

Kosinus-Ähnlichkeitsmaß

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

Stehen in der Term-Dokument-Matrix nur positive Zahlen, beträgt der größte mögliche Winkel zwischen zwei Vektoren 90°

Dann ist Wertebereich Kosinus-Maß $[0...1]$

Sonst $[-1...1]$

Termrelevanz

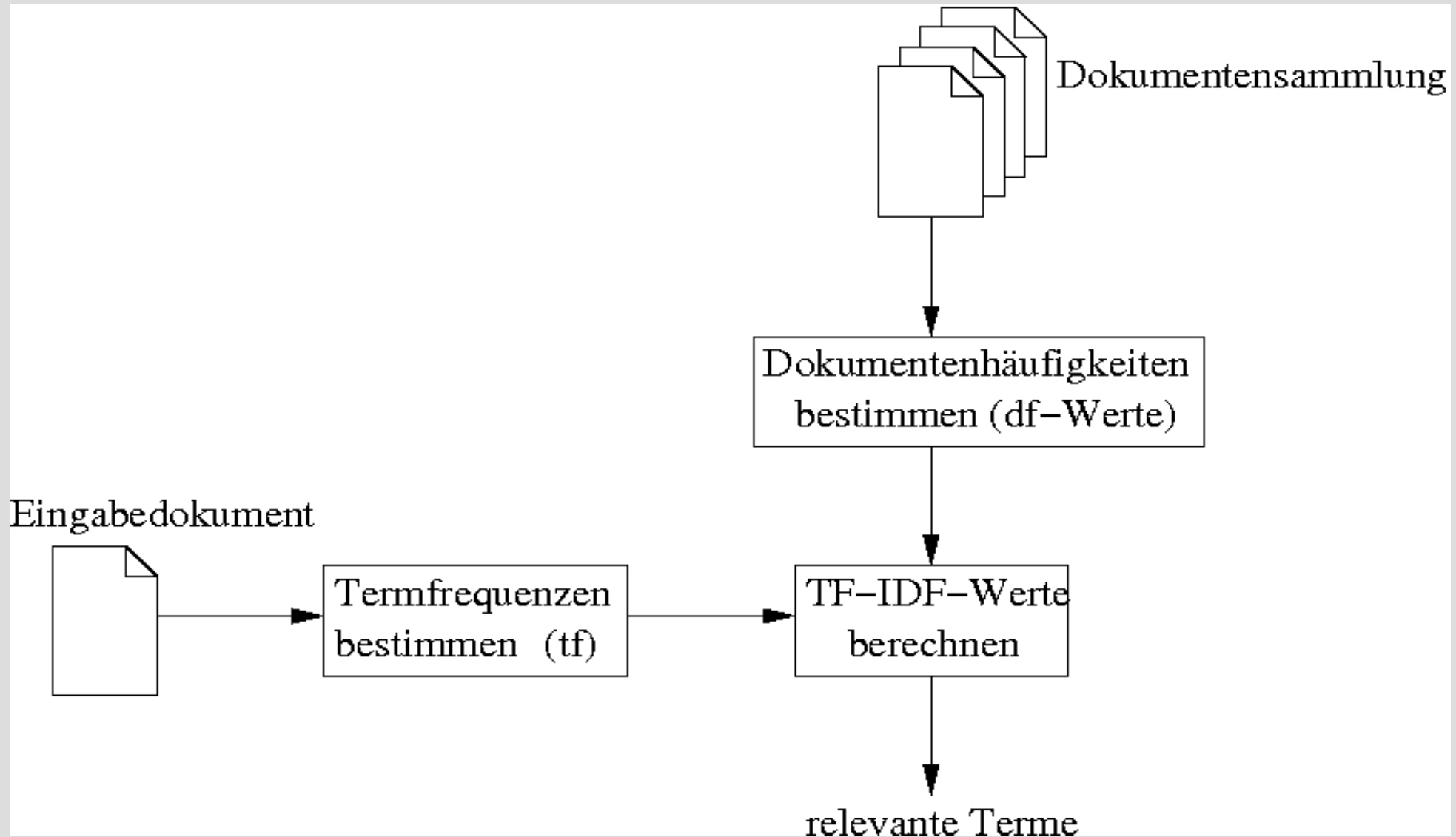
- Beispielttext aus Wikipedia: Man_Ray
- Was sind die relevantesten Terme im Text?

Termrelevanz

- Beispieltext aus Wikipedia: Man_Ray
- Was sind die relevantesten Terme im Text?
- Berechnet mit TF-IDF:

– Ray	29,03	Künstlers	11,51
– Stieglitz	15,60	Johann-Karl	9,12
– Fotografie	14,23	Photogramme	9,12
– Man	12,48	Pseudo-Solarisation	9,12
– .	11,77	Rayographien	9,12
– Cimetière	11,51	Rudnitzky	9,12
– Dada-Sektion	11,51	Sabattier-Effekts	9,12
– Gemälde	11,51	er	8,87
– Gründer	11,51	Stiftung	7,89

Termrelevanz



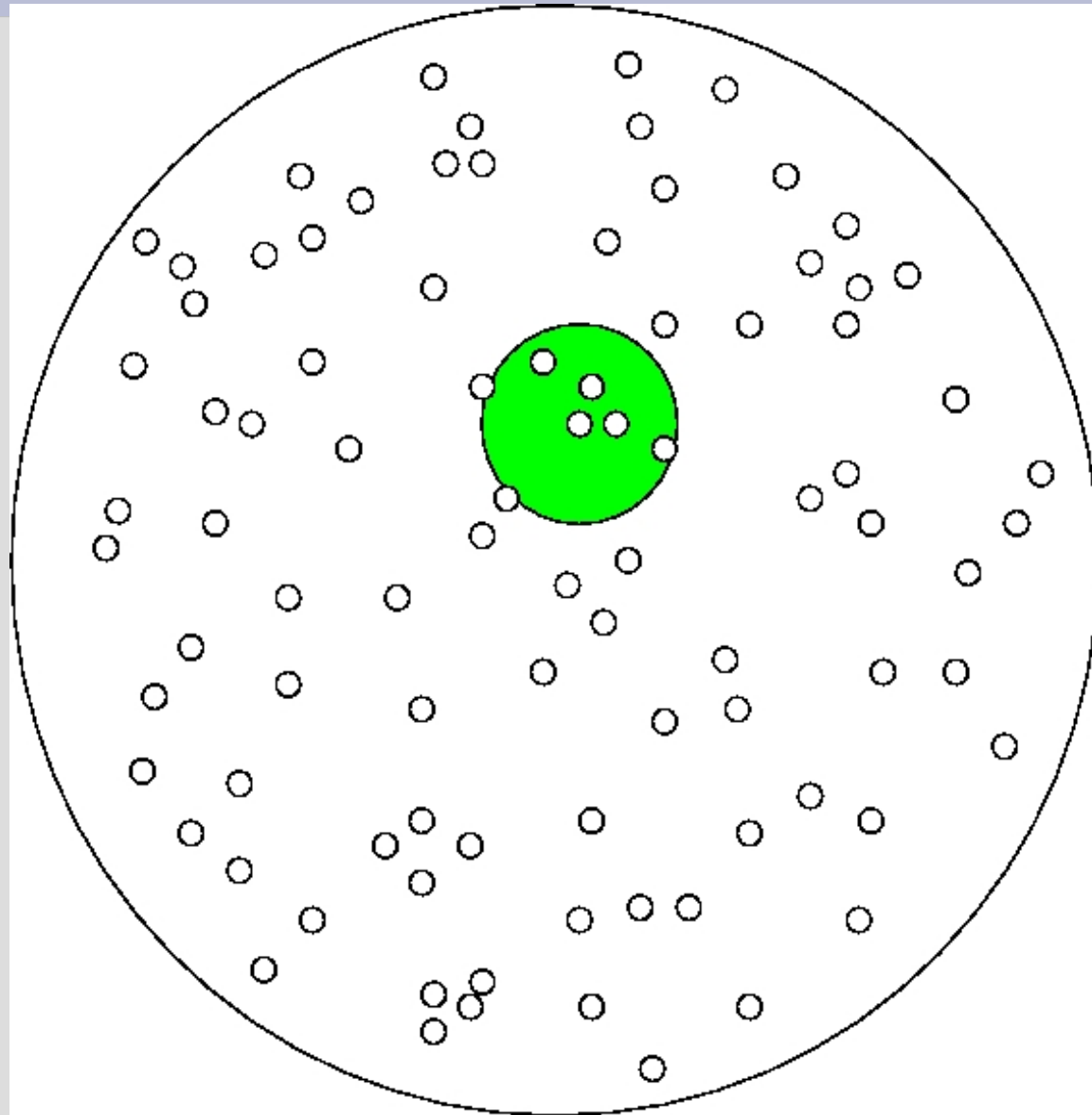
Termrelevanz - Beispielprogramme

- Zwei Perl-Skripte unter <http://www.ling.uni-potsdam.de/~kolb/ir.html>
- *df.pl* und *tfidf.pl*
- df-Speicher *wikipedia10K.df*
- Eingabedokumente sollten tokenisiert sein
- UTF-8 (Unicode)
- Unter Linux: *recode*

Termrelevanz - Anwendungen

- Gewichten von Dokumentvektoren im IR
- Finden ähnlicher Dokumente (“More like this”)
- Erzeugen von Zusammenfassungen:
 - Extraktionsverfahren:
 - Bestimme Termrelevanzen nach TF-IDF
 - Zerlege Eingabetext in Sätze
 - Berechne für jeden Satz die Summe der TF-IDF-Werte der enthaltenen Terme
 - Sortierte Sätze nach summierten TF-IDF-Werten
 - Gebe die n relevantesten Sätze als Zusammenfassung aus
- Automatische Verschlagwortung

Dokumentenraum



Ähnlichste Dokumente finden

	d_1	d_2	\dots	d_m
t_1	$f_{1,1}$	$f_{1,2}$	\dots	$f_{1,m}$
t_2	$f_{2,1}$	$f_{2,2}$	\dots	$f_{2,m}$
\dots	\dots	\dots	\dots	\dots
t_n	$f_{n,1}$	$f_{n,2}$	\dots	$f_{n,m}$

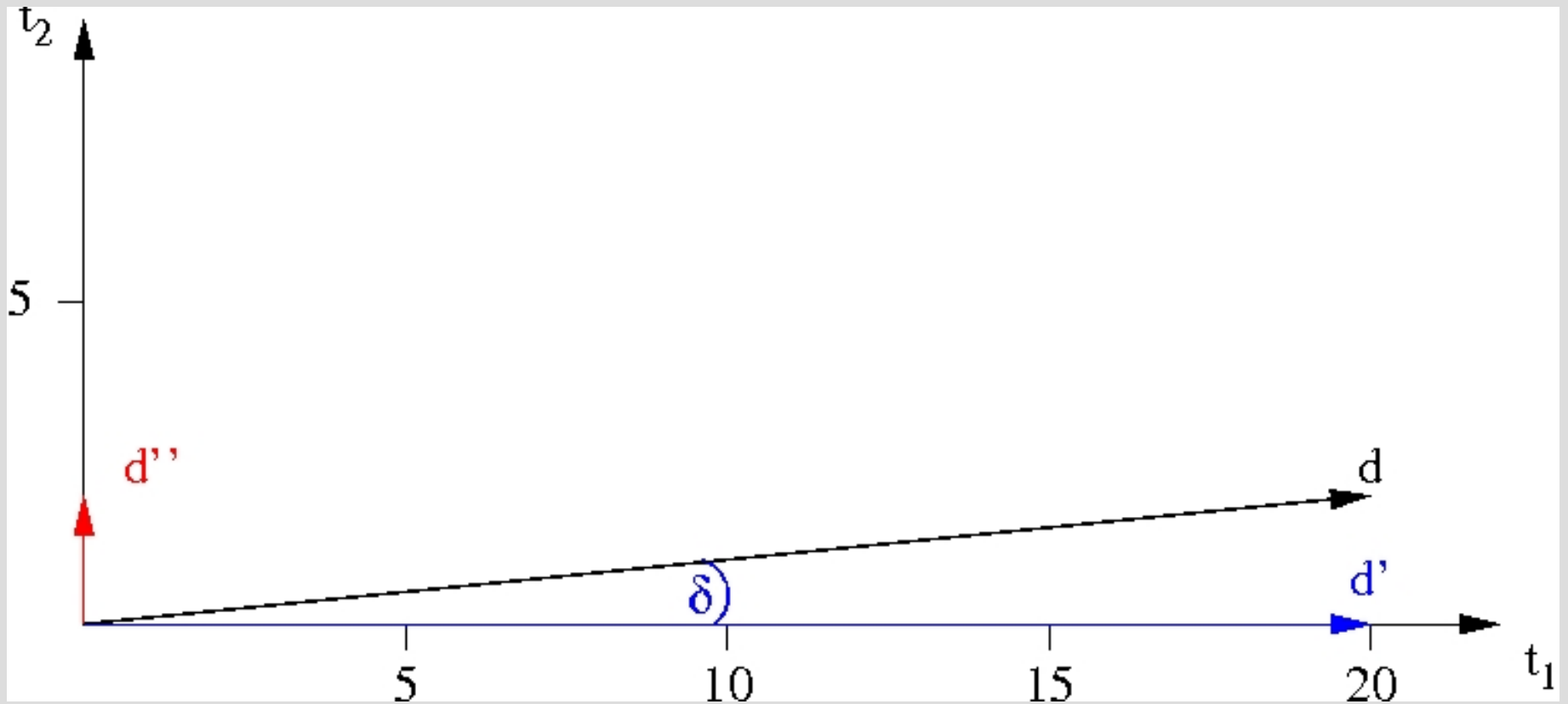
Ähnlichste Dokumente finden

- Dokumentvektor des Ausgangsdokuments mit allen anderen Dokumentvektoren auf Ähnlichkeit vergleichen
- per Kosinus-Maß
- n Dokumente, t Terme: $O(n*t)$
- Rangfolge der ähnlichsten Dokumente
- Duplikate, Versionen und Plagiate finden (Schwellwert?)

Ähnlichste Dokumente im Web finden

- Ausgangsdokument bzw. -vektor
- n relevanteste Terme auswählen
- Anfrage an Suchmaschine konstruieren und abschicken
- Woher den df-Speicher nehmen?
- Dokumentvektor wird stark gekürzt – nicht schlimm (siehe nächste Folie!)
- Nachteil: keine Gewichtung der Terme in der Anfrage möglich
- nicht das Gleiche wie “ähnliche Seiten” bei Google!

Löschen nicht-relevanter Dimensionen



Ähnliche Dokumente in Lucene finden

- n relevanteste Terme auswählen
- Suchanfrage konstruieren
- Terme in Suchanfrage mit Hilfe des “term boost factors” gewichten
- z.B.: Ray^{29.03} AND Stieglitz^{15.6} AND Fotografie^{14.23} AND Man^{12.48}
- 50%-60% der relevantesten Terme am besten
- Ähnlichkeitsfunktion FindSimilar bereits eingebaut

Termähnlichkeit

- Term-Dokument-Matrix: Spalten = Dokumentvektoren
- Zeilen = Termvektoren
- Termvektoren mit Kosinus-Maß auf Ähnlichkeit vergleichen
- → Liste mit ähnlichen Termen zum Ausgangsterm
- automatische Thesaurusgenerierung
- eigene, zweite Adjazenzstruktur nötig
- Manning u. Schütze (1999): Abschnitt 8.5
- <http://bscw.sfb632.uni-potsdam.de/vbsa/disco.html>

Termähnlichkeit

	d_1	d_2	\dots	d_m
t_1	$f_{1,1}$	$f_{1,2}$	\dots	$f_{1,m}$
t_2	$f_{2,1}$	$f_{2,2}$	\dots	$f_{2,m}$
\dots	\dots	\dots	\dots	\dots
t_n	$f_{n,1}$	$f_{n,2}$	\dots	$f_{n,m}$

Termähnlichkeit

	A	B	C	D	E	F	G	H
A	0	1	0	0	1	0	0	1
B	1	0	1	1	0	0	0	0
C	1	1	0	1	0	0	0	0
D	1	1	1	0	1	0	1	0
E	1	1	0	1	0	1	0	0
F	0	0	1	0	1	0	1	0
G	1	0	0	0	0	1	0	0
H	1	0	1	0	0	0	0	0

