



Information Retrieval

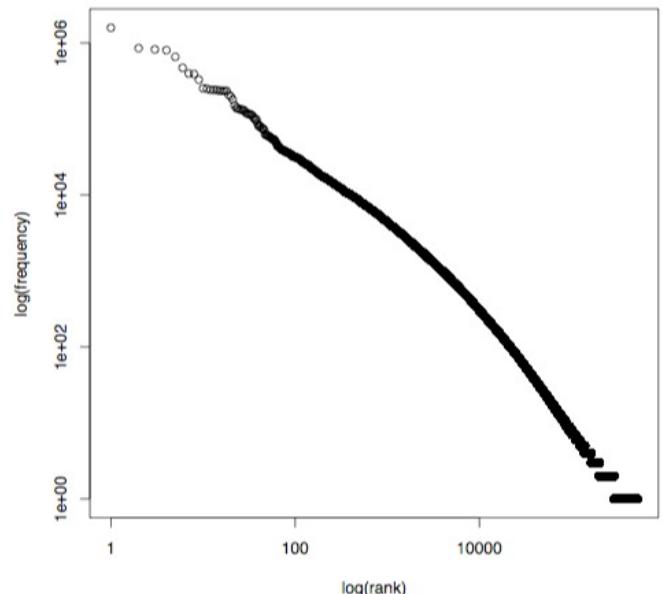
04: Vector Space Model und Ranking

Philipp Schaer, Technische Hochschule Köln, Cologne, Germany

Version: 2021-04-08

Letzte Woche: tf-idf

- Binär → Termfrequenz → tf-idf
- Warum Logarithmus?
- Textstatistik
- Scoring als Grundlage des Ranking
- Zipfs Gesetz
- Von Zipf zu idf
- Erweitertes boolesches Retrieval



Textstatistik



- **Film:** Rocky (1976)
- **Plot:**

Rocky Balboa is a struggling boxer trying to make the big time. Working in a meat factory in Philadelphia for a pittance, he also earns extra cash as a debt collector. When heavyweight champion Apollo Creed visits Philadelphia, his managers want to set up an exhibition match between Creed and a struggling boxer, touting the fight as a chance for a "nobody" to become a "somebody". The match is supposed to be easily won by Creed, but someone forgot to tell Rocky, who sees this as his only shot at the big time. Rocky Balboa is a small-time boxer who lives in an apartment in Philadelphia, Pennsylvania, and his career has so far not gotten off the canvas. Rocky earns a living by collecting debts for a loan shark named Gazzo, but Gazzo doesn't think Rocky has the viciousness it takes to beat up deadbeats. Rocky still boxes every once in a while to keep his boxing skills sharp, and his ex-trainer, Mickey, believes he could've made it to the top if he was willing to work for it. Rocky, goes to a pet store that sells pet supplies, and this is where he meets a young woman named Adrian, who is extremely shy, with no ability to talk to men. Rocky befriends her. Adrian later surprised Rocky with a dog from the pet shop that Rocky had befriended. Adrian's brother Paulie, who works for a meat packing company, is thrilled that someone has become interested in Adrian, and Adrian spends Thanksgiving with Rocky. Later, they go to Rocky's apartment, where Adrian explains that she has never been in a man's apartment before. Rocky sets her mind at ease, and they become lovers. Current world heavyweight boxing champion Apollo Creed comes up with the idea of giving an unknown a shot at the title. Apollo checks out the Philadelphia boxing scene, and chooses Rocky. Fight promoter Jergens gets things in gear, and Rocky starts training with Mickey. After a lot of training, Rocky is ready for the match, and he wants to prove that he can go the distance with Apollo. The 'Italian Stallion', Rocky Balboa, is an aspiring boxer in downtown Philadelphia. His one chance to make a better life for himself is through his boxing and Adrian, a girl who works in the local pet store. Through a publicity stunt, Rocky is set up to fight Apollo Creed, the current heavyweight champion who is already set to win. But Rocky really needs to triumph, against all the odds..

Dokumenten-Repräsentation - tf

Rang	Term	Frequenz
1	a	22
2	rocky	19
3	to	18
4	the	17
5	is	11
6	and	10
7	in	10
8	for	7
9	his	7
..

- tf wird jeweils für das einzelne Dokument bestimmt.
- Mit log, ohne log, ...
- Was sind hier die wichtigen, aussagekräftigen Terme?
- Welches Mittel haben wir, um diese zu finden?

Teststatistik - IDF

Rang	Term	ifd
1	doesn	11,66
2	adrain	10,96
3	viciousness	9,95
4	deadbeats	9,86
5	touting	9,64
6	jergens	9,35
7	gazzo	9,21
8	pittance	7
..
53	rocky	5,09

- ifd wird global für das gesamte Testkorpus erstellt

TF-IDF

Rang	Term	tf-idf
1	rocky	96,72
2	apollo	34,20
3	creed	34,18
4	philadelphia	30,95
5	adrian	26,44
6	balboa	25,83
7	boxing	22,37
8	boxer	22,19
9	heavyweigh	21,54
..

Einfache tf-idf-Berechnung:

- $tf_{\text{rocky}} = 19$
- $idf_{\text{rocky}} = 5,09$
- $tf\text{-}idf = 96,72$

Karikaturen-Analogie



- **tf-idf** hebt Terme hervor, die oft im Dokument vorkommen, aber insgesamt selten sind.
- **Karikaturen** betonen Eigenheiten, die charakteristisch für Personen sind (im Vergleich zum Durchschnitt)

tf, idf oder tf-idf?

adrian all already also an and apartment apollo as aspiring at
balboa become better big boxer boxing but by can career champion
chance creed current debt doesn't earns every exhibition extra far fight for gazzo gets girl
go has he heavyweight her himself his in is it keep later life living loan lovers
make man match meat men mickey named nobody of paulie pet philadelphia
rocky set she shot small somebody someone still store struggling supplies surprised
that **the** they think this through time title to trainer training up want when where
who willing with woman won works

tf, idf oder tf-idf?

ability adrain **adrian** already apartment **apollo** aspiring **balboa** become
befriended befriends big **boxer** boxes **boxing** canvas champion chance checks
chooses collecting collector **creed** current deadbeats debt debts distance doesn't downtown
earns ease easily exhibition extra extremely factory **fight** forgot **gazzo** gear gotten
heavyweight his is jergens later loan lot lovers managers **match** meat mickey named
nobody odds packing paulie pennsylvania **pet philadelphia** pittance promoter
publicity ready **rocky** sells set shark sharp shot shy somebody someone stallion store
struggling stunt supplies supposed surprised thanksgiving think thrilled time title **touting** trainer training
triumph up ve **viciousness** visits where who willing won works

tf, idf oder tf-idf?

ability **adrain** adrian already apollo aspiring **balboa**
beat **befriended** befriends better boxer **boxes** boxing
canvas cash champion checks chooses collecting
collector creed current **deadbeats** debt debts
distance **doesn** downtown earns ease easily
exhibition explains extra extremely factory far forgot
gazzo gear giving gotten **heavyweight** idea interested
italian **jergens** keep living loan lot lovers managers match meat
mickey nobody odds packing paulie pennsylvania pet
philadelphia **pittance** promoter prove publicity
ready rocky sells shark sharp shop shy skills **somebody** spends
stallion struggling **stunt** supplies supposed surprised
thanksgiving think thrilled title **touting** trainer training
triumph unknown ve **viciousness** visits want willing win
won

Erweiterung des Booleschen Retrieval

Mit Hilfe der tf-idf-Gewichte lässt sich ein **Relevance Ranking** für das bisher bekannte Boolesche Retrieval umsetzen:

1. Ergebnismenge mit Hilfe einer booleschen Anfrage erzeugen,
2. dann mit Hilfe der tf-idf-Gewichte pro Anfrageterm den Dokumenten einen Score zuweisen und
3. letztlich nach diesem die Ergebnisliste sortieren/ranken.

- **Dies löst aber nicht die zuvor beschriebenen Probleme**, wie bspw. Feast-or-Famine...
- Wir suchen immer noch die zur Anfrage ähnlichen Dokumente und wollen ein richtiges **Ranked Retrieval, ohne die Probleme des Booleschen Modells!**

Relevanz

Es gibt viele Faktoren, die bestimmen, ob ein Dokument ein Informationsbedürfnis stillt, oder nicht:

- Topicality
 - Novelty, Freshness, Authority
 - Formatting, Reading level, ...
-
- **Topical-Relevanz:** Dokument und Anfrage gehören zum gleichen „Topic“ bzw. Thema
 - **User-Relevanz:** Alle anderen Punkt der oberen Liste

Wir versuchen nun erst einmal Topical-Relevanz zu berechnen!

Topical-Relevanz

- Indem wir uns auf die thematische Relevanz beschränken, ignorieren wir nicht automatisch den Rest.
- Es bedeutet vielmehr, dass wir uns auf ein Kriterium **konzentrieren**, auf Grundlage dessen Nutzer oft Ihre Relevanz-Entscheidung treffen.
- Und ganz ehrlich: Es ist ein wichtiges Kriterium und der erste Schritt in Richtung eines **Best-Match-Retrieval**.
 - War wir bisher z.B. mit dem **booleschen Modell** gemacht haben, ist **Exact-Match-Retrieval!**

Die Ideen hinter dem Vektorraummodell

- **Kernidee 1:** Dokumente sind Vektoren in einem mehrdimensionalen Vektorraum
- **Kernidee 2:** Auch Anfragen sind Vektoren
- **Kernidee 3:** Dokumente werden anhand Ihrer „Nähe“ zum Anfragevektor gerankt

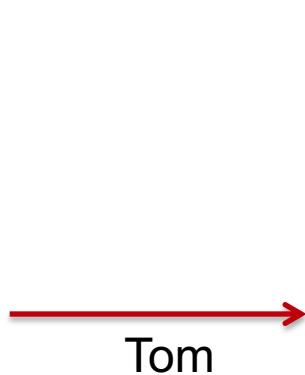
Aber was ist Nähe?

- „Nähe“ = **Ähnlichkeit** der Vektoren
- „Nähe“ = Gegenteil von **Distanz**

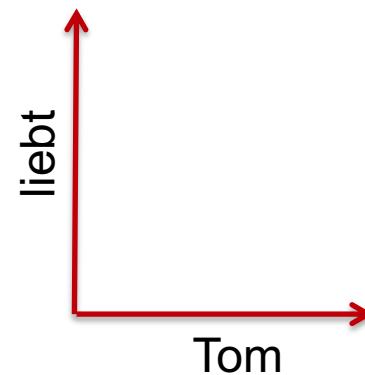
Dokumente als Vektoren

- Jedes Wort ist eine Dimension in einem mehrdimensionalen Vektorraum
- Dokumente sind Vektoren/Punkte in diesem Raum

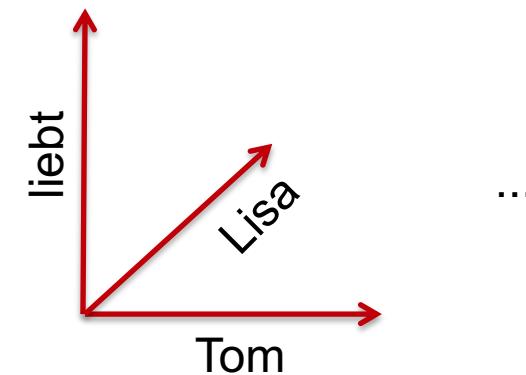
„Tom“



„Tom liebt“



„Tom liebt Lisa“



Aber erst einmal Schritt für Schritt...

Das sollten Sie bereits kennen...

- Die Gesamtheit aller Terme nennt man **Vokabular**.
- Jedes Dokument wird anhand des Vokabulars analysiert und die Worthäufigkeiten werden in einer Tabelle eingetragen („**Term-Dokument-Matrix**“).
- Die Tabelle (oder Matrix) setzt sich aus einzelnen Vektoren (pro Dokument) zusammen, die sich aus dem Vokabular speisen.

Erstellen einer Term-Dokument-Matrix

A



„Ein Hund und ein Huhn.“

B



„Ein Vogel.“

C



„Ein Hund und noch ein Hund.“



Zunächst muss das Vokabular erstellt werden

ein Hund und Huhn Vogel noch

Erstellen einer Term-Dokument-Matrix

A



„Ein Hund und ein Huhn.“

B



„Ein Vogel.“

C



„Ein Hund und noch ein Hund.“

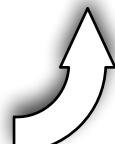


	ein	Hund	und	Huhn	Vogel	noch
A	2	1	1	1	0	0
B	1	0	0	0	1	0
C	2	2	1	0	0	1

Erstellen einer Term-Dokument-Matrix

- A  „Ein Hund und ein Huhn.“ → (2,1,1,1,0,0)
- B  „Ein Vogel.“ → (1,0,0,0,1,0)
- C  „Ein Hund und noch ein Hund.“ → (2,2,1,0,0,1)

	ein	Hund	und	Huhn	Vogel	noch
A	2	1	1	1	0	0
B	1	0	0	0	1	0
C	2	2	1	0	0	1



Anfragen als Vektoren

- Anfragen können auch als Vektoren dargestellt werden

	ein	Hund	und	Huhn	Vogel	noch
A	2	1	1	1	0	0
B	1	0	0	0	1	0
C	2	2	1	0	0	1

„Hund“



Q_{Hund}	0	1	0	0	0	0
-------------------	---	---	---	---	---	---

Anfragen als Vektoren

- Anfragen können auch als Vektoren dargestellt werden

	ein	Hund	und	Huhn	Vogel	noch
A	2	1	1	1	0	0
B	1	0	0	0	1	0
C	2	2	1	0	0	1
Q_{Hund}	0	1	0	0	0	0

„ein Hund“



Q_{einHund}	1	1	0	0	0	0
----------------------	---	---	---	---	---	---

Boolesche Anfragen mit der TD-Matrix

- Boolesche Anfrage (exact match) wären jetzt möglich, wenn man alle Werte auf [0,1] abbildet!

	ein	Hund	und	Huhn	Vogel	noch
A	2	1	1	1	0	0
B	1	0	0	0	1	0
C	2	2	1	0	0	1
Q_{Hund}	0	1	0	0	0	0
$Q_{einHund}$	1	1	0	0	0	0

- Aber wir wollen die **Ähnlichkeit der Dokumente zur Anfrage** für ein Ranking (best match)!

Die Ideen hinter dem Vektorraummodell

- ✓ **Kernidee 1:** Dokumente sind Vektoren in einem mehrdimensionalen Vektorraum
- ✓ **Kernidee 2:** Auch Anfragen sind Vektoren
- **Kernidee 3:** Dokumente werden anhand Ihrer „Nähe“ zum Anfragevektor gerankt

Aber was ist Nähe?

- „Nähe“ = **Ähnlichkeit** der Vektoren
- „Nähe“ = Gegenteil von **Distanz**

Fall 1: Ähnlichkeit der Vektoren

Ein Weg die Ähnlichkeit von Vektoren zu berechnen ist das **Skalarprodukt** (oder innere Produkt) der Vektoren.

$$\sum_{i=1}^{|V|} q_i d_i$$

	q_i	d_i	$q_i \times d_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
:	:	:	:
zoom	0	0	0
Summe:			2

Zur Erinnerung: Vektor-Skalarprodukt

Beim **Skalarprodukt** zweier Vektoren werden alle zusammengehörigen Elemente der Vektoren multipliziert und letztlich aufaddiert

Ein Beispiel: $\sum_{i=1}^{|V|} q_i d_i$

$$q = (1, 0, 0)$$

$$d = (2, 1, 1)$$

$$q \cdot d = (1 * 2) + (0 * 1) + (0 * 1) = 2$$

Skalarprodukt - Einfacher Fall

- Wenn wir nur 0 oder 1 erfassen, dann ist das Skalarprodukt nichts anderes, als die **Anzahl der gemeinsamen Terme zwischen Dokument und Anfrage.**
- Scoring auf Grundlage des Skalarproduktes hat einen großen Nachteil. Welchen?

	q_i	d_i	$q_i \times d_i$
a	1	1	1
aardvark	0	1	0
abacus	1	1	1
abba	1	0	0
able	0	1	0
:	:	:	:
zoom	0	0	0
Summe:			2

Skalarprodukt

Was ist relevanter zu einer Anfrage?

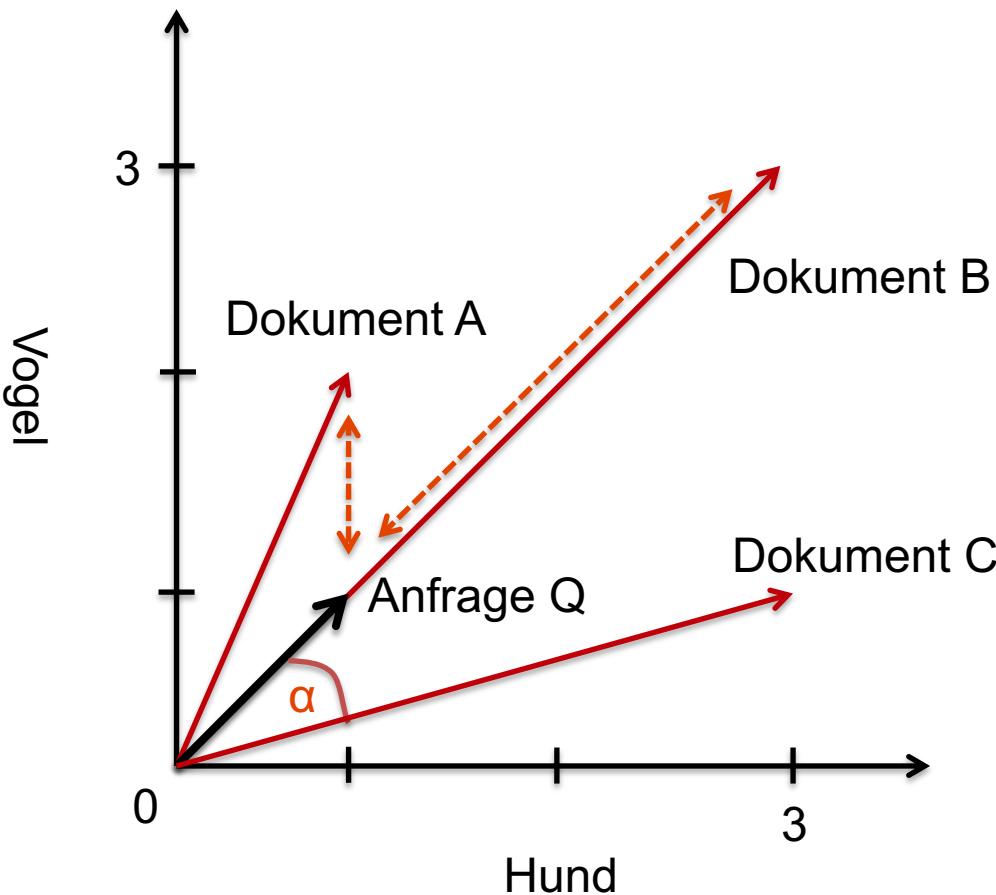
- Ein 50-Worte-Dokument, das 3 der Anfrageterme enthält?
- Ein 100-Worte-Dokument, das 3 der Anfrageterme enthält?

- Solange alle anderen Gegebenheiten gleich sind, haben lange Dokumente eine **höhere Chance Anfrageterme zu enthalten.**
- Das Skalarprodukt berücksichtigt nicht, dass Dokumente **unterschiedlich lang** sein können.
- Also **bevorzugt** das Skalarprodukt **lange Dokumente!**

Fall 2: Nähe bzw. Distanz?

- Aus der Schule kennen Sie noch **die Euklidische Distanz** zwischen zwei Punkten...
- Aber die Euklidische Distanz ist keine gute Lösung, denn die Distanz vergrößert sich bei großen Vektoren...

Euklidische Distanz funktioniert nicht



	Hund	Vogel
A	1	2
B	3	3
C	3	1
Q	1	1

- Euklidische Distanz ist ungeeignet, da abhängig von Länge!
- Der **Winkel** zwischen zwei Vektoren ist aber stabil gegenüber der Länge von Vektoren

Die Ideen hinter dem Vektorraummodell

- ✓ **Kernidee 1:** Dokumente sind Vektoren in einem mehrdimensionalen Vektorraum
- ✓ **Kernidee 2:** Auch Anfragen sind Vektoren
- ✓ **Kernidee 3:** Dokumente werden anhand Ihrer „Nähe“ zum Anfragevektor gerankt

Aber was ist Nähe?

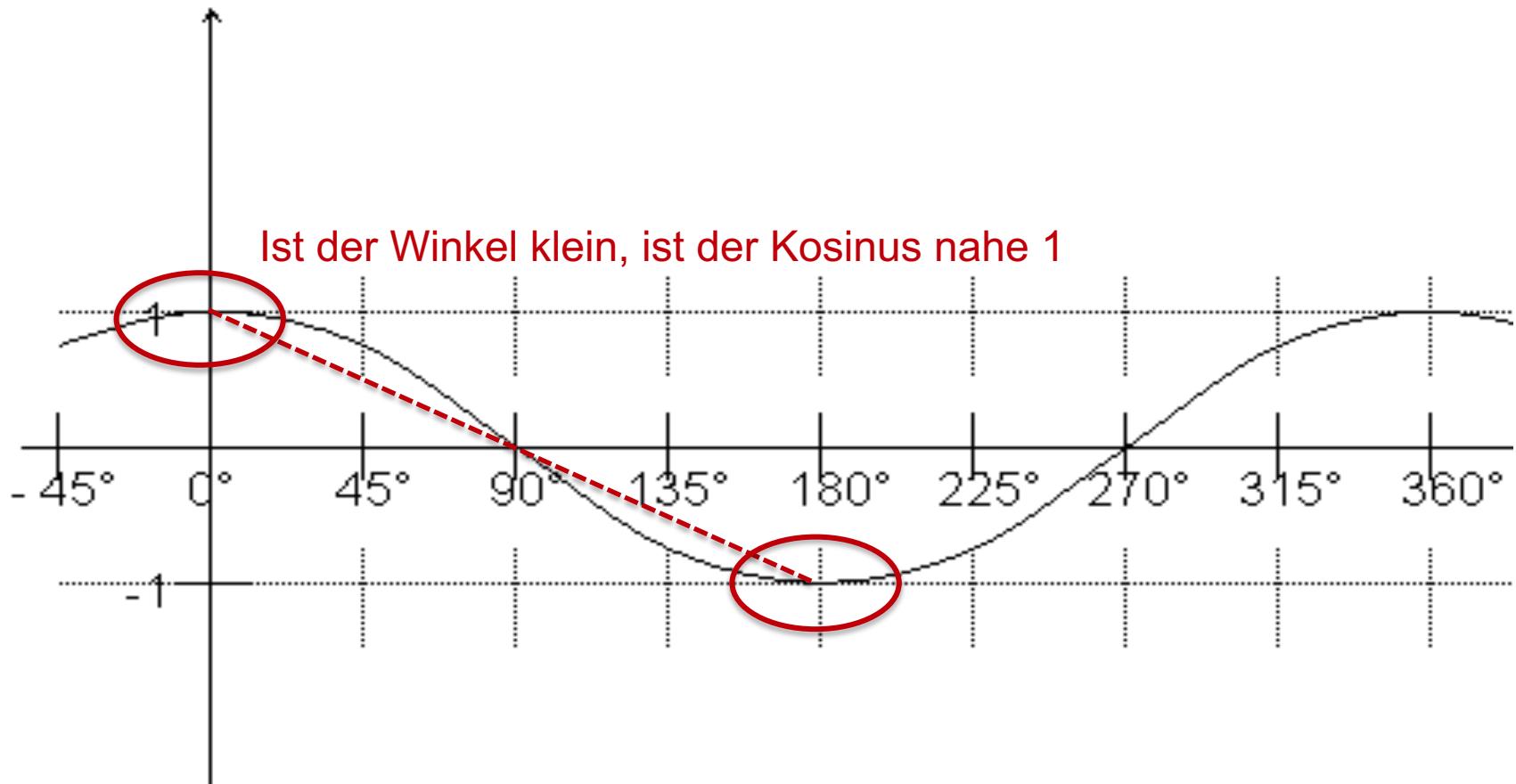
- ✗ „Nähe“ = ~~Ähnlichkeit~~ der Vektoren
- ✗ „Nähe“ = ~~Gegenteil von Distanz~~
- ✓ Nähe = möglichst **kleiner Winkel** zwischen Anfrage- und Dokumentvektor

Von Winkeln zum Kosinus

Die beiden folgenden Aussagen sind äquivalent:

- Man rankt Dokumente in aufsteigender Reihenfolge der Winkelgröße zwischen Anfrage und Dokument
- Man rankt Dokumente in absteigender Reihenfolge von $\cos(\text{Anfrage}, \text{Dokument})$

Von Winkeln zum Kosinus



Warum benutzen wir den Kosinus?

- Der Kosinus ist sehr **schnell und einfach zu berechnen...**

Die Kosinus-Ähnlichkeit

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|}$$

Skalarprodukt Einheitsvektor mit der Länge 1
 Länge der Vektoren

- q_i und d_i sind die **Termfrequenzen** des Terms i in der Anfrage bzw. dem Dokument
- $q \cdot d$ ist das **Vektorprodukt (Skalarprodukt)** von q und d
- $|q|$ bzw. $|d|$ sind die **Länge** der Vektoren

Längennormalisierung

Ein Vektor wird (längen-)normalisiert, wenn jedes seiner Elemente durch seine Länge geteilt wird

$$|\vec{x}| = \sqrt{\sum_i x_i^2} \quad \vec{x}_{norm} = \frac{\vec{x}}{|\vec{x}|}$$

Ein Beispiel:

$$x = (2, 1, 1)$$

$$|x| = \sqrt{2^2 + 1^2 + 1^2} = \sqrt{6} \approx 2,45$$

$$x_{norm} = \left(\frac{2}{2,45}, \frac{1}{2,45}, \frac{1}{2,45} \right)$$

$$x_{norm} = (0,82, 0,41, 0,41)$$

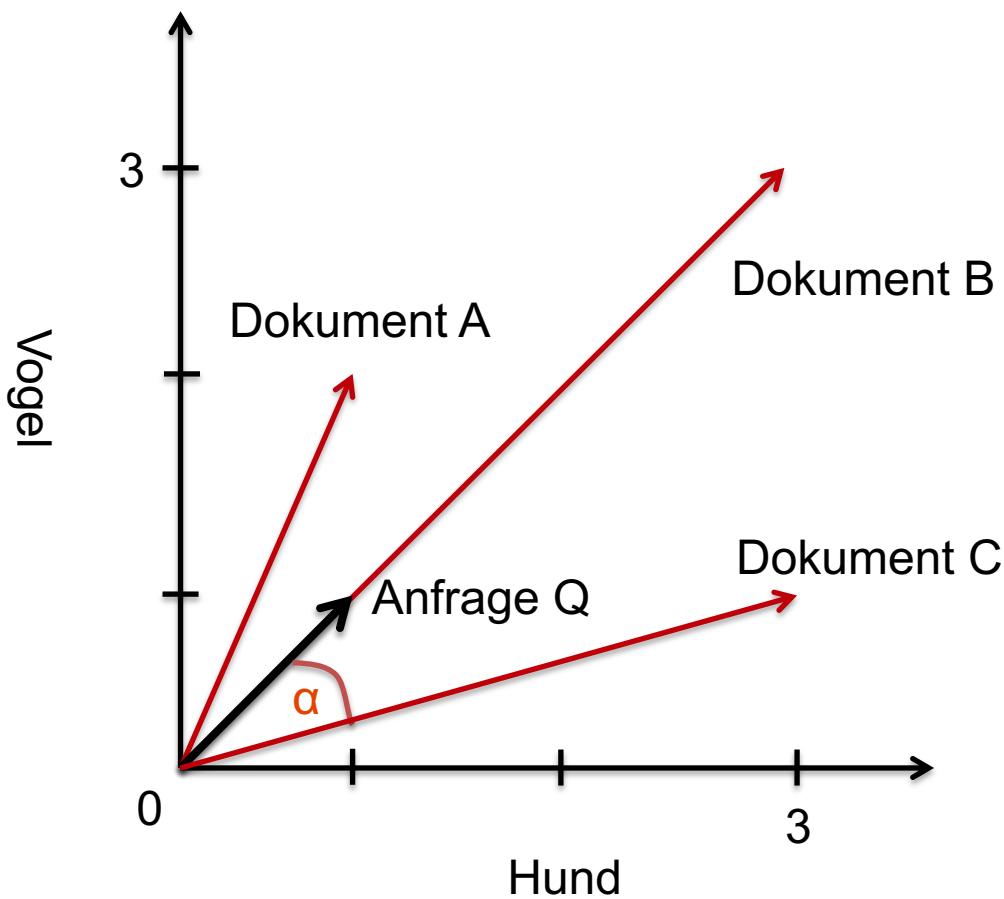
Was bringt uns das?

- Für längennormalisierte Vektoren ist die Kosinus-Ähnlichkeit gleich dem Skalarprodukt:

$$\cos(\vec{q}, \vec{d}) = \vec{q} \cdot \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

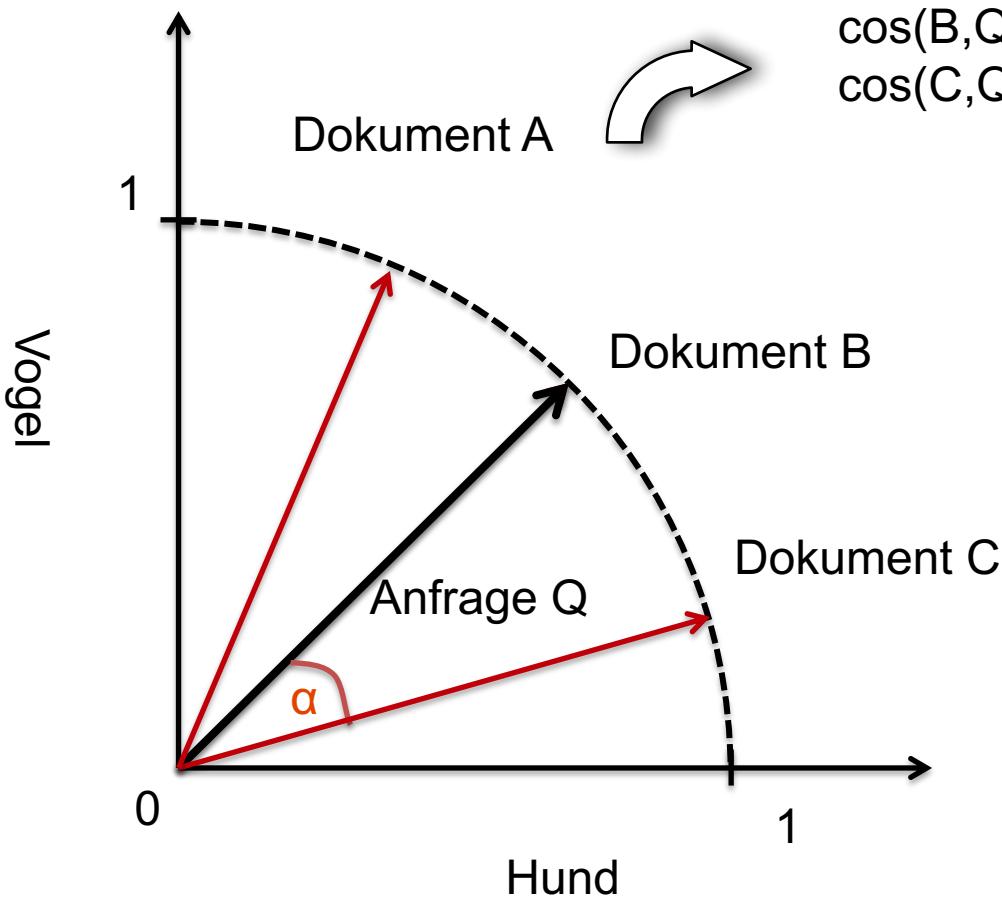
- Normalisieren wir also alle Vektoren **vor unserer Berechnung**, ist die Ähnlichkeit einfach zu berechnen

Kosinus-Ähnlichkeit veranschaulicht



	Hund	Vogel
A	1	2
B	3	3
C	3	1
Q	1	1

Kosinus-Ähnlichkeit veranschaulicht



$$\cos(A, Q) = (0,45 * 0,71) + (0,89 * 0,71) = 0,95$$

$$\cos(B, Q) = (0,71 * 0,71) + (0,71 * 0,71) = 1,00$$

$$\cos(C, Q) = (0,95 * 0,71) + (0,32 * 0,71) = 0,90$$

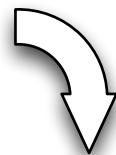
	Hund	Vogel
A	0,45	0,89
B	0,71	0,71
C	0,95	0,32
Q	0,71	0,71



„Hund Vogel“

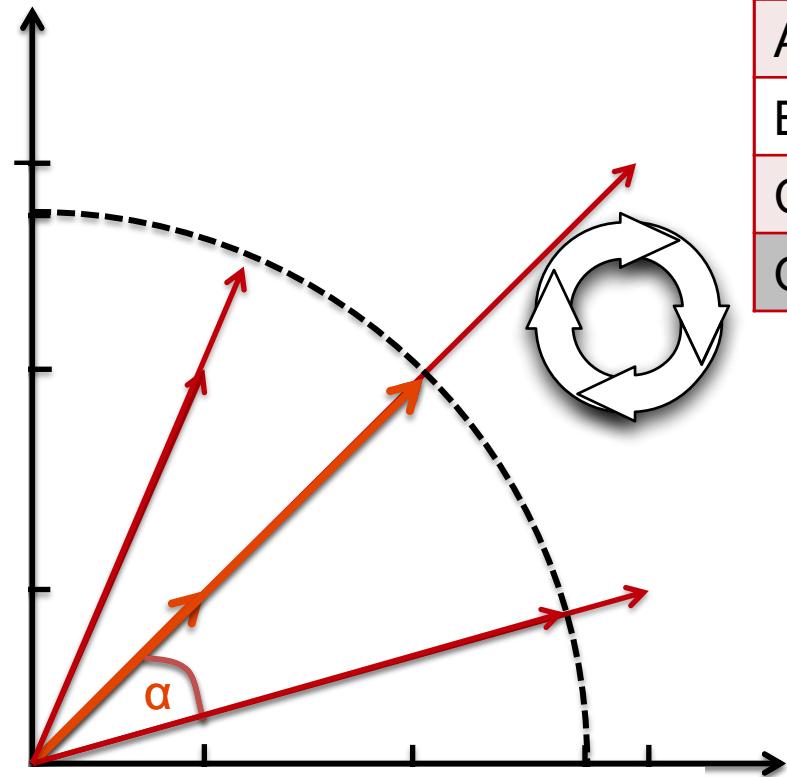


Vektor (1,1)

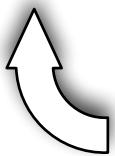


Kosinus

	Hund	Vogel
A	0,45	0,89
B	0,71	0,71
C	0,95	0,32
Q	0,71	0,71



Vektor (3,1)



Zwischenfazit

Wir wissen nun schon eine ganze Menge, u.a.:

- was Ranking nach dem Prinzip des Vektorraummodells ist,
- was Termfrequenzen sind,
- wie man eine Term-Dokument-Matrix baut,
- wie Anfrage-Dokument-Ähnlichkeit mittels Termfrequenzen und Vektor-Vergleiche funktioniert.

We
know you
don't like it but
IT'S
QUIZ TIME!

- *Was hat das nun mit diesem Score zu tun?*
- *Wofür brauchen wir nun nochmal tf-idf?*
- *Wie passt das alles zusammen?*

Zusammenfassung TF-IDF/Vektorraum

- Die Anfrage und jedes Dokument wird als ein **gewichteter tf-idf-Vektor** in einem mehrdimensionalen **Vektorraum** dargestellt.
- Die **Kosinus-Ähnlichkeit** zwischen dem Anfrage-Vektor und jedem Dokumenten-Vektor wird berechnet. Dies ist der Score.
- Die Dokumente werden **gerankt** nach ihrem jeweiligen Score.
- Der Benutzer erhält nun die Top-k (z.B. 10) Dokumente.

Vorteile:

- **Keine Anfragesyntax** (z.B. Boolesche Anfrage) notwendig!
- **Kein Feast or Famine-Problem!**

Multitalent Vektorraummodell

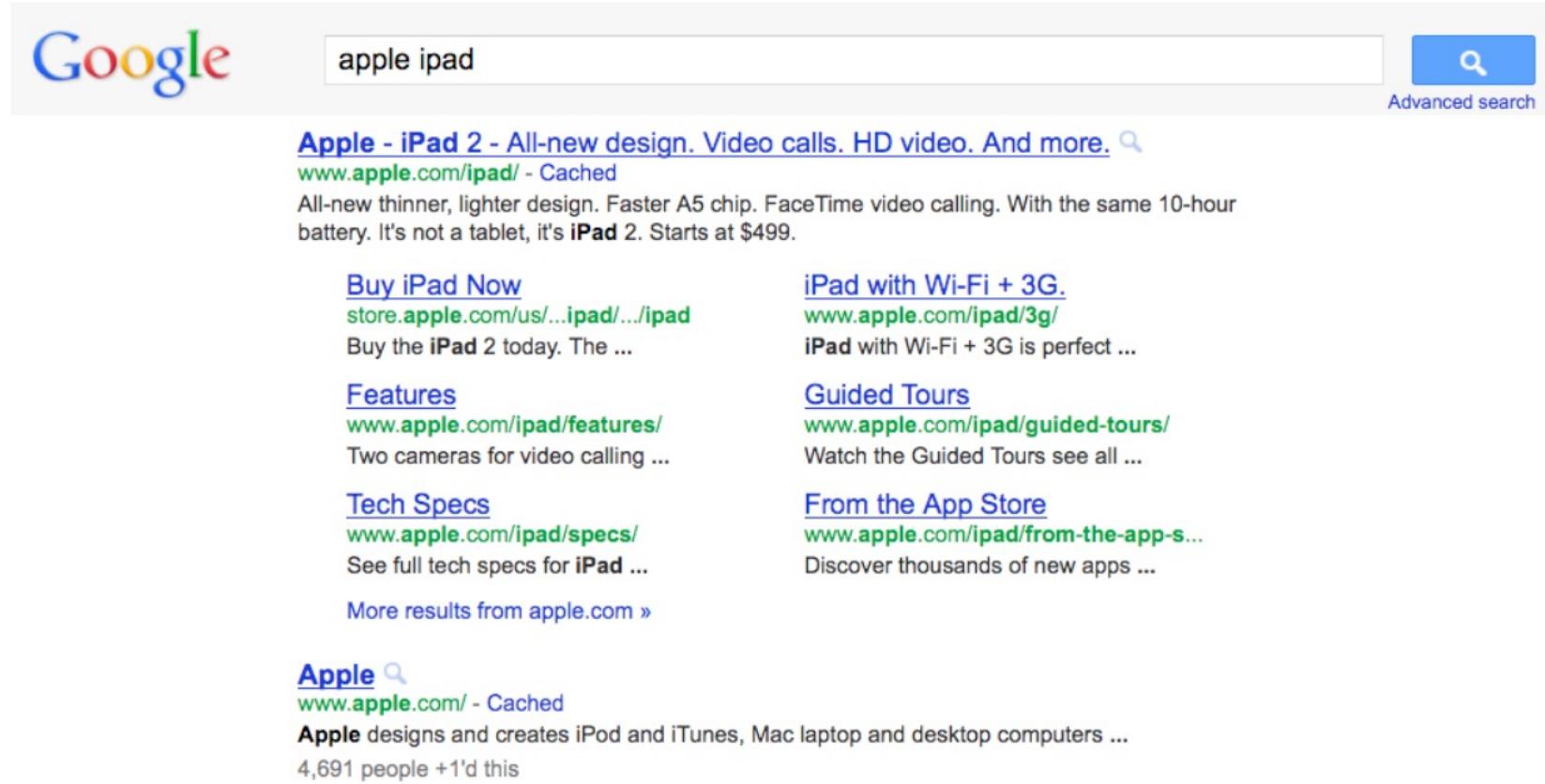
Viele **Probleme** im Information Retrieval lassen sich **umformen** in ein Problem nach folgendem Muster:

- Finde _____, das ähnlich ist zu _____ !

- Solange _____ und _____ **Texte** sind, klappt das.
 - Forme die Texte-Elemente in tf-idf-gewichtete Vektoren um und berechne die Kosinus-Ähnlichkeit.
 - Gebe die Elemente zurück, die die höchste Ähnlichkeit haben.

Multitalent Vektorraummodell

- Finde Dokumente, die ähnlich sind zu dieser Anfrage.



A screenshot of a Google search results page. The search bar at the top contains the query "apple ipad". Below the search bar, there is a link to the Apple website for the iPad 2, followed by a snippet of text describing its features. To the right of the snippet are two columns of links related to the iPad 2, such as "Buy iPad Now", "Features", "Tech Specs", "iPad with Wi-Fi + 3G.", "Guided Tours", and "From the App Store". At the bottom of the search results, there is a link to the official Apple website.

apple ipad

[Apple - iPad 2 - All-new design. Video calls. HD video. And more. !\[\]\(33061038f0134529c71705443c187557_img.jpg\)](#)
[www.apple.com/ipad/](#) - Cached

All-new thinner, lighter design. Faster A5 chip. FaceTime video calling. With the same 10-hour battery. It's not a tablet, it's iPad 2. Starts at \$499.

[Buy iPad Now](#)
[store.apple.com/us/...ipad/.../ipad](#)
Buy the iPad 2 today. The ...

[Features](#)
[www.apple.com/ipad/features/](#)
Two cameras for video calling ...

[Tech Specs](#)
[www.apple.com/ipad/specs/](#)
See full tech specs for iPad ...

[iPad with Wi-Fi + 3G.](#)
[www.apple.com/ipad/3g/](#)
iPad with Wi-Fi + 3G is perfect ...

[Guided Tours](#)
[www.apple.com/ipad/guided-tours/](#)
Watch the Guided Tours see all ...

[From the App Store](#)
[www.apple.com/ipad/from-the-app-s...](#)
Discover thousands of new apps ...

[More results from apple.com »](#)

[Apple !\[\]\(857c2e2c973a85e95f9989bc88a027c9_img.jpg\)](#)
[www.apple.com/](#) - Cached

Apple designs and creates iPod and iTunes, Mac laptop and desktop computers ...
4,691 people +1'd this

Multitalent Vektorraummodell

- Finde **Werbung**, die ähnlich ist zu **diesen Ergebnissen**.

The screenshot shows a Google search results page with the query "apple ipad" in the search bar. The results are organized into several sections:

- Apple - iPad 2 - All-new design. Video calls. HD video. And more.** www.apple.com/ipad/ - Cached

All-new thinner, lighter design. Faster A5 chip. FaceTime video calling. With the same 10-hour battery. It's not a tablet, it's iPad 2. Starts at \$499.

 - Buy iPad Now** store.apple.com/us/...ipad/.../ipad
 - Features** www.apple.com/ipad/features/
 - Tech Specs** www.apple.com/ipad/specs/
 - More results from apple.com »**
- iPad with Wi-Fi + 3G.** www.apple.com/ipad/3g/

iPad with Wi-Fi + 3G is perfect ...
- Guided Tours** www.apple.com/ipad/guided-tours/

Watch the Guided Tours see all ...
- From the App Store** www.apple.com/ipad/from-the-app-s...

Discover thousands of new apps ...
- iPad On Verizon. On Sale.** www.verizonwireless.com/iPad

Magic of iPad. Power of Verizon. Free Shipping With Online Orders 5319 New Hope Commons Ext, Durham
- iPad Apple at Amazon** www.amazon.com/iPad+Apple

amazon.com is rated ★★★★★ Big Savings on iPad apple! Free 2-Day Shipping w/Amazon Prime.
- Apple iPad** www.walmart.com/Ipad

walmart.com is rated ★★★★★ Save On Ipad At Walmart Apple iPad
- Apple** www.apple.com/ - Cached

Apple designs and creates iPod and iTunes, Mac laptop and desktop computers ... 4,691 people +1'd this

Multitalent Vektorraummodell

- Finde Werbung, die ähnlich ist zu diesem Dokument.

Anatidaephobia - The Fear That You are Being Watched by a Duck

December 08, 2008 by Tammy Duffey ▾

Single page Font Size Read comments (44) Share



Popular searches: [YouTube](#) | [Rihanna](#) | [Tiger Woods](#) | [Search more](#)

What Is Anatidaephobia?

Anatidaephobia is defined as a pervasive, irrational fear that one is being watched by a duck. The anatidaephobic individual fears that no matter where they are or what they are doing, a duck watches.

Anatidaephobia is derived from the Greek word "anatidae", meaning ducks, geese or swans and "phobos" meaning fear.

Aflac can help attract and retain employees, at no direct cost to your company.

Aflac
We've got you under our wing.™

Learn More Now

The advertisement features a large white Aflac duck with a yellow beak and orange feet. The background is blue. The text is white and clearly legible.

What Causes Anatidaephobia?

As with all phobias, the person coping with Anatidaephobia has experienced a real-life trauma. For the anatidaephobic individual, this trauma most likely occurred during childhood.

Perhaps the individual was intensely frightened by some species of water fowl. Geese and swans are relatively well known for their aggressive tendencies and perhaps the anatidaephobic person was actually bitten or flapped at. Of course, the Far Side comics did little to minimize the fear of being watched by a duck.

Multitalent Vektorraummodell

- Finde Anfragen, die ähnlich ist zu dieser Anfrage.

A screenshot of a Google search results page. The search bar at the top contains the query "apple ipad". To the right of the search bar is a blue search button with a white magnifying glass icon. Below the search bar, the text "Advanced search" is visible. Underneath the search bar, the heading "Searches related to apple ipad" is displayed. A grid of search terms follows:

Search Term	Search Term
ipad rumor	apple rumors
apple competition kindle	apple ipad pictures
Borders apple tablet	apple iphone
apple ipad review	apple itouch

Kosinus-Ähnlichkeit für 3 Dokumente

Wie ähnlich sind sich die drei folgenden **Parteiprogramme**?

- Sozialdemokratische Partei Deutschlands (**SPD**),
- Christlich Demokratische Union (**CDU**) und
- Alternative für Deutschland (**AFD**).

Term	SPD	CDU	AFD
Arbeit	115	58	20
Familie	10	7	11
Migration	2	0	6
Islam	0	0	38

Wie vereinfachen das Beispiel, indem wir kein idf-Gewichte verwenden, sondern nur $1 + \log(tf_{t,d})$.

Beispielrechnung (Fortsetzung)

Log-Häufigkeits-Gewichtung

Term	SPD	CDU	AFD
Familie	3,06	2,76	2,30
Arbeit	2,00	1,85	2,04
Migration	1,30	0	1,78
Islam	0	0	2,58

Nach der Normalisierung

Term	SPD	CDU	AFD
Familie	0,78	0,83	0,52
Arbeit	0,51	0,55	0,46
Migration	0,33	0	0,40
Islam	0	0	0,58

$$\cos(\text{SPD}, \text{CDU}) \approx 0,78 * 0,83 + 0,51 * 0,55 + 0,33 * 0,0 + 0,0 * 0,0 \approx 0,94$$

$$\cos(\text{CDU}, \text{AFD}) \approx 0,79$$

$$\cos(\text{SPD}, \text{AFD}) \approx 0,69$$

Warum ist
 $\cos(\text{SPD}, \text{CDU}) > \cos(\text{SPD}, \text{AFD})$?

Zusätzliche Materialien

- Eine Beispielrechnung:
<http://www.youtube.com/watch?v=HW9W6EBytLg>
- Manning et al.: Introduction to Information Retrieval,
Kapitel 6.2 – 6.4.3

Es gibt noch viel mehr!

- Bisher haben wir nur Termhäufigkeiten ausgezählt und die Ähnlichkeit von Dokumenten bzw. Anfragen mit Hilfe des **Kosinus-Abstandes** auf Grundlage des Vektorraummodells berechnet.
- Aber es gibt noch viel mehr
 - **Probabilistisches Modell**
 - Statistische Sprachmodelle (**Language Models**)
 - uvm.