Retrievalmodelle

Retrievalmodelle

Das Vektorraum-Modell

Sichten auf ein Dokument

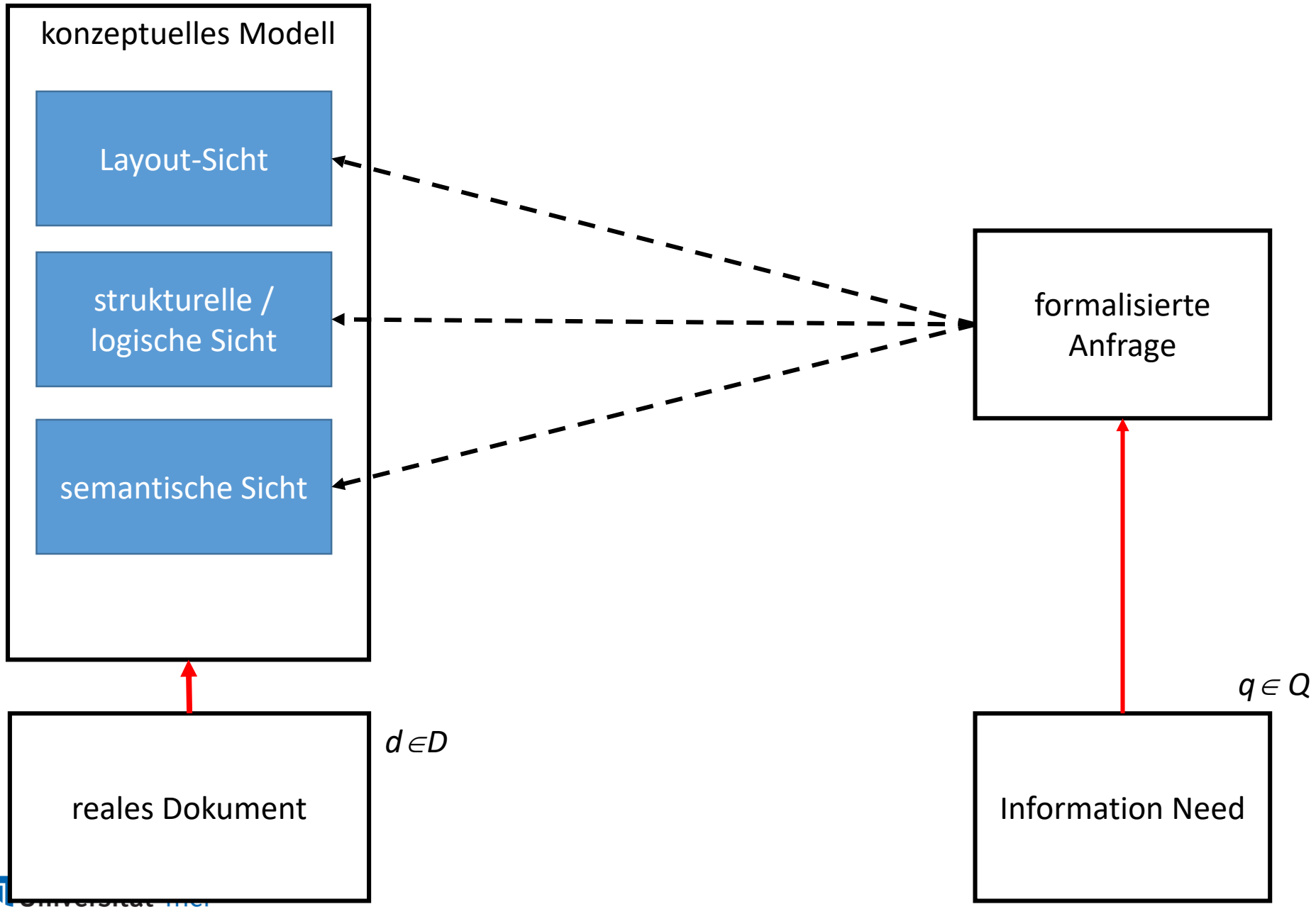
Die Automatisierung von Retrieval-Aufgaben erfordert die **Modellierung** und **Repräsentation** von Dokumenten auf einem Rechner. Dabei lassen sich drei orthogonale Sichten auf den Inhalt unterscheiden:

1. **Layout-Sicht**
   Darstellung eines Dokuments auf einem zweidimensionalen Medium.
2. **Strukturelle bzw. logische Sicht**
   Definiert den Aufbau bzw. die logische Struktur eines Dokuments.
   Beispiel:

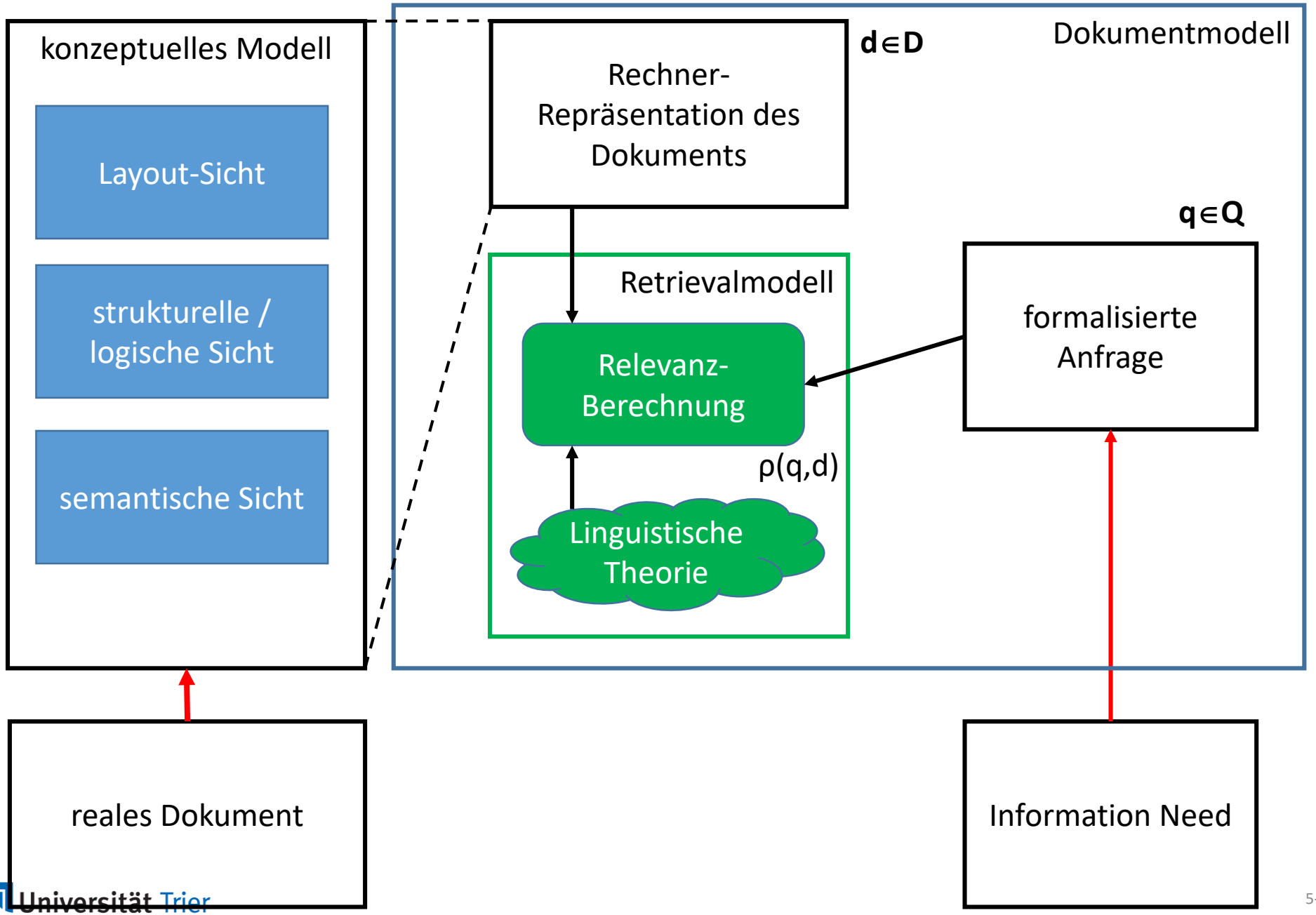
```
\documentclass[twocolumn,german]{article}
\title{...}
\author{...}
\section{...}
```

3. **Semantische Sicht**
   Betrifft die Aussage eines Dokuments und ermöglicht dessen Interpretation.

Vom Dokument zum Dokumentmodell



Vom Dokument zum Dokumentmodell



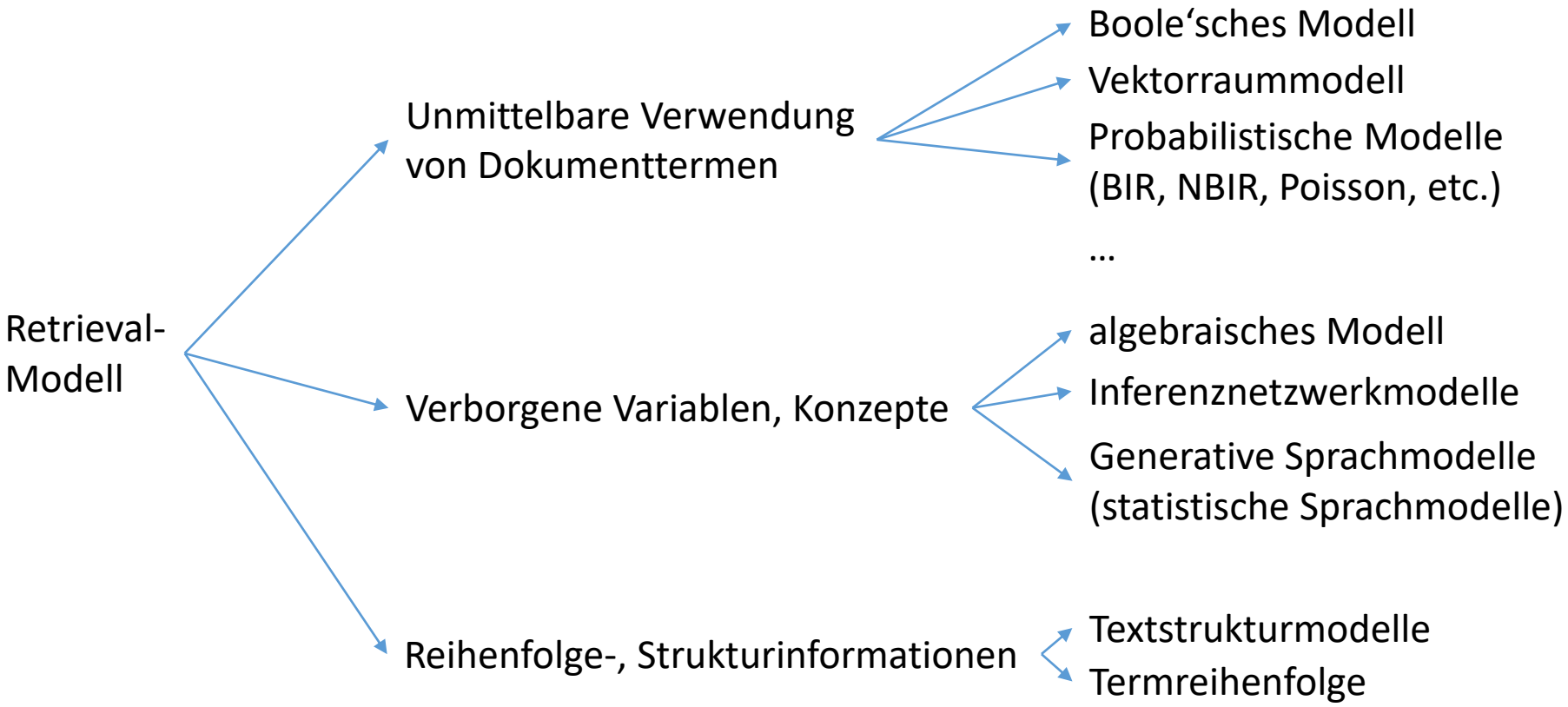
Modelle

Definition: (Dokumentmodell, Retrieval-Modell, Retrieval-Funktion)

Sei *D* eine Menge von Dokumenten und *Q* eine Menge von Anfragen. Ein **Dokument-Modell** für *D*,*Q* ist ein Tupel $(\mathbf{D},\mathbf{Q},\rho_{\mathcal{R}})$, dessen Elemente wie folgt definiert sind:

- **D** ist die Menge der **Repräsentationen der Dokumente** $d \in D$. In $\mathbf{d} \in \mathbf{D}$ können Layout-, logische und semantische Sicht codiert sein.
- **Q** ist die Menge der **formalisierten Anfragen**.
- $\mathcal{R}$ ist ein Retrieval-Modell und formalisiert ein Prinzip, ein Paradigma oder eine linguistische Theorie.
  Auf der Grundlage von $\mathcal{R}$ ist die **Retrieval-Funktion** $\boldsymbol{\rho_{\mathcal{R}}(q,d)}$ definiert. Sie **quantifiziert die Systemrelevanz** zwischen einer formalisierten Anfrage $\mathbf{q} \in \mathbf{Q}$ und einer Dokumentrepräsentation $\mathbf{d} \in \mathbf{D}$:
  $\boldsymbol{\rho_{\mathcal{R}} : Q \times D \rightarrow R}$
  Die von $\boldsymbol{\rho_{\mathcal{R}}}$ berechneten Werte heißen **Retrieval-Werte** (Retrieval Status Value, RSV) oder auch **Scores**.

Taxonomie von Retrieval-Modellen



Klassische Retrieval-Modelle

Die **klassischen Retrieval-Modelle** abstrahieren ein Dokument $d \in D$ zu einer unstrukturierten Menge von Indextermen, die sich quasi unmittelbar und automatisch aus *d* gewinnen lassen.

Die Dokumentrepräsentation **d** eines Dokumentes *d* besteht aus **gewichteten Indextermen**, die aus d stammen.

Unterscheidung der klassischen Retrieval-Modelle:

1. Art und Weise, wie sich Gewichte w_i für die Indexterme t_i berechnen.
2. Art und Weise, wie formalisierte Anfragen **q** konstruierbar sind.
3. Art und Weise, wie sich die Retrieval-Funktion $\boldsymbol{\rho_{\mathcal{R}}(q,d)}$ berechnet.
4. Art und Weise, wie die Menge relevanter Dokumente *R(q)* konstruiert wird.

Boolesches Modell

Dokumentrepräsentationen D:

- Typischerweise bilden normalisierte Terme eines Korpus die Menge der Indexterme T = {t1,…,tm}. Die Repräsentation **d** eines Dokumentes *d* ist eine Abbildung von T nach {0,1}, wobei **d**(w) = 1 bzw. **d**(w) = 0 als „Term in d vorhanden“ bzw. „nicht vorhanden“ interpretiert wird.

Formalisierte Anfragenmenge Q:

- Eine formalisierte Anfrage **q**$\in$**Q** entspricht einer logischen Formel über dem Alphabet Σ=T, in der die Junktoren $\wedge$, $\vee$, $\neg$ und Klammern verwendet werden können.

Retrieval-Funktion $\rho_{\mathcal{R}}$:

- Die Dokumentrepräsentation **d** eines Dokumentes *d* induziert eine Interpretation $\mathcal{I}_d$ für **q**; man setzt $\boldsymbol{\rho}_{\mathcal{R}}(\mathbf{q},\mathbf{d}) = \mathcal{I}_d(\mathbf{q})$.
- Gilt $\boldsymbol{\rho}_{\mathcal{R}}(\mathbf{q},\mathbf{d}) = 1$, wird das Dokument **d** Element der Antwortmenge *R(q)*.

Boolesches Modell

Vorteile:

- Mächtigkeit: Prinzipiell kann mit einer Bool'schen Anfrage jede beliebige Teilmenge von Dokumenten aus einer Kollektion selektiert werden.
- einfache und genaue Implementierbarkeit

Nachteile:

- die Schwarz-Weiß-Aufteilung in die Menge R (bzw. $D \backslash R$) der als relevant (bzw. nicht-relevant) geschätzten Dokumente ist zu streng
- keine Ordnung auf der Antwortmenge R hinsichtlich der geschätzten Relevanz
- die Größe der Antwortmenge ist schwierig zu kontrollieren
- keine Möglichkeit zur Gewichtung von Fragetermen
- umständliche Formulierung von Anfragen
- schlechte Retrieval-Qualität

Vektorraummodell

Dokumentrepräsentationen D.

- Typischerweise bilden die normalisierten Terme, ggf. nach Entfernung der Stoppwörter, eines Korpus die Menge der Indexterme T = {t1,…,tm}. Der Wertebereich der Termgewichte ist $\mathbb{R}$ (reelle Zahlen); für die Gewichtsberechnung existieren verschiedene Konzepte.

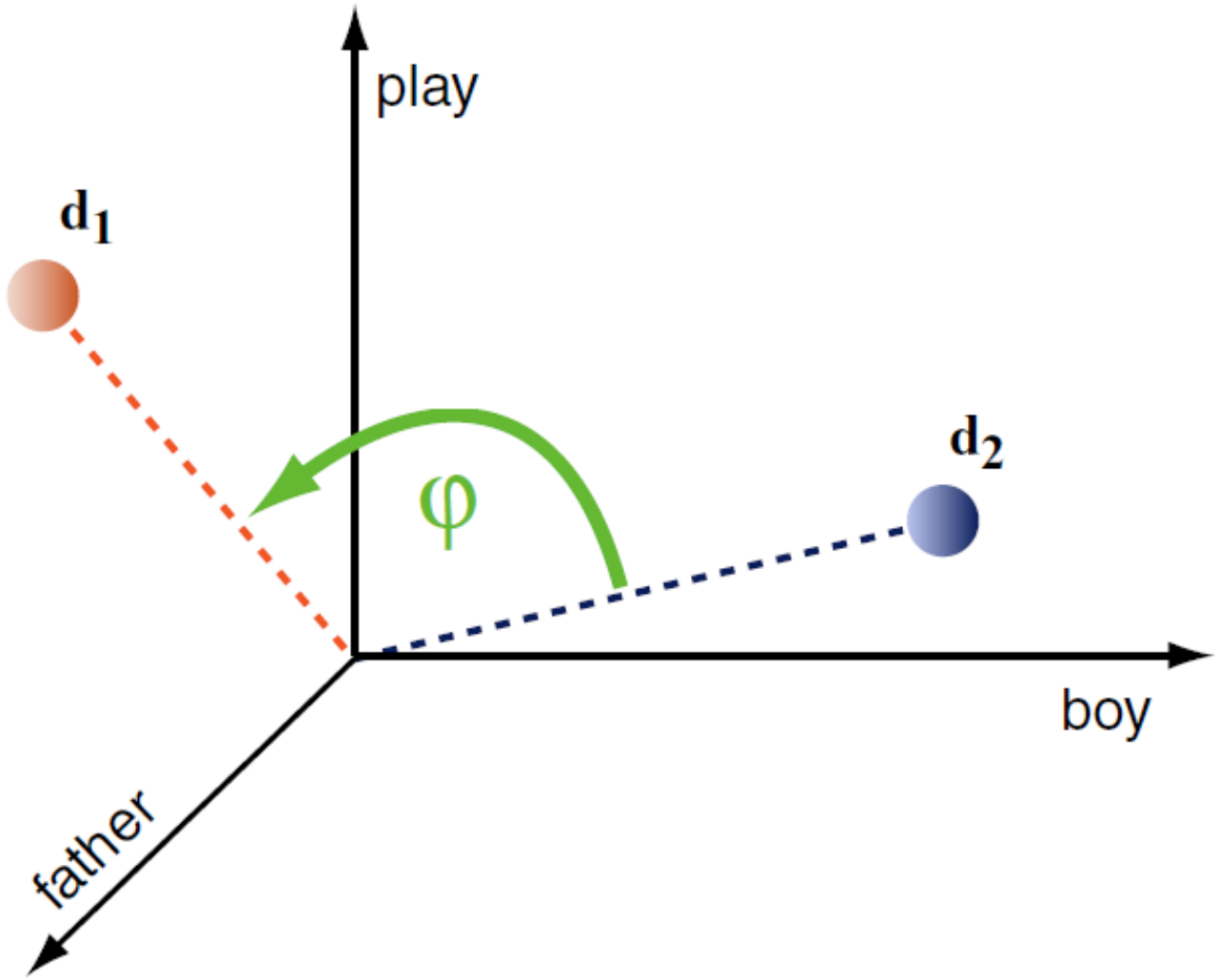
Formalisierte Anfragenmenge Q.

- Eine formale Anfrage $\mathbf{q} \in \mathbf{Q}$ hat den gleichen Aufbau wie eine Dokumentrepräsentation $\mathbf{d} \in \mathbf{D}$.

Retrieval-Funktion $\rho_{\mathcal{R}}$.

- Dokumentrepräsentationen und formalisierte Fragen werden als Punkte eines **orthonormalen Vektorraums** interpretiert, der durch die Terme aufgespannt wird.
- Wichtige Ansätze zur Berechnung von $\boldsymbol{\rho}_{\mathcal{R}}$ sind das **Cosinus-Ähnlichkeitsmaß** und die euklidische Distanz.

Retrieval-Funktion Kosinus-Ähnlichkeit



Einschub: Vektorrechnung

Das **Skalarprodukt** $\boldsymbol{a}^T \boldsymbol{b}$ zweier (Spalten-)Vektoren $\boldsymbol{a}$ und $\boldsymbol{b}$ ist definiert als

$$\boldsymbol{a}^T \boldsymbol{b} = \sum_i a_i \cdot b_i$$

Die **Länge** (oder euklidische Norm, 2-Norm) eines Vektors $\boldsymbol{a}$ ist definiert als

$$\|\boldsymbol{a}\| = \sqrt{\sum_i a_i^2}$$

Retrievalfunktion Kosinus-Ähnlichkeit

Definition des Skalarproduktes $\mathbf{a}^T\mathbf{b}$ zwischen zwei Vektoren **a** und **b**, mit φ als Winkel zwischen **a** und **b**

$$\mathbf{a}^T\mathbf{b} = ||\mathbf{a}|| \cdot ||\mathbf{b}|| \cdot \cos(\varphi)$$

$$\Leftrightarrow \quad \cos(\varphi) = \frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}$$

Normalisiert man **a** und **b** – hier bezeichnet als **a'** und **b'** – gilt:

$$\cos(\varphi) = \frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||} = \frac{(\mathbf{a}')^T\mathbf{b}'}{||\mathbf{a}'|| \cdot ||\mathbf{b}'||} = (\mathbf{a}')^T\mathbf{b}' = \sum_{i=1}^{n} a_i' \cdot b_i'$$

(**D**,**Q**,$\rho_{\mathcal{R}}$) mit cos-Ähnlichkeitsmaß:

Definition von $\rho_{\mathcal{R}}(\mathbf{q}, \mathbf{d})$ als $\cos(\varphi)$, mit φ als Winkel zwischen $\mathbf{q}$ und $\mathbf{d}$.

VRM-Beispiel

$$\mathbf{d}_1 = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix}$$

Wie soll man die Termgewichte wählen?

VRM-Beispiel

$$\mathbf{d}_1 = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix} = \begin{pmatrix} \texttt{chrysler} & 1 \\ \texttt{usa} & 4 \\ \texttt{cat} & 3 \\ \texttt{dog} & 7 \\ \texttt{mouse} & 5 \end{pmatrix}$$

Einfachster Ansatz: Termfrequenz

VRM-Beispiel

$$\mathbf{d}_1 = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix} = \begin{pmatrix} \texttt{chrysler} & 1 \\ \texttt{usa} & 4 \\ \texttt{cat} & 3 \\ \texttt{dog} & 7 \\ \texttt{mouse} & 5 \end{pmatrix}$$

$$\mathbf{d}'_1 = \begin{pmatrix} \texttt{chrysler} & 0.1 \\ \texttt{usa} & 0.4 \\ \texttt{cat} & 0.3 \\ \texttt{dog} & 0.7 \\ \texttt{mouse} & 0.5 \end{pmatrix}, \quad \mathbf{d}'_2 = \begin{pmatrix} \texttt{chrysler} & 0.4 \\ \texttt{usa} & 0.1 \\ \texttt{cat} & 0.7 \\ \texttt{dog} & 0.5 \\ \texttt{mouse} & 0.3 \end{pmatrix}$$

Normierung, so dass Länge des Vektors 1 ist

VRM-Beispiel

$$\mathbf{d}_1 = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix} = \begin{pmatrix} \texttt{chrysler} & 1 \\ \texttt{usa} & 4 \\ \texttt{cat} & 3 \\ \texttt{dog} & 7 \\ \texttt{mouse} & 5 \end{pmatrix}$$

$$\mathbf{d}'_1 = \begin{pmatrix} \texttt{chrysler} & 0.1 \\ \texttt{usa} & 0.4 \\ \texttt{cat} & 0.3 \\ \texttt{dog} & 0.7 \\ \texttt{mouse} & 0.5 \end{pmatrix}, \qquad \mathbf{d}'_2 = \begin{pmatrix} \texttt{chrysler} & 0.4 \\ \texttt{usa} & 0.1 \\ \texttt{cat} & 0.7 \\ \texttt{dog} & 0.5 \\ \texttt{mouse} & 0.3 \end{pmatrix}$$

Der Winkel φ zwischen $\mathbf{d}'_1$ und $\mathbf{d}'_2$ ist etwa $38°$, $\cos(\varphi) \approx 0.79$.

Termhäufigkeit

- Bereits 1957 stellte H. P. Luhn in [Luh57] die These auf, dass die **Termverteilung** in einem Dokument dessen Inhalt recht genau widerspiegelt.
- Daher sei die **Termhäufigkeit**, also die Anzahl der Auftreten eines Terms in einem Dokument, für die Relevanz eines Dokuments bzgl. einer Anfrage wichtig. Die Termverteilung und -häufigkeit sollten deshalb bei der Indizierung aufgezeichnet werden.

[Lun57] Hans Peter Luhn. **A statistical approach to mechanized encoding and searching of literary information**. IBM Journal of Research and Development **1**(4), 309-317,1957.

Termgewichte

- Die **Term-Dokument-Inzidenzmatrix** (vgl. Kapitel 2) des Booleschen Retrieval-Modells enthält für jeden Term t und jedes Dokument d einen Eintrag **m(t,d)**. Sein Wert ist 1, falls t in d auftritt und 0 sonst.
- Wir ersetzen nun die binäre Angabe, ob ein Term in dem Dokument auftritt oder nicht, durch ein **Gewicht w(t,d)**, das im Zusammenhang mit der Anzahl der Auftreten des Terms in dem jeweiligen Dokument steht.
- **Term Frequency** oder **Term-Häufigkeit** bezeichnet das Gewichtungsschema, in dem direkt die Anzahl $tf_{t,d}$ der Auftreten des Terms t in Dokument d als Gewicht verwendet wird.

Term-Dokument Häufigkeitsmatrix

Eine **Term-Dokument-Häufigkeitsmatrix M** enthält eine Zeile für jeden Term $t \in V$ aus dem Vokabular V und eine Spalte für jedes in der Dokumentkollektion D vorkommende Dokument $d \in D$. Tritt t in dem Dokument d an k Stellen auf, so enthält das Matrixelement m(t,d) den Wert k, sonst eine 0:

$$m(t,d) = \begin{cases} k & \text{falls } t \text{ in } d \text{ an } k \text{ Stellen vorkommt} \\ 0 & \text{sonst} \end{cases}$$

In diesem Modell wird also jedes Dokument *d* durch einen Vektor $(w_{1,d},\ldots,w_{|V|,d})$ mit **Termgewichten** repräsentiert, dessen i-te Komponente die Häufigkeit $tf_{t,d}$ des Terms t_i in dem Dokument *d* angibt.

Beispiel

Wir betrachten wieder den Datenbestand der Shakespeare-Dramen aus Kapitel 2, und einen Ausschnitt aus der entsprechenden Term-Dokument Häufigkeitsmatrix für die Terme **Antony**, **Brutus**, **Caesar**, **Calpurnia** , **Cleopatra**, **mercy**, **worser**.

	Antony and Cleopatra	**Julius Caesar**	**The Tempest**	**Hamlet**	**Othello**	**Macbeth**
Antony	157	73	0	0	0	1
Brutus	4	157	0	2	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	8	5	8
worser	2	0	1	1	1	0

Bag-of-Words-Modell

Repräsentiert man Dokumente durch Term-Häufigkeitsvektoren, so wird die **Wortordnung innerhalb eines Dokuments** nicht berücksichtigt.

Beispiel:

```
John is quicker than Mary
```

und

```
Mary is quicker than John
```

haben die gleiche Repräsentation.

Dieser Ansatz wird auch als **Bag-of-Words-Modell** (auf Deutsch entsprechend **Wort-Multimengen-Modell**) bezeichnet.

- Positionale Indexe können demgegenüber die Wortreihenfolge darstellen.
- Es gibt verschiedene Ansätze, um Information über die Wortreihenfolge ganz oder teilweise einzubeziehen.

Absolute Termhäufigkeit

Ziel:

- Einsatz der Termhäufigkeit $tf_{t,d}$ zur Bestimmung von Scores für das Retrieval.
- Die absolute Anzahl von Auftreten eines Terms t in dem Dokument d ist als Maß nicht geeignet.
- Ein Dokument d1, in dem t 10-mal auftritt, ist nicht unbedingt 10-mal so relevant, wie ein Dokument d2, in dem t nur einmal auftritt.
- In bestimmten Kontexten sind „typische“ und deshalb häufig vorkommende Termini wie Stoppwörter zu behandeln.

Absolute Termhäufigkeit

Eine Möglichkeit, den Einfluss der absoluten Termhäufigkeit etwas abzumildern, ist der Einsatz eines **logarithmischen Häufigkeitsmaßes** als Termgewicht im Dokumentvektor.

Beispiel:

$$w(t,d) = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{falls } tf_{t,d} > 0 \\ 0 & \text{sonst} \end{cases}$$

Die Gewichte für ein Dokument d1, in dem t 10-mal auftritt, und ein Dokument d2, in dem t einmal auftritt, sind dann

$w_{t,d1} = 1 + \log10(10) = 2$

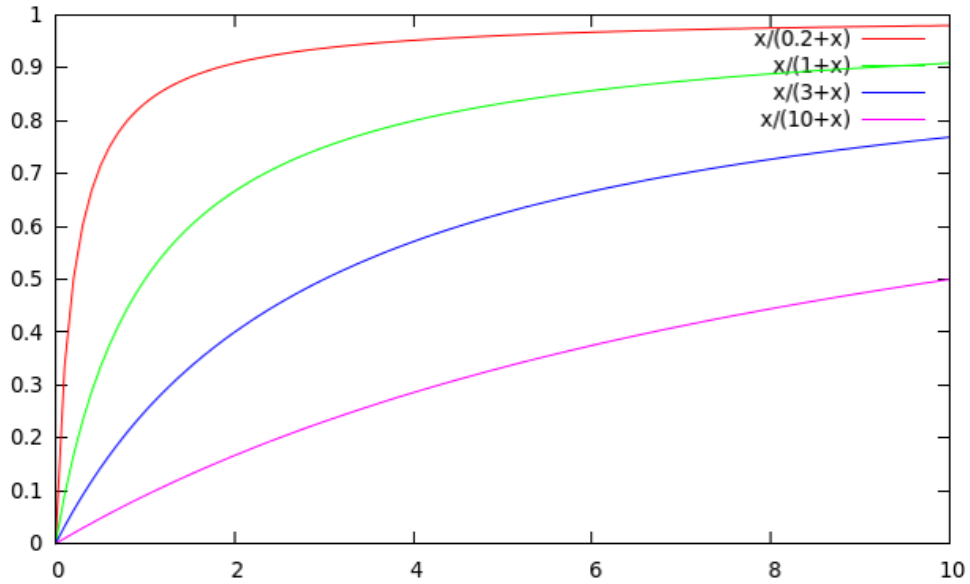
$w_{t,d2} = 1 + \log10(1) = 1$

Absolute Termhäufigkeit

In der Praxis verwendet man u.a. folgende Dämpfungsfunktion:

$$w_{t,d} = \frac{tf_{t,d}}{K + tf_{t,d}}$$

Dies führt dazu, dass das Termgewicht immer zwischen 0 und 1 liegt:



Folgende Alternative sorgt dafür, dass das Gewicht für tf = 1 den Wert 1 hat (und für größere tf-Werte >1 wird):

$$w_{t,d} = \frac{(K+1)tf_{t,d}}{K + tf_{t,d}}$$

Dokumenthäufigkeit

Bei der Gewichtung mehrerer Terme in einer Anfrage spielt es eine Rolle, wie häufig jeder Term in der Dokumentkollektion vorkommt:

- "**Seltene**" Terme werden als **signifikanter**, genauer: **trennschärfer**, angesehen als "häufige".
  Ein Dokument, das einen seltenen Term enthält, ist für diesen Term mit großer Wahrscheinlichkeit relevant. Seltene Terme sollten also ein hohes Termgewicht erhalten.
- Ein Dokument, das einen **häufigen** Term enthält, ist sicherlich relevanter für diesen Term als ein Dokument, das den Term nicht enthält.
  Andererseits ist der häufige Term **weniger trennscharf**.
  Häufige Terme sollten daher berücksichtigt werden, aber ein geringeres Termgewicht bekommen als seltene Terme.

Dokumenthäufigkeit

Sei D eine Dokumentkollektion. Sei t ein Term des Vokabulars. Dann bezeichnet die **Dokumenthäufigkeit** (document frequency) $\mathbf{df_t}$ die Anzahl der Dokumente $d \in D$, in denen t auftritt.

Nach obigen Überlegungen gilt

- **Hohe Dokumenthäufigkeit** df_t bedeutet **geringe Signifikanz** oder Trennschärfe von t.
- **Geringe Dokumenthäufigkeit** df_t bedeutet **hohe Signifikanz** (Trennschärfe) von t.

Inverse Dokumenthäufigkeit

Sei D eine Dokumentkollektion, die N = |D| Dokumente enthält. Für einen Term t des Vokabulars ist die **inverse Dokumenthäufigkeit** (inverse document frequency) **idf_t von t in der Kollektion D** definiert durch

$$idf_t := \log \frac{N}{df_t}$$

Bemerkung:

Da $df_t \leq N$ gilt, ist $0 \leq idf_t$.

- Hohe inverse Dokumenthäufigkeit idf_t bedeutet hohe Trennschärfe von t.
- Geringe inverse Dokumenthäufigkeit idf_t bedeutet geringe Trennschärfe von t.
- Der Einfluss der Dokumenthäufigkeit wird durch das logarithmische Maß gedämpft.

Einschub: Reuters RCV1-Kollektion

Reuters RCV1-Kollektion:

- (alter und kleiner) Standardcorpus für die Evaluierung von Suchmaschinen.
- enthält englische Agenturmeldungen aus den Jahren 1995 und 1996 (ein Jahr).

Für heutige Verhältnisse sehr klein und regulär, wird daher in der Regel nicht mehr zur Evaluierung von IR-Systemen verwendet.

Heutige Kollektionen wie z.B. **TREC ClueWeb09:**

- wesentlich größer (bis 1 Milliarde Dokumente)
- inhaltlich wie strukturell diverser (Dokumente aus dem Web).

Statistiken von Reuters RCV1

Symbol	Metrik	Wert
N	Dokumente	800.000
L	Ø #Token pro Dokument	200
M	Terme	400.000
	Ø #Bytes pro Token (inkl. Leerzeichen/Punktuation)	6
	Ø #Bytes pro Token (ohne Leerzeichen/Punktuation)	4,5
	Ø #Bytes pro Term	7,5
	nicht-positionale Postings	100.000.000

Alle Werte aus [MRS08] und gerundet.

Beispiel

In der **Reuters-Kollektion** mit 800000 Dokumenten findet man folgende idf-Werte:

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Kollektionshäufigkeit

Kollektionshäufigkeit (collection frequency) eines Terms t: Anzahl der Auftreten von t in der gesamten Dokumentkollektion.

$$cf_t := \sum_{d \in D} tf_{t,d}$$

Grundsätzlich könnte man der Termgewichtung statt der Dokumenthäufigkeit auch die Kollektionshäufigkeit zugrunde legen. Die Dokumenthäufigkeit erweist sich jedoch als das besser geeignete Maß.

tf·idf-Gewichtung

Das **tf·idf-Gewichtungsschema** ordnet einem Term t des Vokabulars das wie folgt definierte Gewicht $tf{\cdot}idf_{t,d}$ in dem Dokument $d \in D$ der Dokumentkollektion zu:

$$w_{t,d} := tf{\cdot}idf_{t,d} := tf_{t,d} \cdot idf_t = tf_{t,d} \cdot \log \frac{N}{df_t}$$

Das tf·idf - Schema ist das bekannteste und am weitesten verbreitete (aber in der Regel nicht das beste!) Gewichtungsschema des Information Retrieval.

$tf{\cdot}idf_{t,d}$ ist als ein Bezeichner für das tf·idf-Gewicht des Terms t in dem Dokument d zu interpretieren, nicht als Produkt!

tf·idf-Gewichtung

Seien nun die Termgewichte gemäß tf·idf-Gewichtungsschema definiert durch

$$w_{t,d} = tf_{t,d} \cdot \log \frac{N}{df_t}$$

Falls t in d vorkommt, ist das Termgewicht $w_{t,d}$ also

- **am höchsten**, wenn t häufig in d, aber insgesamt in einer geringen Zahl von Dokumenten der Kollektion auftritt;
- **geringer**, wenn t seltener in d oder insgesamt in einer größeren Zahl von Dokumenten der Kollektion auftritt;
- **am geringsten**, wenn t in praktisch allen Dokumenten auftritt.

Dokumentvektoren

Wenn man die tf·idf-Gewichtung anwendet, wird jedes Dokument d durch einen Vektor

$$(w_{1,d},\ldots,w_{|V|,d}) = (tf{\cdot}idf_{1,d},\ldots,tf{\cdot}idf_{|V|,d})$$

repräsentiert.

Für Terme $t \in V$ des Vokabulars V, die nicht in d vorkommen, hat das entsprechende Gewicht den Wert 0.

Vektorraum der Dokumente

Sei D die Dokumentkollektion. Sei V das Termvokabular.

- Jedes Dokument $d \in D$ wird durch einen Vektor $(w_{1,d},\ldots,w_{|V|,d}) \in \mathbb{R}^{|V|}$ repräsentiert.
- Zusammen bilden die Dokumente einen $|V|$-dimensionalen reellen Vektorraum.
- Die Terme $t \in V$ des Vokabulars (bzw. strenggenommen ihre IDs) bilden die Dimensionen des Vektorraums.
- Der Vektorraum der Dokumente ist im Allgemeinen von sehr hoher Dimension.
- Bereits in ClueWeb09 mit “nur” einer Milliarde Dokumenten gibt es mehr als 400 Millionen verschiedene Terme!

Cosinus-Ähnlichkeit von Dokumenten

Die **Cosinus-Ähnlichkeit** (cosine similarity) sim(**d**,**d'**) zweier (Dokument-)Vektoren **d** und **d'** ist definiert durch

$$sim(\boldsymbol{d}, \boldsymbol{d}') := \frac{\boldsymbol{d}^T \boldsymbol{d}'}{\|\boldsymbol{d}\| \|\boldsymbol{d}'\|}$$

Sei V das Vokabular und sei M := |V|. Dann gilt für die Cosinus-Ähnlichkeit

$$sim(\boldsymbol{d}, \boldsymbol{d}') = \frac{\sum_{i=1}^{M} d_i d'_i}{\sqrt{\sum_{i=1}^{M} d_i^2} \sqrt{\sum_{i=1}^{M} d'^2_i}}$$

Beispiel: Cosinus-Ähnlichkeiten von Dokumentvektoren

Betrachten wir wieder Dokumente aus der **Reuters-Kollektion**.

	idf	Doc1			Doc2			Doc3		
Term		tf	tf•idf	tf•idf (norm.)	tf	tf•idf	tf•idf (norm.)	tf	tf•idf	tf•idf (norm.)
car	1,65	27,00	44,55	0,8974	4,00	6,60	0,0756	24,00	39,60	0,5953
auto	2,08	3,00	6,24	0,1257	33,00	68,64	0,7867	0,00	0,00	0,0000
insurance	1,62	0,00	0,00	0,0000	33,00	53,46	0,6127	29,00	46,98	0,7062
best	1,5	14,00	21,00	0,4230	0,00	0,00	0,0000	17,00	25,50	0,3833
Eu.-Length			49,65	1,0000		87,25	1,0000		66,52	1,0000

Für die Cosinus-Ähnlichkeit der Dokumente untereinander gilt:

sim(Doc1,Doc2)

= 0,8974·0,0756 + 0,1257·0,7867 + 0,0000·0,6127 + 0,4230·0,0000

= 0,1668

sim(Doc1,Doc3)

= 0,8974·0,5953 + 0,1257·0,0000 + 0,0000·0,7062 + 0,4230·0,3833

= 0,6963

sim(Doc2,Doc3)

= 0,0756·0,5953 + 0,7867·0,0000 + 0,6127·0,7062 + 0,0000·0,3833

= 0,4777

Anfragen als Vektoren

Anfragen werden im Vektorraummodell als **Freiformanfragen** aufgefasst, die (nur) durch eine Menge von Termen des Vokabulars spezifiziert werden.

Beispiel: brutus caesar

Ein fundamentales Konzept des Vektorraummodells besteht darin, dass auch **Anfragen als Vektoren** repräsentiert werden.

Sei D die Dokumentkollektion. Sei V das Termvokabular. Dann wird jede Anfrage $q \subseteq V$ an D durch einen Vektor

$(w_{1,q},\ldots,w_{|V|,q}) \in \mathbb{R}^{|V|}$

mit geeigneten Gewichten $w_{i,q}$ als Koeffizienten repräsentiert.

Die Gewichte orientieren sich dabei nur an **der Häufigkeit der Terme in der Anfrage** (ohne idf-Komponente)

Ähnlichkeit von Anfrage und Dokument

Der Score eines Dokuments d für die Anfrage q wird als **Cosinus-Ähnlichkeit** sim(**d**,**q**) der entsprechenden Vektoren berechnet:

$$sim(\boldsymbol{d}, \boldsymbol{q}) := \frac{\boldsymbol{d}^T \boldsymbol{q}}{\|\boldsymbol{d}\| \|\boldsymbol{q}\|}$$

Beispiel: Anfrage im Vektorraummodell

Wir betrachten die Anfrage **car insurance** und die Dokumente von eben:

	idf	Doc1			Doc2			Doc3		
Term		tf	tf•idf	tf•idf (norm.)	tf	tf•idf	tf•idf (norm.)	tf	tf•idf	tf•idf (norm.)
car	1,65	27,00	44,55	0,8974	4,00	6,60	0,0756	24,00	39,60	0,5953
auto	2,08	3,00	6,24	0,1257	33,00	68,64	0,7867	0,00	0,00	0,0000
insurance	1,62	0,00	0,00	0,0000	33,00	53,46	0,6127	29,00	46,98	0,7062
best	1,5	14,00	21,00	0,4230	0,00	0,00	0,0000	17,00	25,50	0,3833
Eu.-Length			49,65	1,0000		87,25	1,0000		66,52	1,0000

Der (normierte) Anfragevektor ist $q = \left(\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 0\right)^T = (0.7071, 0, 0.7071, 0)$

Für die Scores der Dokumente für die Anfrage q gilt:

sim(Doc1,q)

= 0,8974·0,7071 + 0,1257·0 + 0·0,7071 + 0,4230·0=0,6346

sim(Doc2,q)

= 0,0756·0,7071 + 0,7867·0 + 0,6127·0,7071 + 0·0= 0,4867

sim(Doc3,q)

= 0,5953·0,7071 + 0·0 + 0,7062·0,7071 + 0,3833·0= 0,9203

Im Ergebnis wird also doc3 höher als doc1 gerankt, das wiederum höher als doc2

Vektorraummodell: Freitextanfragen

Das Vektorraummodell unterstützt Freitextanfragen, in denen eine Anfrage als Menge von Termen ohne Verknüpfungen untereinander repräsentiert ist.

Zu einer gegebenen Anfrage werden Scores für die möglichen Antwortdokumente berechnet, auf deren Grundlage ein Ranking erfolgt.

- Die ursprüngliche klassische Interpretation von Freitextanfragen sieht vor, dass diejenigen Dokumente für die Retrieval-Antwort in Frage kommen, die **mindestens einen** der Anfrageterme enthalten.
- Mit dem Aufkommen von Web-Suchmaschinen trat eine neue Interpretation hinzu: Die Anfrageterme zusammen werden oft als **konjunktive Anfrage** aufgefasst. Für die Retrieval-Antwort kommen dann die Dokumente in Betracht, die alle Anfrageterme enthalten (oder zumindest die meisten).