# INFO221v12
# IR IV

Truls Pedersen
*Institutt for informasjons- og medievitenskap*
Universitetet i Bergen

# Resten av kurset - forslaget vedtatt

| | Uke 15 | Uke 16 | Uke 17 | Uke 18 |
|---|---|---|---|---|
| Plan | Forel.<br>KO3(ut) | Pres (KO3) | Forel.<br>KO3(inn) | Forel.(resten) |
| Forslag | Forel.<br>KO3(ut) | Forel. | Forel. | Forel.(åpent)<br>KO3(inn) |

# Oversikt

- Parametriske indekser
- Vektet sonepoeng
- Læring av vekter
- Termfrekvens o.l.
- Vektorrepresentasjon
- Alternative poengligninger
- SMART
- CosineScore

# Parametriske indekser

Vi har sett forskjellige måter å indeksere termene i et dokument; enkel forekomst, frekvens, posisjoner.

Vi kan også inkludere et *relevansestimat*.

# Parametriske indekser

Vi har sett forskjellige måter å indeksere termene i et dokument; enkel forekomst, frekvens, posisjoner.

Vi kan også inkludere et *relevansestimat*.

Vi kan søke etter dokumenter som har “star wars” i tittelen:

```
intitle:  ``star wars''
```

# Parametriske indekser

Vi har sett forskjellige måter å indeksere termene i et dokument; enkel forekomst, frekvens, posisjoner.

Vi kan også inkludere et *relevansestimat*.

Vi kan søke etter dokumenter som har “star wars” i tittelen:

```
intitle:  ``star wars''
```

En *sone* er en konkret del av et dokument:
tittel, ingress, sammendrag, ...

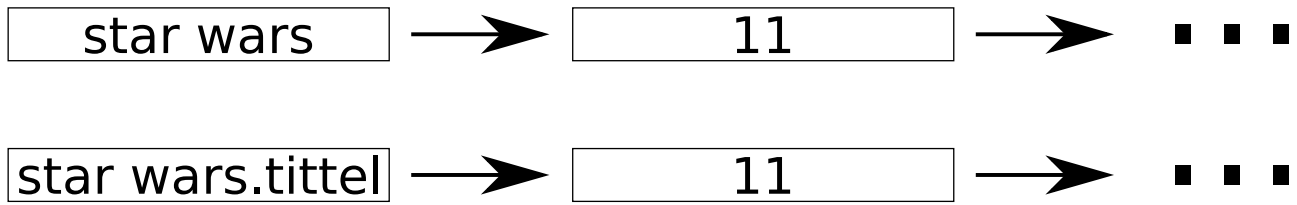
Et *felt* er en attributt fra metadata: forfatter, publiseringsdato, ...

# Parametriske indekser - Representasjon


Representasjons former:

1. Som før: “star wars” forekommer i dokument 11,

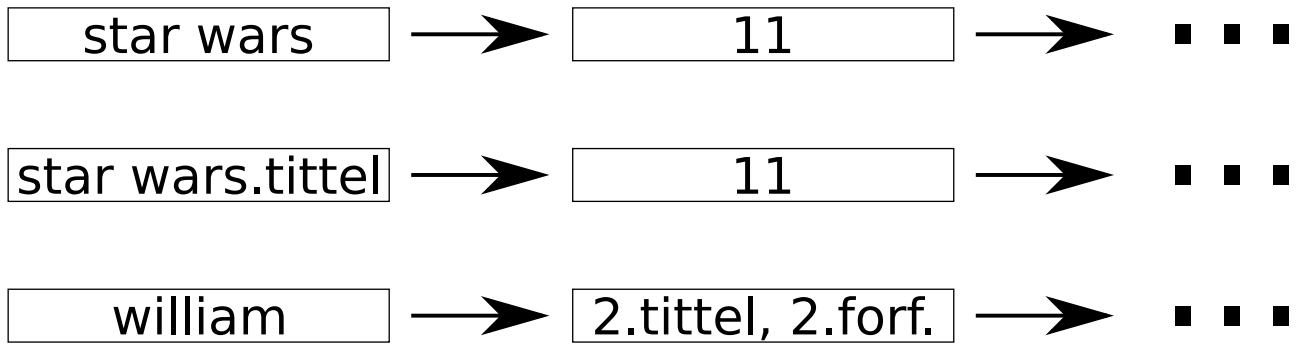
# Parametriske indekser - Representasjon




Representasjons former:

1. Som før: “star wars” forekommer i dokument 11,
2. “star wars” forekommer i tittelen til dokument 11, og

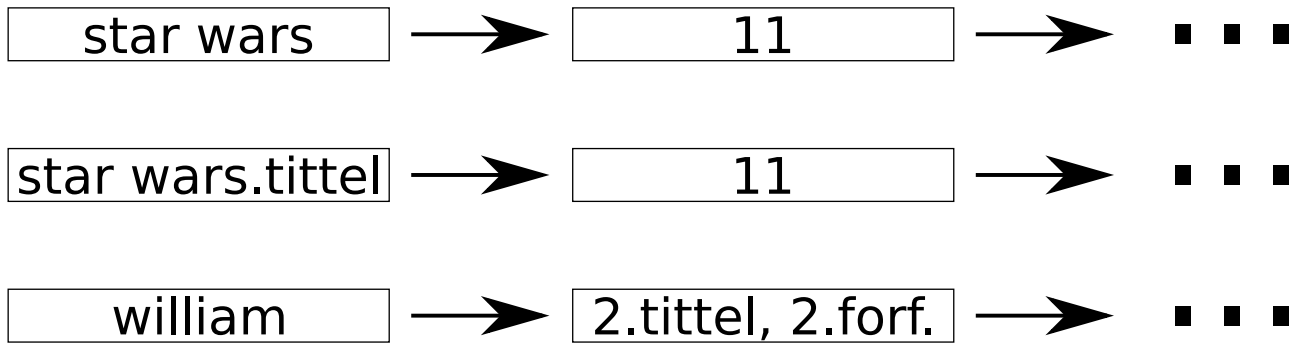
# Parametriske indekser - Representasjon




Representasjons former:

1. Som før: “star wars” forekommer i dokument 11,
2. “star wars” forekommer i tittelen til dokument 11, og
3. “william” forekommer i tittelen og som forfatter til dok. 2.

# Parametriske indekser - Representasjon




Representasjons former:

1. Som før: “star wars” forekommer i dokument 11,
2. “star wars” forekommer i tittelen til dokument 11, og
3. “william” forekommer i tittelen og som forfatter til dok. 2.

Vi kan nå søke i spesifikke soner, men dette tillater også *vektet sonepoeng*.

# Weighted zone scoring (vektet sonepoeng)

Hvis en søketerm forekommer i tittelen til dokument $d_1$, og i sammendraget til dokument $d_2$, hvilket dokument er da mest relevant?

# Weighted zone scoring (vektet sonepoeng)

Hvis en søketerm forekommer i tittelen til dokument $d_1$, og i sammendraget til dokument $d_2$, hvilket dokument er da mest relevant?

Anta at alle dokumentene våre har nøyaktig tre soner:

$$(s_1, s_2, s_3) = \text{(tittel, sammendrag, kropp)}$$

Vi lar hver $s_i$ være 1 hvis søketermen forekommer i sone $i$ og 0 ellers. Hvert dokument får fra 0 til 3 poeng.

# Weighted zone scoring (vektet sonepoeng)

Hvis en søketerm forekommer i tittelen til dokument $d_1$, og i sammendraget til dokument $d_2$, hvilket dokument er da mest relevant?

Anta at alle dokumentene våre har nøyaktig tre soner:

$$(s_1, s_2, s_3) = \text{(tittel, sammendrag, kropp)}$$

Vi lar hver $s_i$ være 1 hvis søketermen forekommer i sone $i$ og 0 ellers. Hvert dokument får fra 0 til 3 poeng.

Vi ønsker ikke at et dokument skal få like mange poeng for å ha et treff i sammendraget som i tittelen.

# Weighted zone scoring

Vi kan la et treff telle mer i tittelen enn elles ved å vekte poengene avhengig av hvilken sone treffet forekom i.

La $(g_1, g_2, g_3)$ være vekter slik at $g_1 + g_2 + g_3 = 1$.

Hvert dokument får en poengsum fra 0 til 1 ($l$ soner):

$$\sum_{i=1}^{l} g_i s_i$$

# Weighted zone scoring

Vi kan la et treff telle mer i tittelen enn elles ved å vekte poengene avhengig av hvilken sone treffet forekom i.

La $(g_1, g_2, g_3)$ være vekter slik at $g_1 + g_2 + g_3 = 1$.

Hvert dokument får en poengsum fra 0 til 1 ($l$ soner):

$$\sum_{i=1}^{l} g_i s_i$$

Vi kan la tittelen være viktigere enn sammendraget ved å f.eks. gi vektene

$$(g_1, g_2, g_3) = (0.4, 0.35, 0.25)$$

Da vil $d_1$ og $d_2$ få hhv. 0.4 og 0.35 poeng.

# Weighted zone scoring

*Hvor viktig* er det at en søketerm forekommer i tittelen i forhold til i sammendraget?

Hva med en søketerm som forekommer bare i tittelen i et dokument, og både (og bare) i sammendraget og kroppen til et annet?

Er 0.4 den *riktige* verdien for $g_1$?

# Læring av vekter

Det er generelt umulig å sette vektene $g_i$ for hånd. Vi må la datamaskinen *lære* hva som er "riktig".

1. Konstruer en mengde *treningsdata*:
   - en mengde dokumenter,
   - en liste med søketermer (spørringer), og
   - en *menneskelig* vurdering av hvor relevant hvert dokument er for hver spørring

# Læring av vekter

Det er generelt umulig å sette vektene $g_i$ for hånd. Vi må la datamaskinen *lære* hva som er "riktig".

1. Konstruer en mengde *treningsdata*:
   - en mengde dokumenter,
   - en liste med søketermer (spørringer), og
   - en *menneskelig* vurdering av hvor relevant hvert dokument er for hver spørring
2. La systemet *lære*:
   - still inn vektene i $g$ slik at systemet dømmer dokumentene like relevant som tilhørende vurdering

# Læring av vekter - eksempel

| Eksempel | docID | spørring | $s_T$ | $s_B$ | Vurdering | $r$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | 37 | `linux` | 1 | 1 | Relevant | 1 |
| $\phi_2$ | 37 | `penguin` | 0 | 1 | Irrelevant | 0 |
| $\phi_3$ | 238 | `system` | 0 | 1 | Relevant | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $\phi_7$ | 3191 | `driver` | 1 | 0 | Irrelevant | 0 |

# Læring av vekter - eksempel

| Eksempel | docID | spørring | $s_T$ | $s_B$ | Vurdering | $r$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | 37 | `linux` | 1 | 1 | Relevant | 1 |
| $\phi_2$ | 37 | `penguin` | 0 | 1 | Irrelevant | 0 |
| $\phi_3$ | 238 | `system` | 0 | 1 | Relevant | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $\phi_7$ | 3191 | `driver` | 1 | 0 | Irrelevant | 0 |

Hvert (dokument,spørring)-par får poeng (som vi så):

$$score(d,q) = g_T s_T(d,q) + g_B s_B(d,q) = \sum_{i \in \{T,B\}} g_i s_i(d,q)$$

# Læring av vekter - eksempel

| Eksempel | docID | spørring | $s_T$ | $s_B$ | Vurdering | $r$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | 37 | `linux` | 1 | 1 | Relevant | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $\phi_7$ | 3191 | `driver` | 1 | 0 | Irrelevant | 0 |

Hvert (dokument,spørring)-par får poeng (som vi så):

$$score(d,q) = g_T s_T(d,q) + g_B s_B(d,q) = \sum_{i \in \{T,B\}} g_i s_i(d,q)$$

Hvert (vekt-vektor,test $j$)-par får en *feil* (definert):

$$\varepsilon(g,\phi_j) = (r(d_j,q_j) - score(d_j,q_j))^2$$

# Læring av vekter - eksempel

| Eksempel | docID | spørring | $s_T$ | $s_B$ | Vurdering | $r$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | 37 | `linux` | 1 | 1 | Relevant | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $\phi_7$ | 3191 | `driver` | 1 | 0 | Irrelevant | 0 |

Hvert (dokument,spørring)-par får poeng (som vi så):

$$score(d,q) = g_T s_T(d,q) + g_B s_B(d,q) = \sum_{i \in \{T,B\}} g_i s_i(d,q)$$

Hvert (vekt-vektor,test $j$)-par får en *feil* (definert):

$$\varepsilon(g,\phi_j) = (r(d_j,q_j) - score(d_j,q_j))^2$$

Feilen for en vekt-vektor for testmengden blir da

$$\sum_j \varepsilon(g,\phi_j).$$

# Læring av vekter - eksempel

| Eksempel | docID | spørring | $s_T$ | $s_B$ | Vurdering | $r$ |
|---|---|---|---|---|---|---|
| $\phi_1$ | 37 | `linux` | 1 | 1 | Relevant | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $\phi_7$ | 3191 | `driver` | 1 | 0 | Irrelevant | 0 |

Å la datamaskinen minimere feilen for testmengden er å finne den vekt-vektoren $g$ som minimerer $\sum_j \varepsilon(g, \phi_j)$.

Her har vi bare to soner, så $g = (x, 1 - x)$; én variabel.
Dette klarer en datamskin å finne en optimal løsning for.

Generelt er det vanskelig for en datamaskin å finne en optimal løsning for realistisk store mengder.

# Termfrekvens

Andre tall vi kan bruke for å estimere relevans er poeng basert på *termfrekvens*.

$$tf_{t,d} = \text{antall forekomster av } t \text{ i } d$$

# Termfrekvens

Andre tall vi kan bruke for å estimere relevans er poeng basert på *termfrekvens*.

$$tf_{t,d} = \text{antall forekomster av } t \text{ i } d$$

*Samlingsfrekvens* (*collection frequency*)

$$cf_t = \text{antall forekomster av } t$$

# Termfrekvens

Andre tall vi kan bruke for å estimere relevans er poeng basert på *termfrekvens*.

$$tf_{t,d} = \text{antall forekomster av } t \text{ i } d$$

*Samlingsfrekvens* (*collection frequency*)

$$cf_t = \text{antall forekomster av } t$$

*Dokumentfrekvens*

$$df_t = \text{antall dokumenter } t \text{ forekommer i}$$

# Termfrekvens

Andre tall vi kan bruke for å estimere relevans er poeng basert på *termfrekvens*.

$$tf_{t,d} = \text{antall forekomster av } t \text{ i } d$$

*Samlingsfrekvens* (*collection frequency*)

$$cf_t = \text{antall forekomster av } t$$

*Dokumentfrekvens*

$$df_t = \text{antall dokumenter } t \text{ forekommer i}$$

... og *invertert dokumentfrekvens*

$$idf_t = \log\left(\frac{N}{df_t}\right)$$

# Termfrekvens

Vi har sett

$$\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t$$

$tf_{t,d}$ er *høy* hvis $t$ forekommer ofte i $d$, og $idf_t$ er *høy* dersom $t$ forekommer i få dokumenter.

Vi kan få et inntrykk av hvordan *tf-idf* oppfører seg. $tf\text{-}idf_{t,d}$ er

1. høyest når $t$ forekommer mange ganger i $d$, men ellers i få dokumenter
2. lavere når $t$ forekommer færre ganger i $d$, eller i mange dokumenter, og
3. lavest når $t$ forekommer få ganger i $d$ og også i mange andre dokumenter.

# Termfrekvens

En spørring $q$ består av $l$ termer, ``$t_1 \quad t_2 \quad \ldots \quad t_l$''.

Vi kan angi hvor relevant et dokument $d$ er for spørringen $q$ ved å regne ut summen av tf-idf poeng.

$$\text{poeng}(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$

# Vektorrepresentasjon

Et dokument har en *vektorrepresentasjon* gitt ved

$$\vec{v}(d_i) = (\text{tf-idf}_{t_1,d_i}, \text{tf-idf}_{t_2,d_i}, \ldots, \text{tf-idf}_{t_k,d_i})$$

hvor $t_1, t_2, \ldots, t_k$ er *alle* termer i dokumentsamlingen.

# Vektorrepresentasjon

Et dokument har en *vektorrepresentasjon* gitt ved

$$\vec{v}(d_i) = (\text{tf-idf}_{t_1,d_i}, \text{tf-idf}_{t_2,d_i}, \ldots, \text{tf-idf}_{t_k,d_i})$$

hvor $t_1, t_2, \ldots, t_k$ er *alle* termer i dokumentsamlingen.

Hvis vi nå regner ut vektorene for alle dokumentene får vi et $k$-dimensjonalt vektorrom hvori vi har en vektor for hvert dokument.

# Vektorrepresentasjon

Et dokument har en *vektorrepresentasjon* gitt ved

$$\vec{v}(d_i) = (\text{tf-idf}_{t_1,d_i}, \text{tf-idf}_{t_2,d_i}, \ldots, \text{tf-idf}_{t_k,d_i})$$

hvor $t_1, t_2, \ldots, t_k$ er *alle* termer i dokumentsamlingen.

Hvis vi nå regner ut vektorene for alle dokumentene får vi et $k$-dimensjonalt vektorrom hvori vi har en vektor for hvert dokument.

Vi kan også se hvor "*nær*" to dokumenter er hverandre ved å se på forskjellen mellom vektorene.

# Vektorrepresentasjon

Et dokument har en *vektorrepresentasjon* gitt ved

$$\vec{v}(d_i) = (\text{tf-idf}_{t_1,d_i}, \text{tf-idf}_{t_2,d_i}, \ldots, \text{tf-idf}_{t_k,d_i})$$

hvor $t_1, t_2, \ldots, t_k$ er *alle* termer i dokumentsamlingen.

Hvis vi nå regner ut vektorene for alle dokumentene får vi et $k$-dimensjonalt vektorrom hvori vi har en vektor for hvert dokument.

Vi kan også se hvor "*nær*" to dokumenter er hverandre ved å se på forskjellen mellom vektorene.

En spørring kan vi også se på som et dokument; vi gir poeng til termene i spørringen som for andre dokumenter
(men ser bort fra termer vi ikke har indeksert).

# Vektorer

For to vektorer $\vec{x}$ og $\vec{y}$ har vi et *indreprodukt*

$$\vec{x} \cdot \vec{y} = \sum_i x_i y_i$$

# Vektorer

For to vektorer $\vec{x}$ og $\vec{y}$ har vi et *indreprodukt*

$$\vec{x} \cdot \vec{y} = \sum_i x_i y_i$$

Vi kan *normalisere* en vektor $\vec{x}/|\vec{x}|$.

Alle normaliserte vektorer har samme *lengde* (1).

# Vektorer

For to vektorer $\vec{x}$ og $\vec{y}$ har vi et *indreprodukt*

$$\vec{x} \cdot \vec{y} = \sum_i x_i y_i$$

Vi kan *normalisere* en vektor $\vec{x}/|\vec{x}|$.

Alle normaliserte vektorer har samme *lengde* (1).

Vi har et naturlig mål for *avstanden* mellom to dokumenter

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} = \cos(\theta)$$

# Søking i vektorrom

En spørring kan vi se på som et kort dokument. Hvis vi har en spørring med to termer $t_i$ og $t_j$ har vi en vektor

$$\vec{V}(q) = (0, 0, \ldots, \text{tf-idf}_{q,i}, \ldots, \text{tf-idf}_{q,j}, \ldots, 0, 0)$$

Altså en vektor der alle elementene (utenom to) er null.

Da blir

$$\vec{V}(d) \cdot \vec{V}(q) = (\text{tf-idf}_{q,i} \times \text{tf-idf}_{d,i}) + (\text{tf-idf}_{q,j} \times \text{tf-idf}_{d,j})$$

lett å beregne.

# Søking i vektorrom

En spørring kan vi se på som et kort dokument. Hvis vi har en spørring med to termer $t_i$ og $t_j$ har vi en vektor

$$\vec{V}(q) = (0, 0, \ldots, \text{tf-idf}_{q,i}, \ldots, \text{tf-idf}_{q,j}, \ldots, 0, 0)$$

Altså en vektor der alle elementene (utenom to) er null.

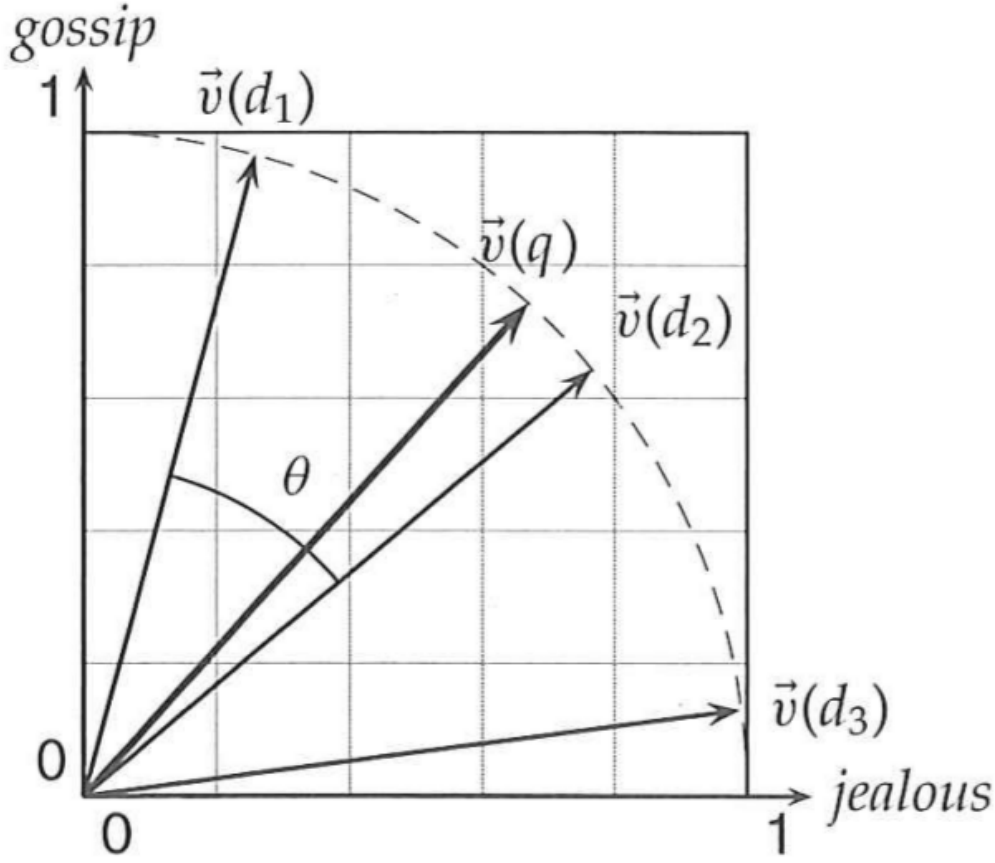
Da blir

$$\vec{V}(d) \cdot \vec{V}(q) = (\text{tf-idf}_{q,i} \times \text{tf-idf}_{d,i}) + (\text{tf-idf}_{q,j} \times \text{tf-idf}_{d,j})$$

lett å beregne. Også resten av ligningen for *sim* blir enkel:

1. $|\vec{V}(q)|$ er enkel å beregne, og
2. $|\vec{V}(d)|$ kan forhåndsberegnes.

# Søking i vektorrom



# Alternative poengligninger - samme prinsipp

Vi har en million dokumenter og vi søker etter ``best car insurance''. Et dokument *d* inneholder ``car'' en gang, ``insurance'' to ganger, men ikke ``best''.

| term | query | | | | document | | | product |
|---|---|---|---|---|---|---|---|---|
| | tf | df | idf | $w_{t,q}$ | tf | wf | $w_{t,d}$ | |
| auto | 0 | 5000 | 2.3 | 0 | 1 | 1 | 0.41 | 0 |
| best | 1 | 50000 | 1.3 | 1.3 | 0 | 0 | 0 | 0 |
| car | 1 | 10000 | 2.0 | 2.0 | 1 | 1 | 0.41 | 0.82 |
| insurance | 1 | 1000 | 3.0 | 3.0 | 2 | 2 | 0.82 | 2.46 |

Dette dokumentet får $0 + 0 + 0.82 + 2.46 = 3.28$ poeng.

# Alternative poengligninger - samme prinsipp

Vi har en million dokumenter og vi søker etter ``best car insurance''. Et dokument *d* inneholder ``car'' en gang, ``insurance'' to ganger, men ikke ``best''.

| term | query | | | | document | | | product |
|---|---|---|---|---|---|---|---|---|
| | tf | df | idf | $w_{t,q}$ | tf | wf | $w_{t,d}$ | |
| auto | 0 | 5000 | 2.3 | 0 | 1 | 1 | 0.41 | 0 |
| best | 1 | 50000 | 1.3 | 1.3 | 0 | 0 | 0 | 0 |
| car | 1 | 10000 | 2.0 | 2.0 | 1 | 1 | 0.41 | 0.82 |
| insurance | 1 | 1000 | 3.0 | 3.0 | 2 | 2 | 0.82 | 2.46 |

Dette dokumentet får $0 + 0 + 0.82 + 2.46 = 3.28$ poeng.

$w_{t,q} = idf_t$ hvis $tf > 0$, og 0 ellers.
$w_{t,d} = tf_{t,d}/|\vec{V}(d)| = tf_{t,d}/\sqrt{6}$.

# Alternative poengligninger - sublineær tf

Hvis et dokument $d_1$ inneholder termen $t$ 20 ganger så ofte som $d_2$, er det rimelig å anta at $d_1$ er 20 ganger så relevant?

# Alternative poengligninger - sublineær tf

Hvis et dokument $d_1$ inneholder termen $t$ 20 ganger så ofte som $d_2$, er det rimelig å anta at $d_1$ er 20 ganger så relevant?

$$wf_{t,d} = \begin{cases} 1 + \log tf_{t,d} & \text{, hvis } tf_{t,d} > 0, \\ 0 & \text{, ellers} \end{cases}$$

# Alternative poengligninger - sublineær tf

Hvis et dokument $d_1$ inneholder termen $t$ 20 ganger så ofte som $d_2$, er det rimelig å anta at $d_1$ er 20 ganger så relevant?

$$wf_{t,d} = \begin{cases} 1 + \log tf_{t,d} & \text{, hvis } tf_{t,d} > 0, \\ 0 & \text{, ellers} \end{cases}$$

På samme måte som vi definerer tf-idf ift. tf, kan vi definere wd-idf som

$$\text{wf-idf}_{t,d} = wf_{t,d} \times idf_t$$

# Alternative poengligninger - SMART

| term frequency | | document frequency | | normalization | |
|---|---|---|---|---|---|
| n (natural) | $\mathrm{tf}_{t,d}$ | n (no) | 1 | n (none) | 1 |
| l (logarithm) | $1+\log(\mathrm{tf}_{t,d})$ | t (idf) | $\log \frac{N}{\mathrm{df}_t}$ | c (cosine) | $\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$ |
| a (augmented) | $0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$ | p (prob idf) | $\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$ | u (pivoted unique) | $1/u$ (Section 17.4.4) |
| b (boolean) | $\begin{cases}1 \text{ if } \mathrm{tf}_{t,d}>0\\0 \text{ otherwise}\end{cases}$ | | | b (byte size) | $1/CharLength^{\alpha}$, $\alpha<1$ |
| L (log ave) | $\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$ | | | | |

SMART systemet angir vektingsalgoritem for query (*qqq*) og dokumenter (*ddd*) som *ddd*.*qqq*.

Hva betyr *nnc.ntn*?

# CosineScore

1. float `Scores[N] = 0`
2. Klargjør `Length[N]`
3. **for each** sprørreterm `t`
4. **do** beregn $w_{t,q}$ og hent treffene for `t`
5. **for each** $(d, tf_{t,d})_t$
6. **do** `Scores[d]` += $wf_{t,d} \times w_{t,q}$
7. Les inn `Length[d]`
8. **for each** $d$
9. **do** `Scores[d] = Scores[d] / Length[d]`
10. **return** beste $K$ komponentene i `Scores[]`