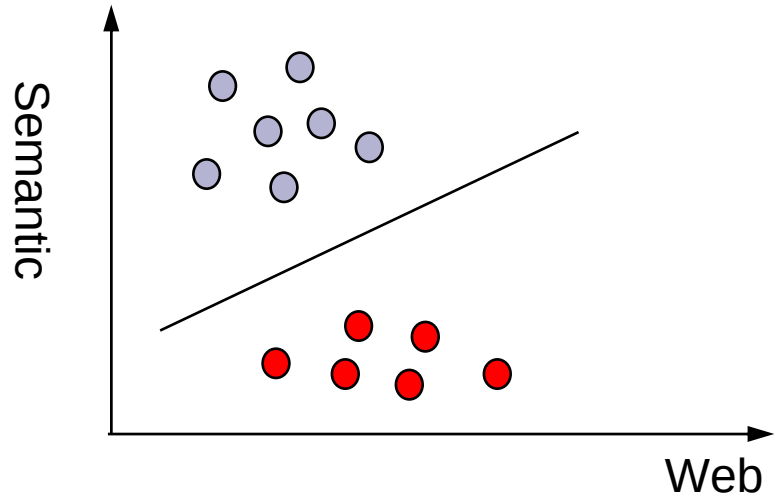
Training: Lazy Learner, d.h. “kein” Training

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
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

Maschinelles Lernen

Anwendungsbereich Supervised Learning

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

Maschinelles Lernen

Unsupervised Learning

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

$$X = \{< x_1 >, < x_2 >, \ldots, < x_{n-1} >, < x_n >\}$$

Ziel: Approximation eines vorgegebenen Modells

- Clustering
- Reinforcement-Learning
- Dimensionalitätsreduktion
- Wahrscheinlichkeitsfunktion

Clustering
Definition

- Gegeben eine Menge an Datenpunkte

$$X = \{< x_1 >, < x_2 >, \ldots, < x_{n-1} >, < x_n >\}$$

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

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

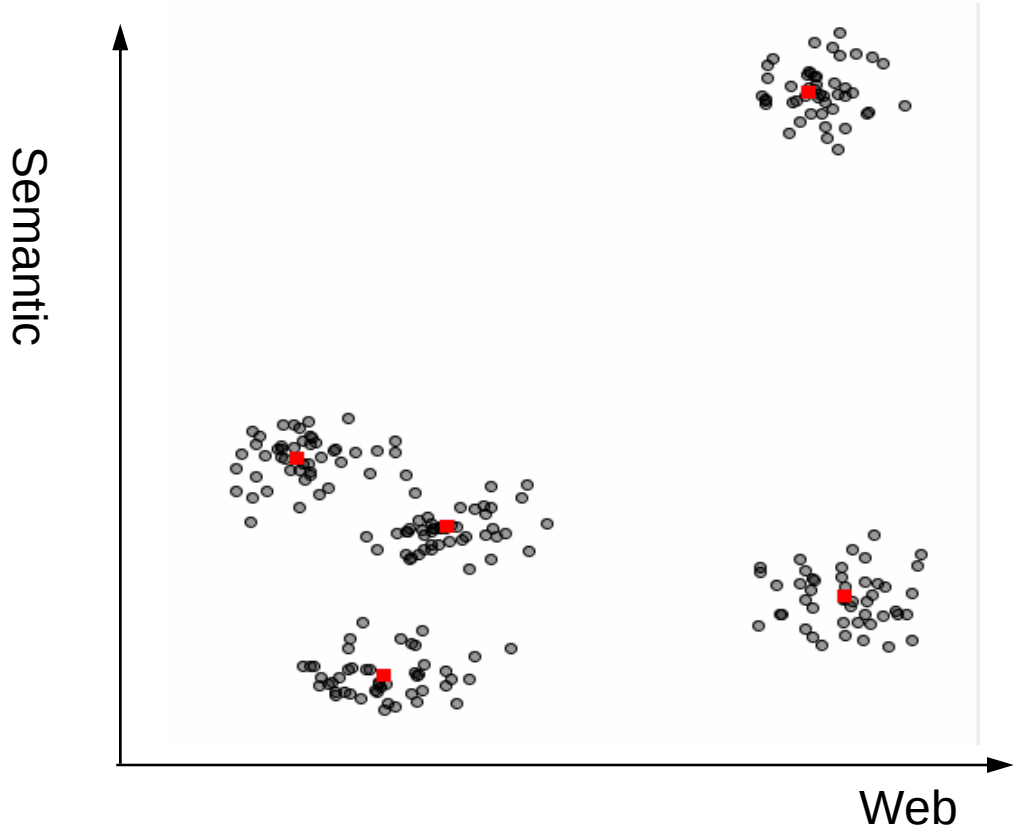
$$C_i \subseteq X; X = \bigcup_i C_i$$

- Hartes Clustering vs. Fuzzy Clustering

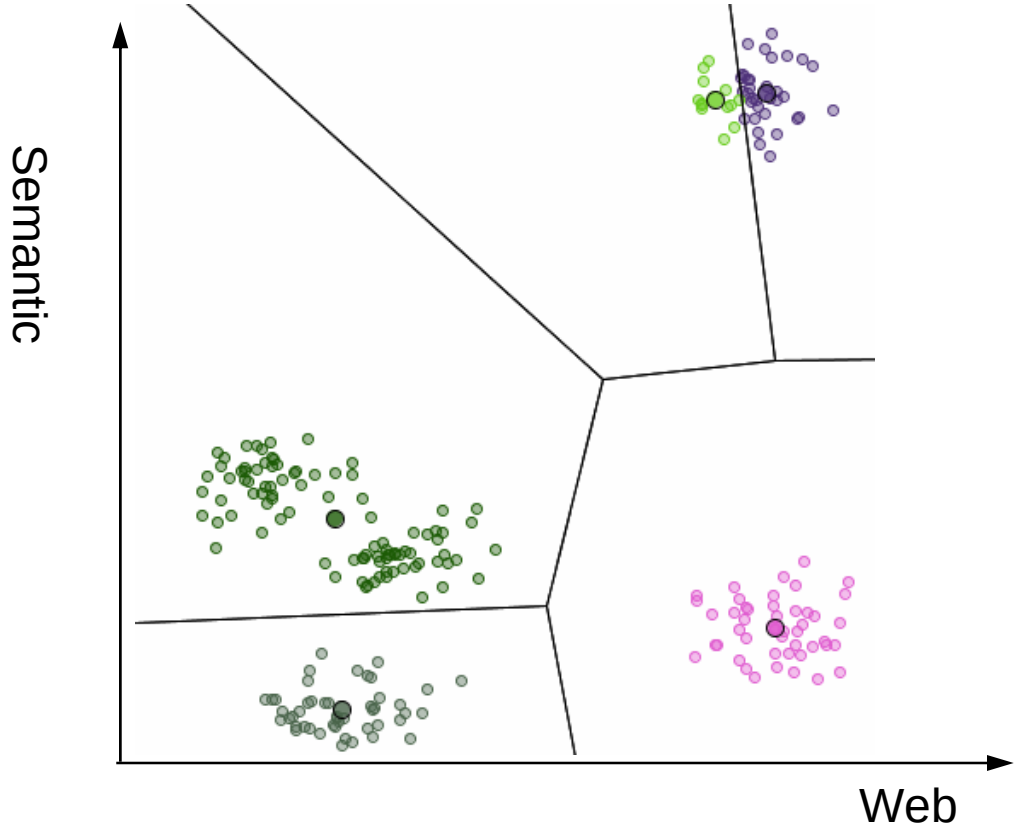
Clustering
Definition

- 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
Beispiel



Clustering
Beispiel



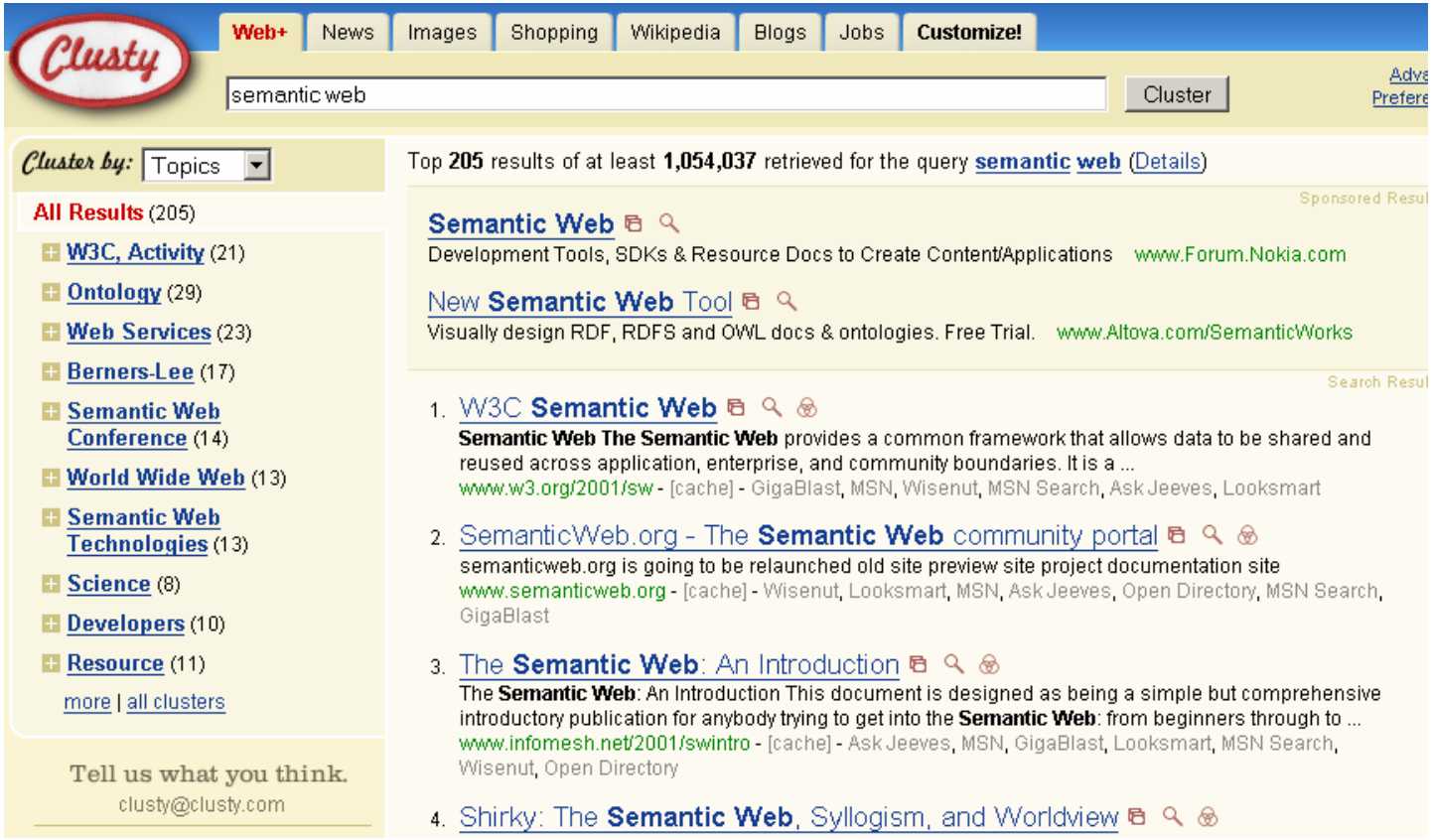
Clustering
Methoden

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

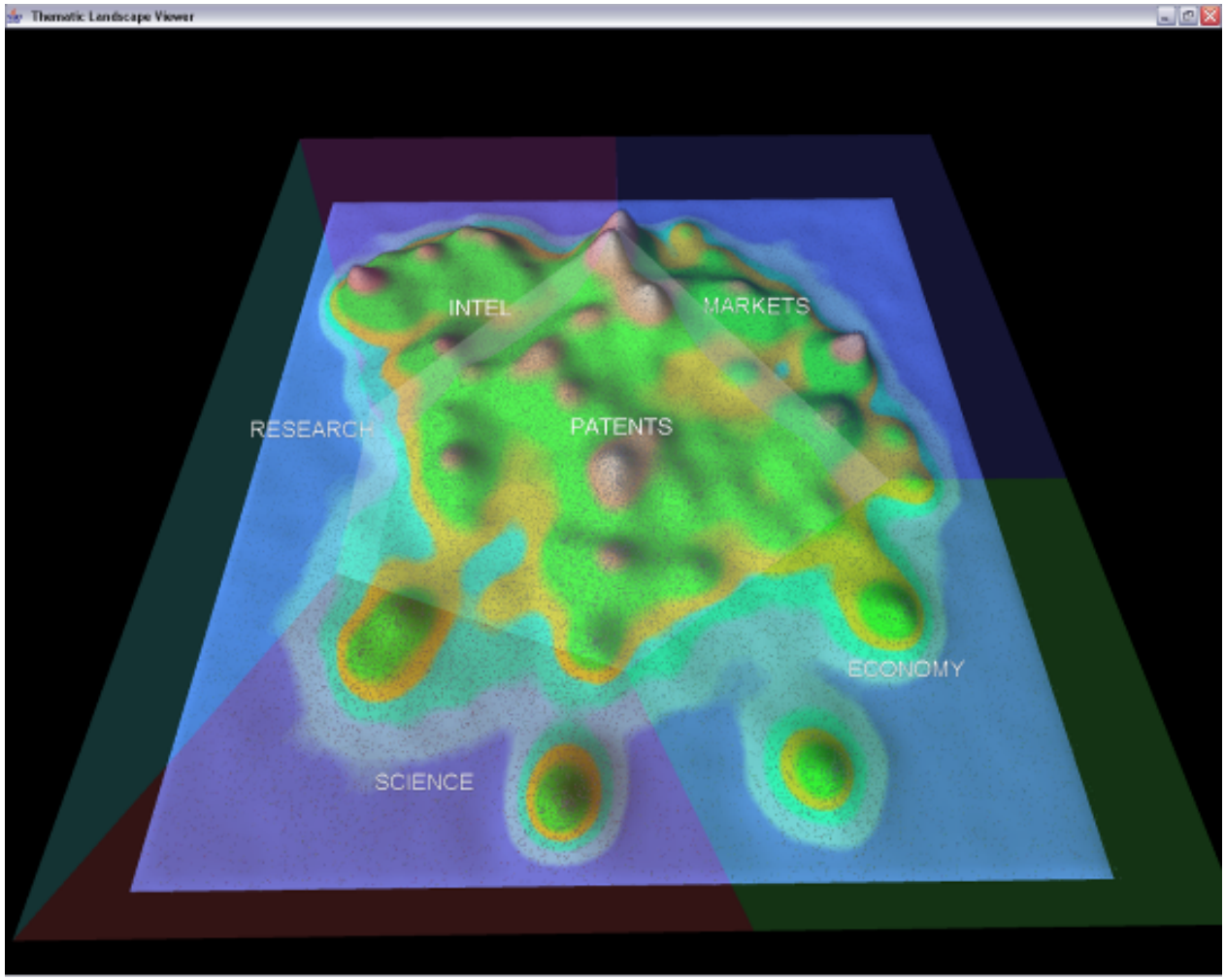
Clustering

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

Clustering
Anwendungsbeispiel



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
    - 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_β-Measure

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

Klasse C_i	True	False
Positive	True Positives (TP)	False Positives (FP)
Negative	False Negatives (FN)	True Negatives (TN)

Evaluierungsmethoden
Supervised

- Macro-Averaging vs. Micro-Averaging

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

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

- Erstellen von Testsamples
  - Split in Training- & Testdaten
  - Random Sampling
  - Cross-Validation

Zusammenfassung

- Linguistische Analysen zur Merkmalsgenerierung aus Text
- Überführung in Vektorform zur Berechnung
  - Gewichtung des Vektorraums
  - Selektion von Merkmalen
  - Transformation des Vektorraums
- Supervised Machine Learning
- Unsupervised Machine Learning

Danke für die Aufmerksamkeit

Michael Granitzer

mgrani@know-center.at

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

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

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)