AI för naturligt språk

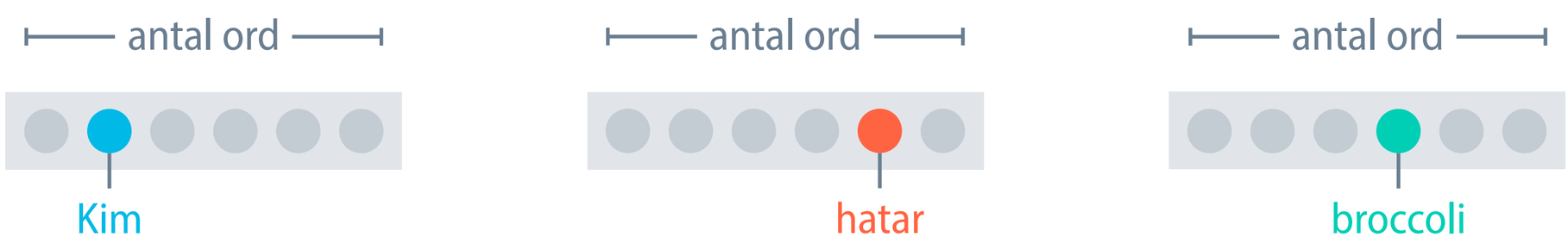
Ordinbäddningar

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

One hot-vektorer

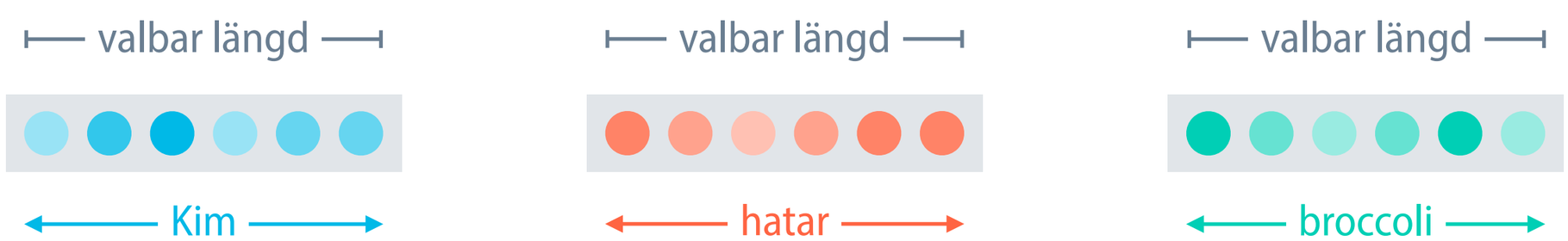
- För att behandla ord med hjälp av neuronnät måste vi representera dem som vektorer.
- Den klassiska representationen är så kallade **one hot-vektorer** – vektorer där alla komponenter förutom en tar värdet noll.



Ordinbäddningar

Ordinbäddningar (eng. *word embeddings*)

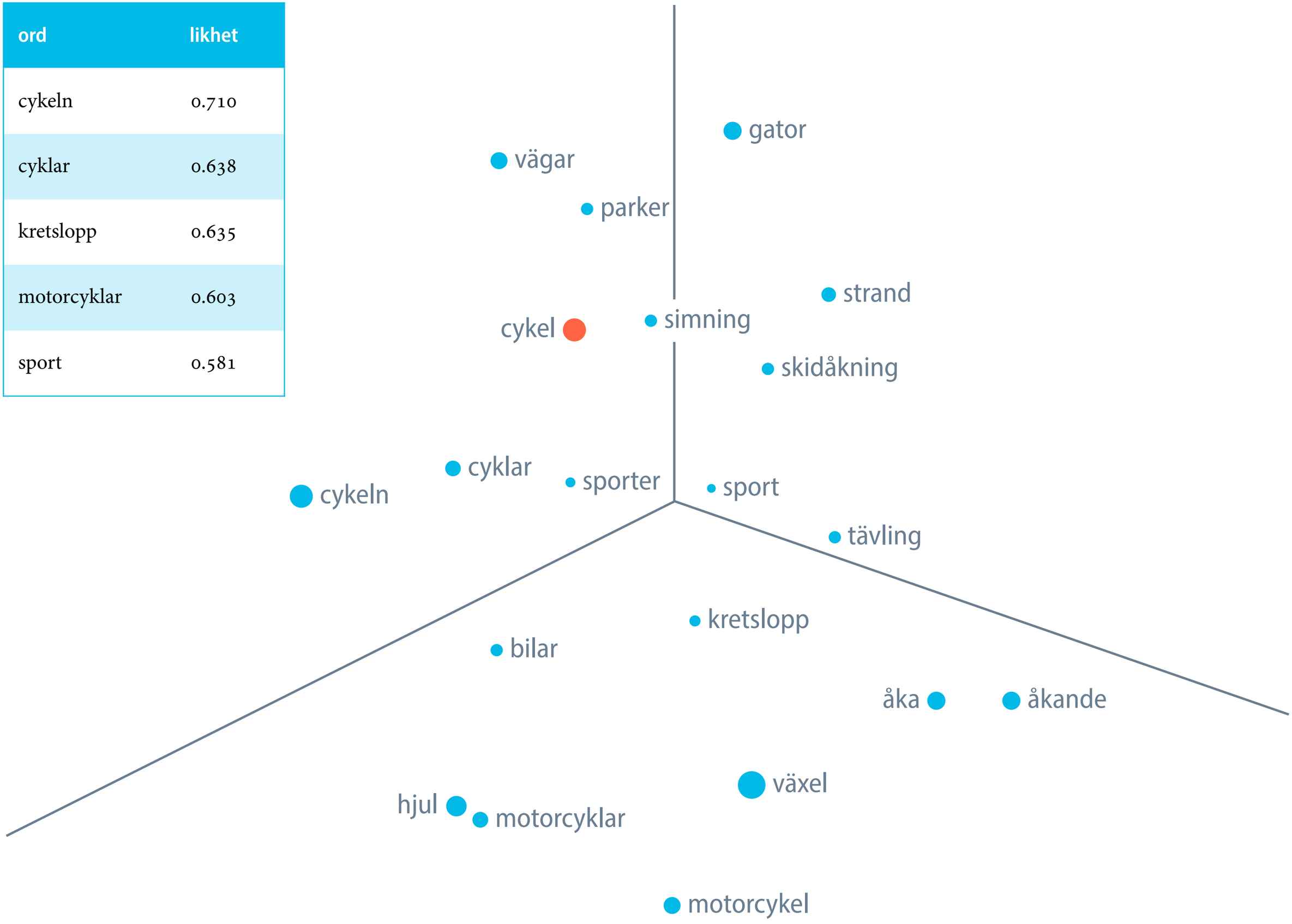
- är mycket kortare än one hot-vektorer, men täta
- gör det möjligt att kvantifiera likhet mellan ord
- kan tränas upp automatiskt utifrån textdata



En 100-dimensionell inbäddning för ordet ”cykel”

0.163095	−0.359647	−0.046412	−0.010649	0.718955
−0.111966	0.050354	0.719060	0.169119	−0.117902
0.425276	0.303177	−0.613056	−0.165227	0.123483
−0.311939	−0.445847	−0.032460	0.231660	0.108420
−0.018938	0.011204	0.052003	0.096326	−0.164183
0.412722	0.090491	−0.377895	0.148951	0.419733
−0.180430	0.090463	0.178629	0.013109	0.008413
0.027425	0.754911	−0.249114	0.104542	−0.106629
−0.324297	0.450337	0.092561	−0.122383	0.149844
0.377515	0.014969	−0.328139	0.329324	0.133836
0.251143	−0.154386	0.084984	0.607057	−0.400912
0.271288	−0.267989	−0.753885	−0.327919	0.055978
−0.077226	−0.031933	−0.236064	−0.139509	−0.109618
−0.651702	0.220722	−0.800192	−0.302251	0.031699
−0.213587	0.491182	−0.048589	−0.490870	0.337070
1.014193	0.285807	−0.689084	−0.101901	−0.066521
0.170587	0.078933	0.213985	0.034212	0.247812
0.173136	−0.162286	−0.509477	−0.570684	−0.165650
−0.227831	0.392880	0.476312	−0.014774	−0.254681
0.119459	−0.424725	−0.429033	−0.208388	0.067162

ord	likhet
cykeln	0.710
cyklar	0.638
kretslopp	0.635
motorcyklar	0.603
sport	0.581



Förtränade ordinbäddningar

- Ordinbäddningar tränas upp på stora textmängder, från miljtontals ord till flera miljarder ord.

- Eftersom träningen är kostsam använder många system förtränade ordinbäddningar.

- För svenska har bl.a. Arbetsförmedlingen och Kungliga biblioteket tagit fram förtränade ordinbäddningar.

  Lista över förtränade inbäddningar för svenska

AI för naturligt språk

Distributionell semantik

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

You shall know a word by the company it keeps

Vad säger dessa meningar om ordet *garrotxa*?

- *Garrotxa* tillverkas av mjölk.
- *Garrotxa* passar bra ihop med grovt bröd.
- *Garrotxa* lagras i grottor för att främja utveckligen av mögel.

Den distributionella hypotesen

- Den **distributionella hypotesen** hävdar att ord som förekommer i liknande textsammanhang även har liknande betydelser.
- Exempel på möjliga textsammanhang är de närmast omkringliggande orden eller alla ord i samma dokument.
- Om den distributionella hypotesen stämmer kan inbäddningar läras in utifrån statistik över samförekomster.

Samförekomstmatris

	cheese	bread	goat	sheep
cheese				
bread				
goat				
sheep				

as olives cheese or bread

Samförekomstmatris

	cheese	bread	goat	sheep
cheese		1		
bread				
goat				
sheep				

as olives cheese or bread

of sheep cheese and milk

Samförekomstmatris

	cheese	bread	goat	sheep
cheese		1		1
bread				
goat				
sheep				

as olives cheese or bread

of sheep cheese and milk

goat milk cheese can be

Samförekomstmatris

	cheese	bread	goat	sheep
cheese		1	1	1
bread				
goat				
sheep				

as olives cheese or bread

of sheep cheese and milk

goat milk cheese can be

bread and cheese for breakfast

Samförekomstmatris

	cheese	bread	goat	sheep
cheese		2	1	1
bread				
goat				
sheep				

as olives cheese or bread

of sheep cheese and milk

goat milk cheese can be

bread and cheese for breakfast

macaroni and cheese with bread

Samförekomstmatris

	cheese	bread	goat	sheep
cheese		3	1	1
bread				
goat				
sheep				

as olives cheese or bread

of sheep cheese and milk

goat milk cheese can be

bread and cheese for breakfast

macaroni and cheese with bread

Samförekomstmatris

	cheese	bread	goat	sheep
cheese	14	7	5	1
bread	7	12	0	0
goat	5	0	8	12
sheep	1	0	12	2

vektor för ordet *cheese*

Samförekomstmatris

	cheese	bread	goat	sheep
cheese	14	7	5	1
bread	7	12	0	0
goat	5	0	8	12
sheep	1	0	12	2

cheese förekommer oftare tillsammans med *bread* än med *goat* och *sheep*

goat förekommer oftare tillsammans med *sheep* än med *cheese* och *bread*

AI för naturligt språk

Att mäta likhet mellan ord

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

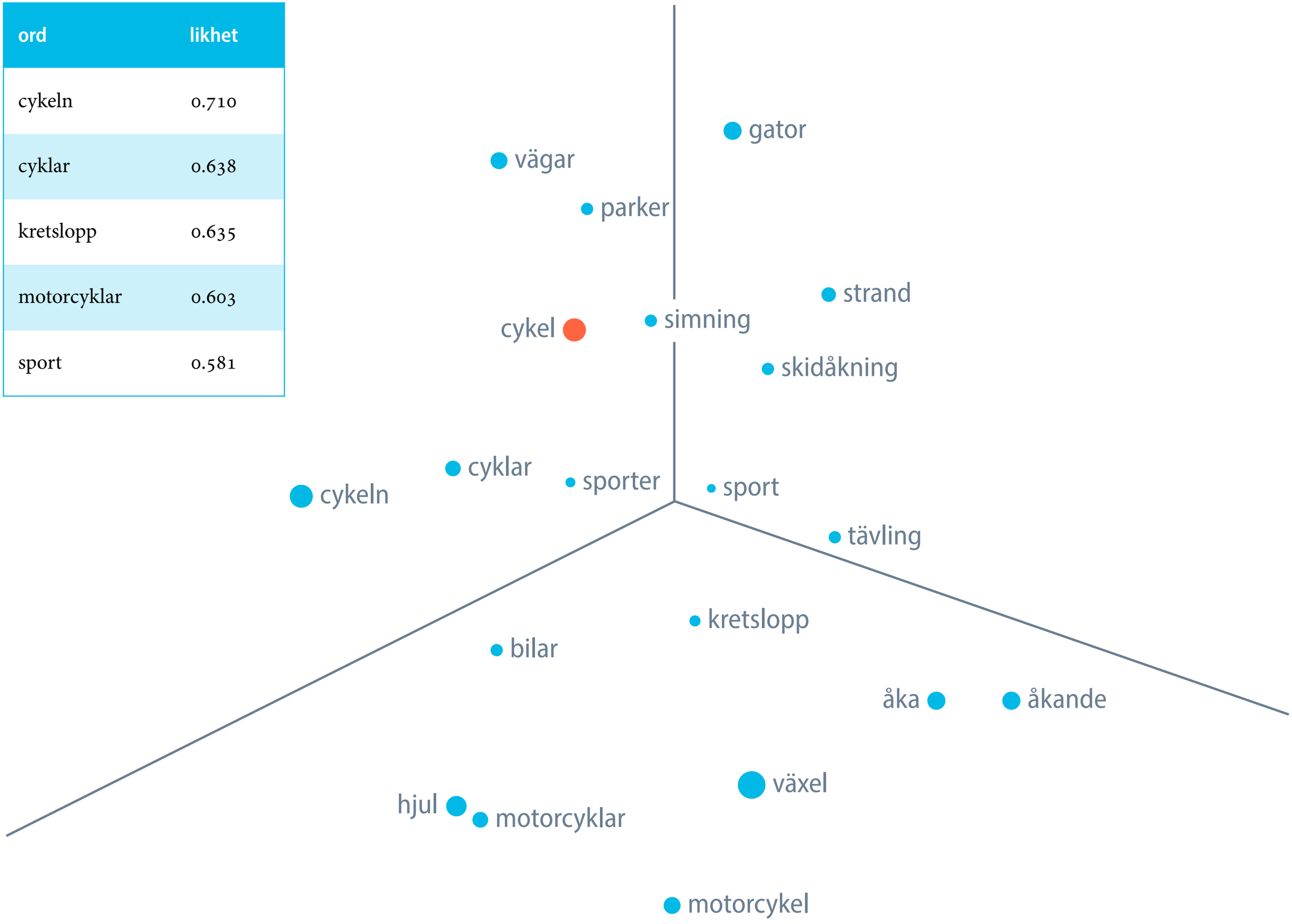
Samförekomstmatris

	cheese	bread	goat	sheep
cheese	14	7	5	1
bread	7	12	0	0
goat	5	0	8	12
sheep	1	0	12	2

cheese förekommer oftare tillsammans med *bread* än med *goat* och *sheep*

goat förekommer oftare tillsammans med *sheep* än med *cheese* och *bread*

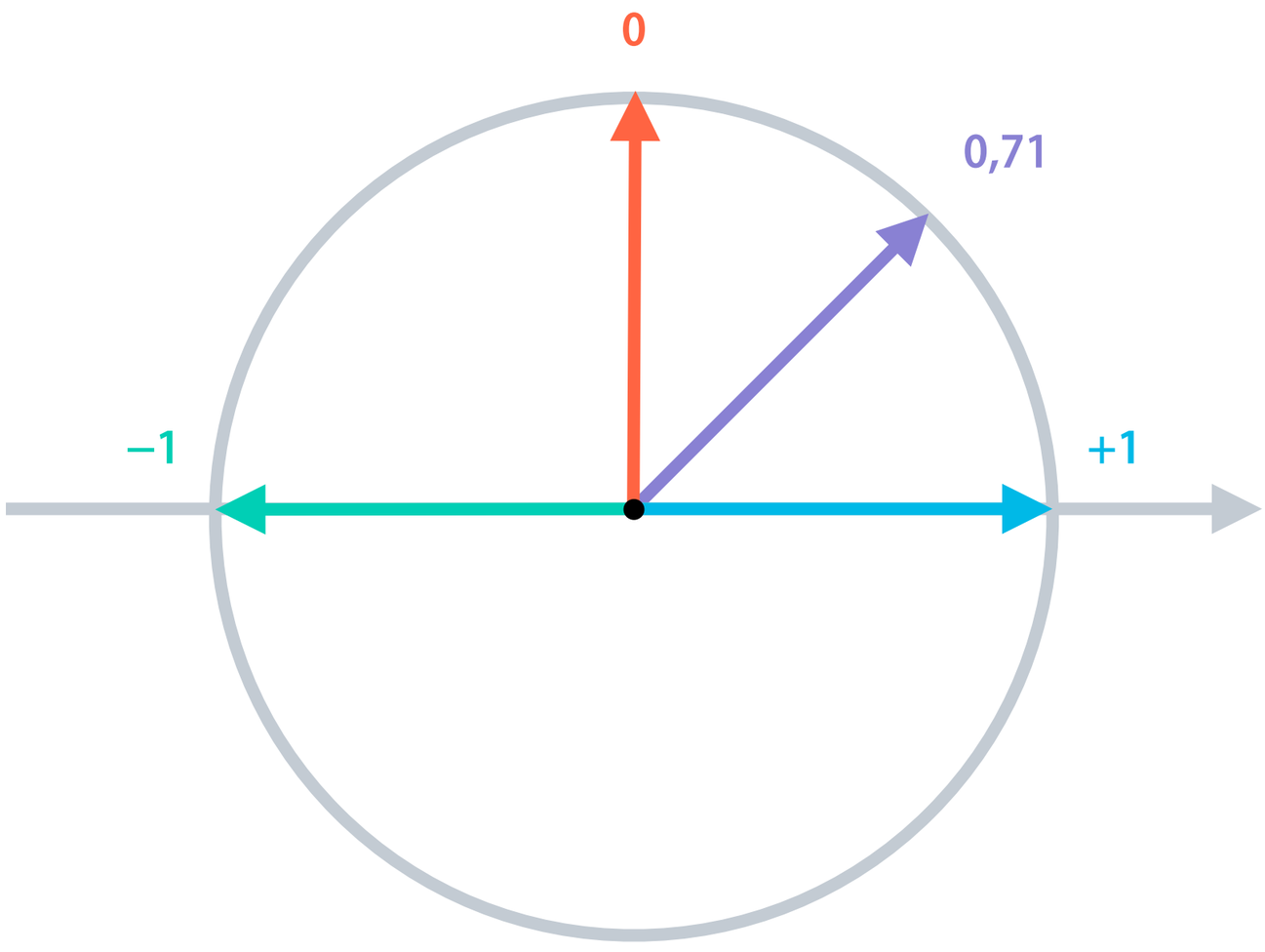
ord	likhet
cykeln	0.710
cyklar	0.638
kretslopp	0.635
motorcyklar	0.603
sport	0.581



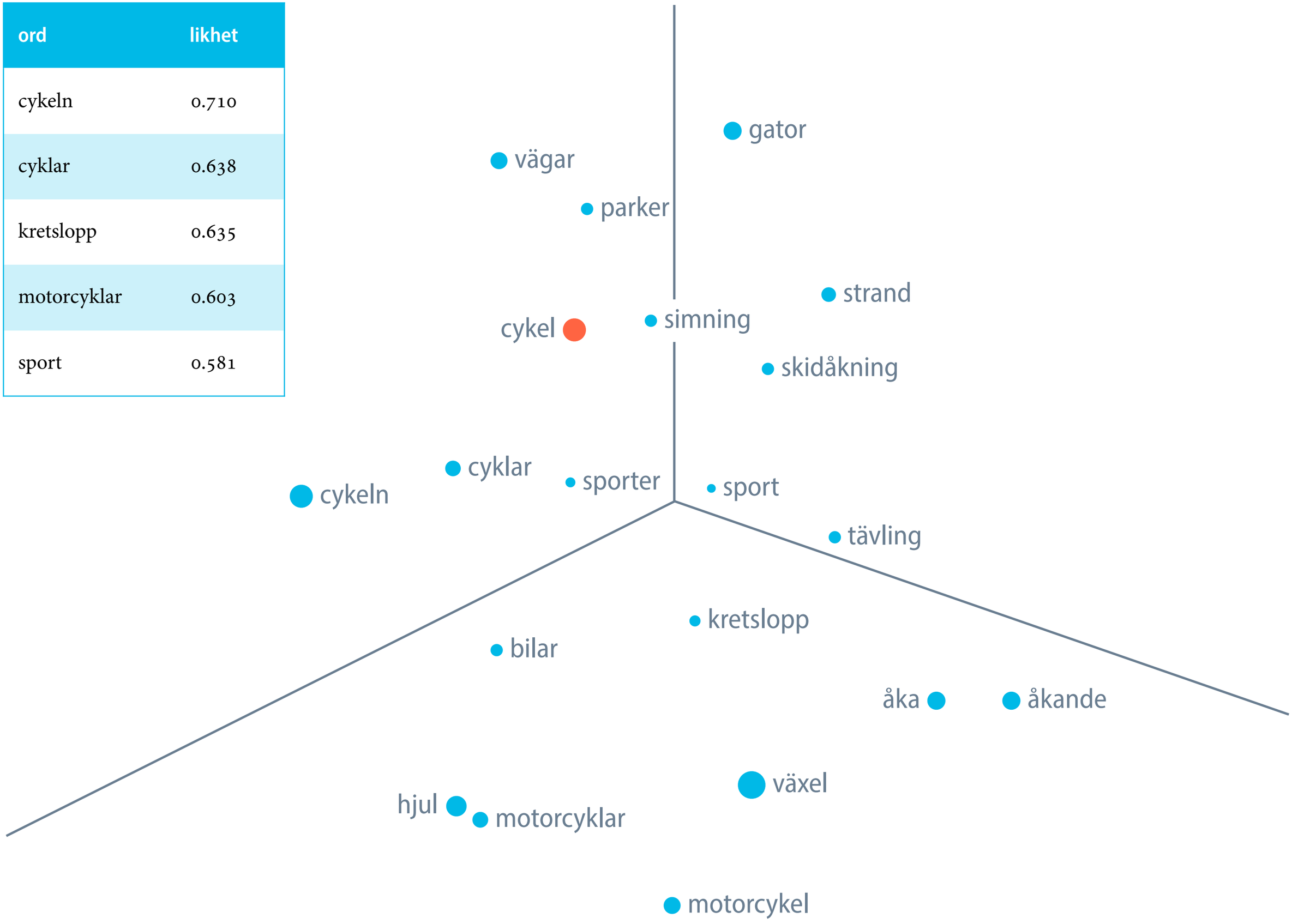
Cosinuslikhet

- För att mäta likhet mellan ordinbäddningar (ordvektorer) används ofta måttet **cosinuslikhet**.
- Cosinuslikhet mellan två vektorer är definierad som cosinusvärdet för vinkeln mellan vektorerna.

Cosinuslikhet



ord	likhet
cykeln	0.710
cyklar	0.638
kretslopp	0.635
motorcyklar	0.603
sport	0.581



AI för naturligt språk

Ordinbäddningar via matriskomprimering

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

Att läsa av ordvektorer från en samförekomstmatris

	cheese	bread	goat	sheep
cheese	14	7	5	1
bread	7	12	0	0
goat	5	0	8	12
sheep	1	0	12	2

ordvektor för *cheese*

Ordinbäddning via matriskomprimering

- De ordvektorer som vi kan läsa av från en samförekomstmatris är långa (hundratusentals komponenter) och glesa (många nollor).

- För att få vektorer som är korta och täta kan vi komprimera samförekomstmatrisen, dvs. reducera antalet kolumner.

- En teknik som ofta används i detta sammanhang är **trunkerad singulärvärdesdesuppdelning**.

  eng. *singular value decomposition* (SVD)

Intuition – bildkompression

d = 200

d = 100

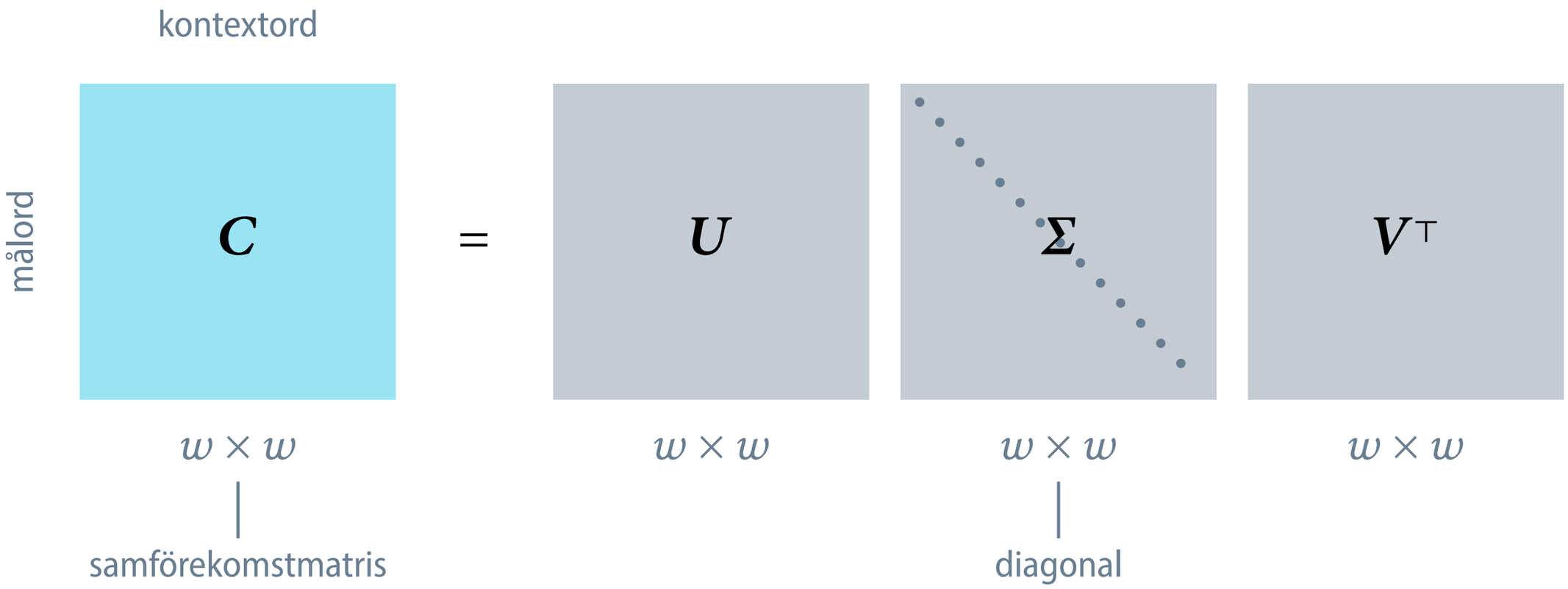
d = 50

d= 20

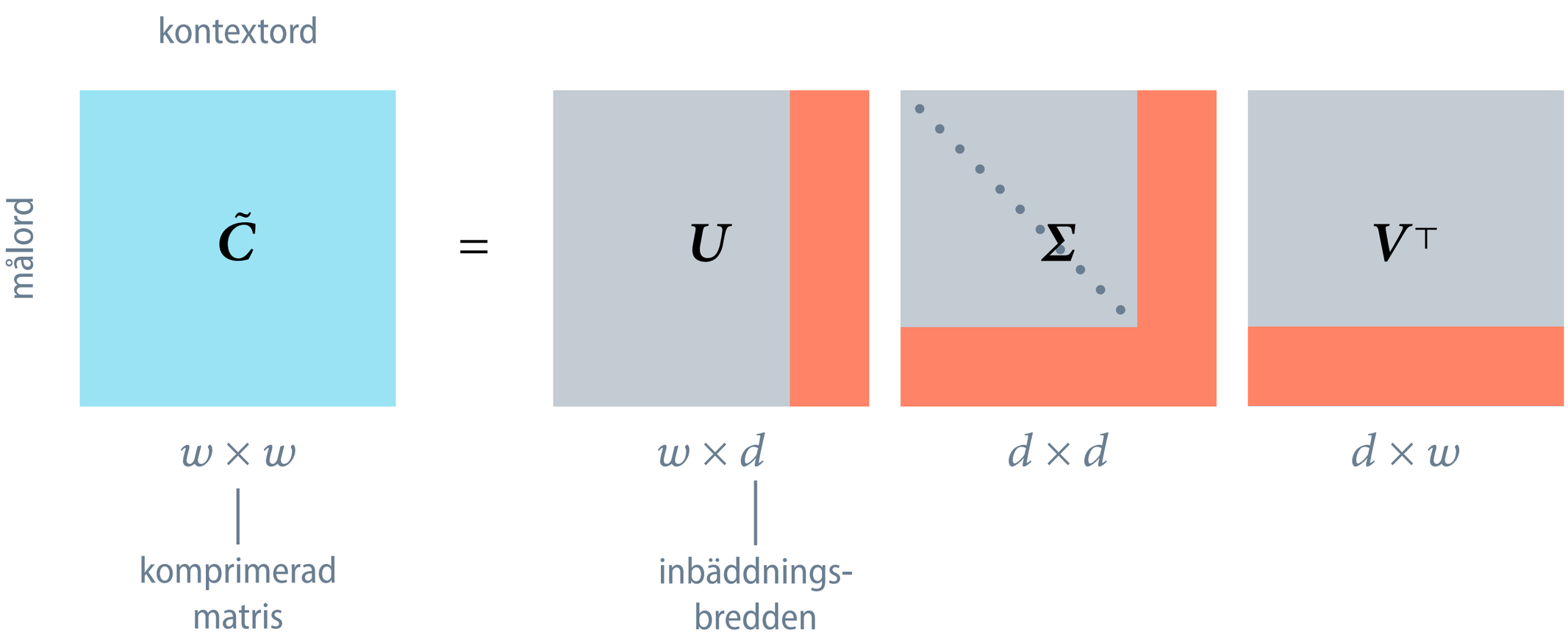
d = 10

d = 5

Singulärvärdesuppdelning



Truncated singular value decomposition



Deerwester et al. (1990)

ursprunglig samförekomstmatris

14	7	5	1
7	12	0	0
5	0	8	12
1	0	12	2

U

−0.678	0.293	0.124	0.662
−0.445	0.562	−0.044	−0.696
−0.491	−0.591	−0.628	−0.124
−0.318	−0.499	0.767	−0.248

Σ

22.676	0.000	0.000	0.000
0.000	15.644	0.000	0.000
0.000	0.000	7.655	0.000
0.000	0.000	0.000	5.335

$V^\top$

−0.678	−0.445	−0.491	−0.318
0.293	0.562	−0.591	−0.499
−0.124	0.044	0.628	−0.767
0.662	−0.696	−0.124	−0.248

ursprunglig samförekomstmatris

14	7	5	1
7	12	0	0
5	0	8	12
1	0	12	2

U

−0.678	0.293	0.124	0.662
−0.445	0.562	−0.044	−0.696
−0.491	−0.591	−0.628	−0.124
−0.318	−0.499	0.767	−0.248

Σ

22.676	0.000	0.000	0.000
0.000	15.644	0.000	0.000
0.000	0.000	7.655	0.000
0.000	0.000	0.000	5.335

$V^\top$

−0.678	−0.445	−0.491	−0.318
0.293	0.562	−0.591	−0.499
−0.124	0.044	0.628	−0.767
0.662	−0.696	−0.124	−0.248

rekonstruktion med d = 3

11.659	9.459	5.439	1.878
9.459	9.418	−0.461	−0.922
5.439	−0.461	7.918	11.835
1.878	−0.922	11.835	1.671

reconstructionsfel: 1.779

ursprunglig samförekomstmatris

14	7	5	1
7	12	0	0
5	0	8	12
1	0	12	2

U

−0.678	0.293	0.124	0.662
−0.445	0.562	−0.044	−0.696
−0.491	−0.591	−0.628	−0.124
−0.318	−0.499	0.767	−0.248

Σ

22.676	0.000	0.000	0.000
0.000	15.644	0.000	0.000
0.000	0.000	7.655	0.000
0.000	0.000	0.000	5.335

$V^\top$

−0.678	−0.445	−0.491	−0.318
0.293	0.562	−0.591	−0.499
−0.124	0.044	0.628	−0.767
0.662	−0.696	−0.124	−0.248

rekonstruktion med d = 3

11.659	9.459	5.439	1.878
9.459	9.418	−0.461	−0.922
5.439	−0.461	7.918	11.835
1.878	−0.922	11.835	1.671

rekonstruktionsfel: 1.779

rekonstruktion med d = 2

11.776	9.417	4.845	2.605
9.417	9.432	−0.249	−1.181
4.845	−0.249	10.934	8.148
2.605	−1.181	8.148	6.178

rekonstruktionsfel: 5.442

ursprunglig samförekomstmatris

14	7	5	1
7	12	0	0
5	0	8	12
1	0	12	2

U

−0.678	0.293	0.124	0.662
−0.445	0.562	−0.044	−0.696
−0.491	−0.591	−0.628	−0.124
−0.318	−0.499	0.767	−0.248

Σ

22.676	0.000	0.000	0.000
0.000	15.644	0.000	0.000
0.000	0.000	7.655	0.000
0.000	0.000	0.000	5.335

$V^\top$

−0.678	−0.445	−0.491	−0.318
0.293	0.562	−0.591	−0.499
−0.124	0.044	0.628	−0.767
0.662	−0.696	−0.124	−0.248

rekonstruktion med d = 3

11.659	9.459	5.439	1.878
9.459	9.418	−0.461	−0.922
5.439	−0.461	7.918	11.835
1.878	−0.922	11.835	1.671

rekonstruktionsfel: 1.779

rekonstruktion med d = 2

11.776	9.417	4.845	2.605
9.417	9.432	−0.249	−1.181
4.845	−0.249	10.934	8.148
2.605	−1.181	8.148	6.178

rekonstruktionsfel: 5.442

rekonstruktion med d = 1

10.436	6.843	7.552	4.888
6.843	4.486	4.951	3.205
7.552	4.951	5.465	3.537
4.888	3.205	3.537	2.289

rekonstruktionsfel: 20.737

AI för naturligt språk

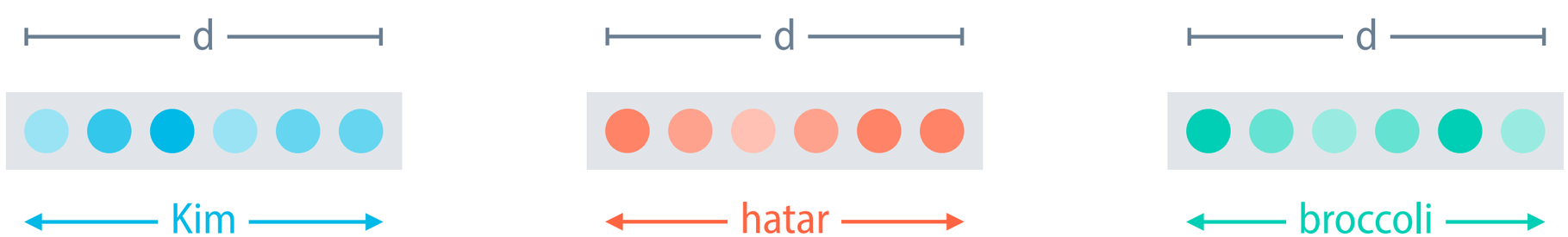
Ordinbäddningar via neuronnät

Marco Kuhlmann
Institutionen för datavetenskap

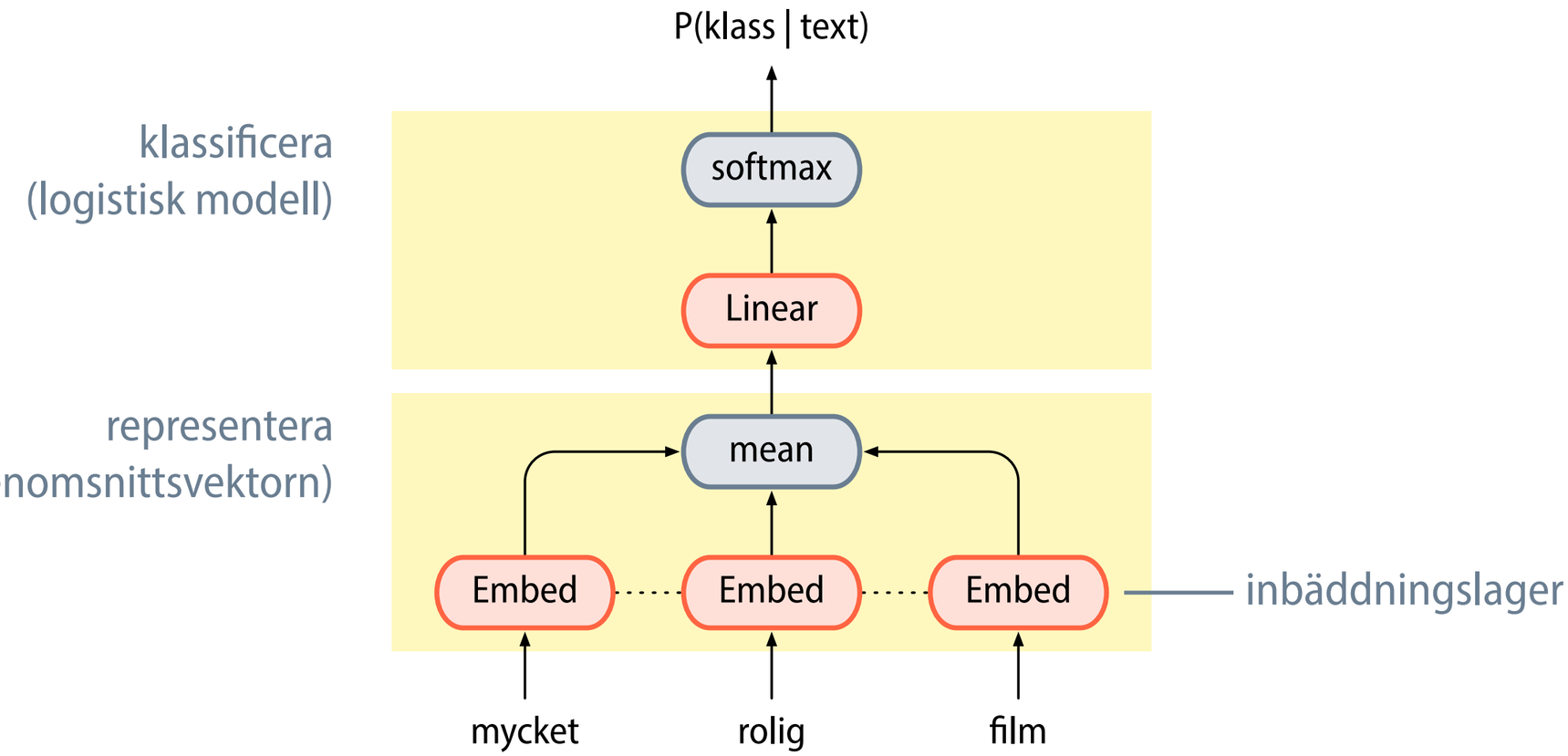
LINKÖPINGS UNIVERSITET

Inbäddningslager

- För att träna upp ordinbäddningar med hjälp av neuronnät används så kallade **inbäddningslager** (eng. *embedding layers*).
- Ett inbäddningslager är i princip inte mer än en tabell som kopplar varje ord till en d-dimensionell vektor.
- Dessa vektorer tränas, likt alla andra parametrar i nätet.



En enkel neuronal textklassificerare



Uppgiftsspecifika ordinbäddningar

- När vi tränar ett neuronnät optimeras nätets parametrar för den specifika uppgift som vi har valt att träna på.

  exempel: sentimentanalys

- Eftersom ordinbäddningarna nu ingår bland dessa parametrar, optimeras även dem för den specifika träningsuppgiften.

- Det är träningsuppgiften som avgör vad de tränade ordinbäddningarna ”betyder”.

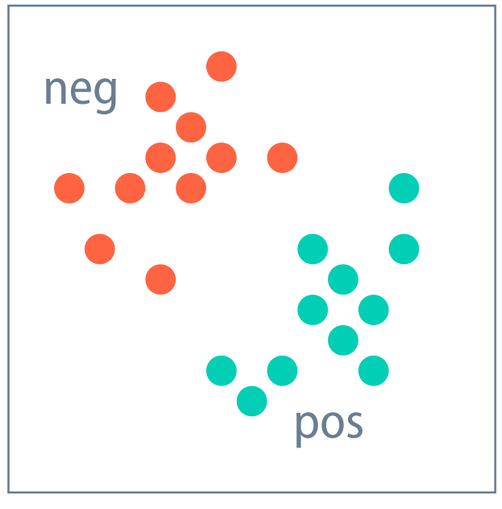
Två olika perspektiv på ordinbäddningar

- **Distributionell semantik**

  liknande ordinbäddningar ⇒ orden har liknande betydelse

- **Neuronnät**

  liknande ordinbäddningar ⇒ orden beter sig på liknande sätt relativt till uppgiften



AI för naturligt språk

Googles word2vec

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

Googles word2vec

- Googles word2vec är en välkänd mjukvara för att träna upp ordinbäddningar från textdata.
- Den enda träningsdata word2vec behöver är segmenterad text; den kräver inga annotationer.

  bra – för segmented text finns det mycket av!

- Googles word2vec implementerar två olika algoritmer. Här går vi igenom den så kallade **skip gram-algoritmen**.

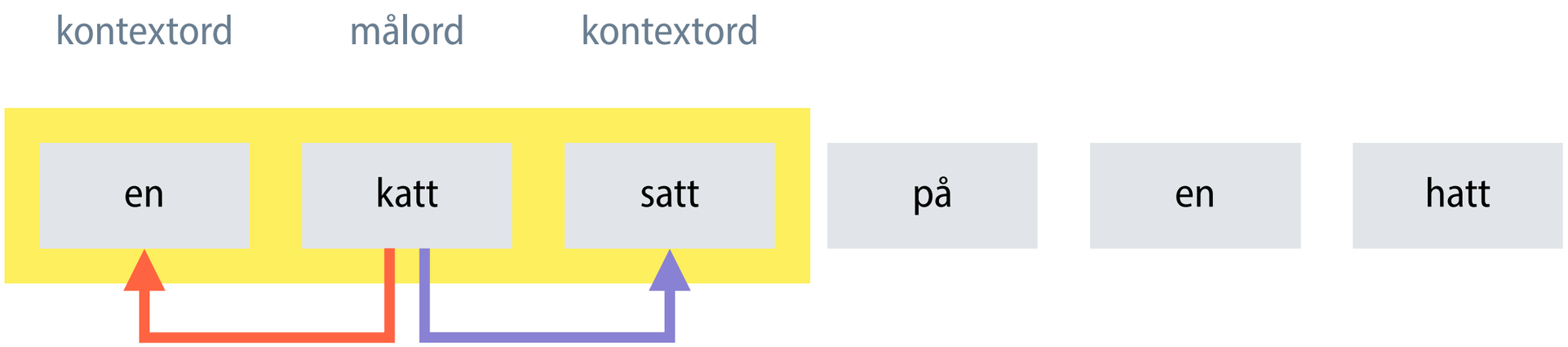
Mikolov et al. (2013)

Att träna en skip gram-modell

- Börja med framslumpade ordvektorer.
- Gå igenom alla målord w i texten, från början till slut, och välj ut kontextord c i närheten av w.
- Låt modellen beräkna likheten mellan w och c och konvertera denna likhet till en betingad sannolikhet $P(c|w)$.

  Hur sannolikt är det att se c i närheten av w?
- Träna ordvektorerna så att dessa sannolikheter maximeras.

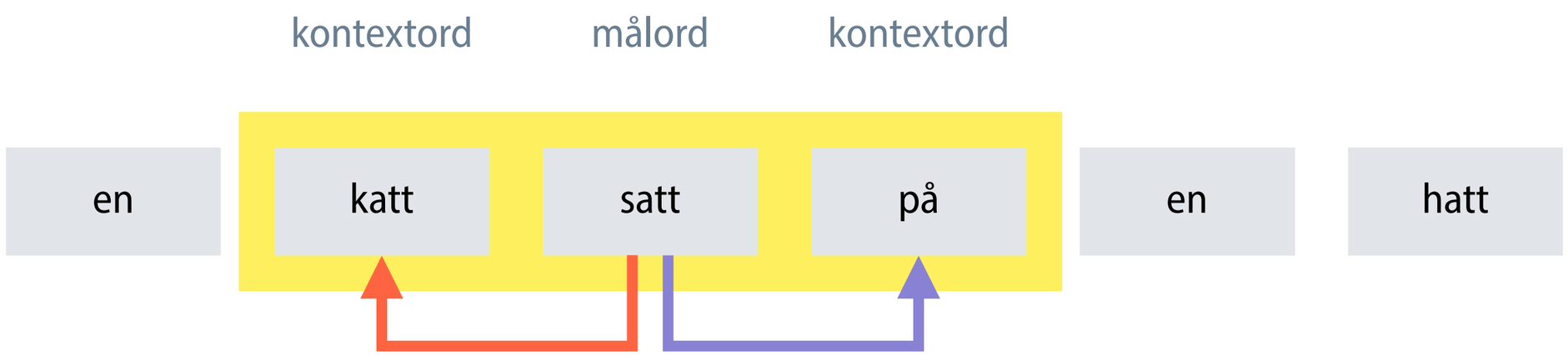
Att träna en skip gram-modell



Sannolikheten $P(en|katt)$ är desto större, ju mer lika vektorerna för *katt* och *en* är.

vektorer för målorden	$\boldsymbol{v}_{\text{en}}$	$\boldsymbol{v}_{\text{hatt}}$	$\boldsymbol{v}_{\text{katt}}$	$\boldsymbol{v}_{\text{på}}$	$\boldsymbol{v}_{\text{satt}}$
vektorer för kontextorden	$\boldsymbol{v}'_{\text{en}}$	$\boldsymbol{v}'_{\text{hatt}}$	$\boldsymbol{v}'_{\text{katt}}$	$\boldsymbol{v}'_{\text{på}}$	$\boldsymbol{v}'_{\text{satt}}$

Att träna en skip gram-modell



Sannolikheten $P(katt|satt)$ är desto större, ju mer lika vektorerna för *satt* och *katt* är.

vektorer för målorden	$\boldsymbol{v}_{\text{en}}$	$\boldsymbol{v}_{\text{hatt}}$	$\boldsymbol{v}_{\text{katt}}$	$\boldsymbol{v}_{\text{på}}$	$\boldsymbol{v}_{\text{satt}}$
vektorer för kontextorden	$\boldsymbol{v}'_{\text{en}}$	$\boldsymbol{v}'_{\text{hatt}}$	$\boldsymbol{v}'_{\text{katt}}$	$\boldsymbol{v}'_{\text{på}}$	$\boldsymbol{v}'_{\text{satt}}$

AI för naturligt språk

Transferinlärning

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

Ordinbäddningar för transferinlärning

- **Transferinlärning** handlar om att återanvända kunskap som vi fått då vi tränat på en uppgift när vi vill lösa nästa uppgift.

  snabbar upp inlärningen, reducerar behovet av träningsdata

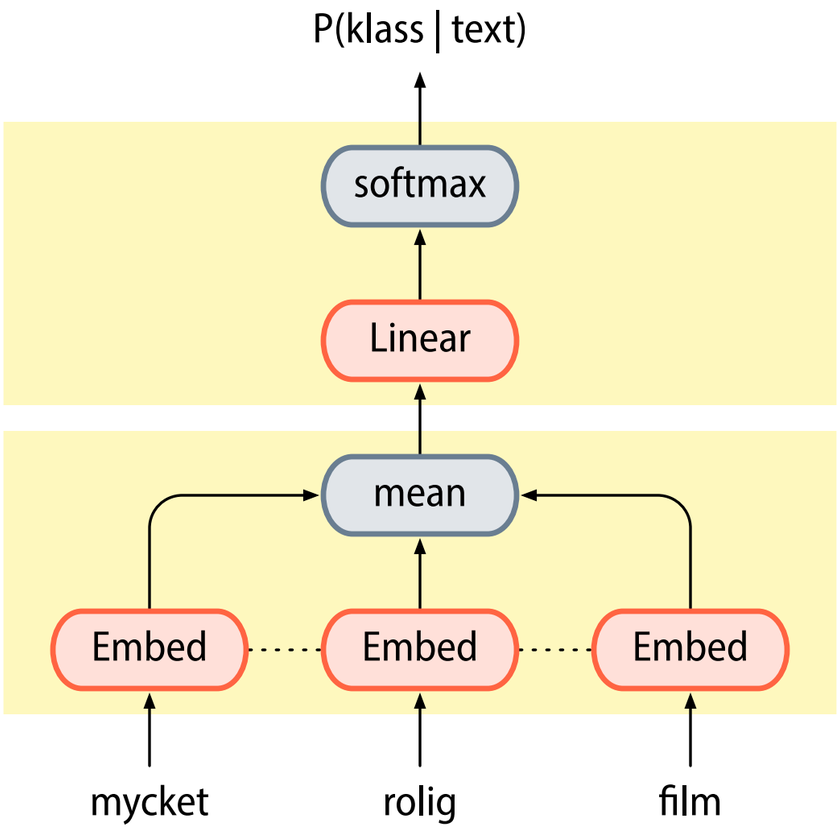
- Rent praktiskt innebär detta att vi återanvänder redan färdigt tränade delar av ett neuronnät i ett annat neuronnät.

- Vi skulle t.ex. kunna återanvända inbäddningslagren, snarare än att träna upp ordinbäddningar ”from scratch”.

En enkel neuronnätsklassificerare

klassificera
(logistisk modell)

representera
(genomsnittsvektorn)



svårt att återanvända

enkelt att återanvända

Att återanvända förtränade ordinbäddningar

Träna inbäddningar på uppgift *A* och använd dem för att initialisera inbäddningslagren i ett neuronnät för uppgift *B*.

Sedan finns två alternativ:

- **Alternativ 1:** Frysa vikterna i inbäddningslagren så att de inte uppdateras när nätet nu tränas på uppgift *B*.

- **Alternativ 2:** Träna nätet på uppgift *B* som vanligt; detta finjusterar de förtränade inbäddningarna för den nya uppgiften.

Vilka förträningsuppgifter ska vi välja?

- Vi vill ha representationer som är användbara i många olika sammanhang, så vi föredrar generella uppgifter.
- Förträning kräver data, så vi föredrar uppgifter för vilka vi kan hitta mycket data – det idealiska är segmenterad text.
- Den vanligaste uppgiften för att förträna ordinbäddningar är att förutsäga samförekomster, som i word2vec.

AI för naturligt språk

Begränsningar med ordinbäddningar

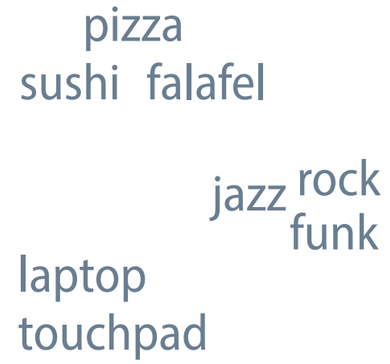
Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

Hur vet vi om vi har bra ordinbäddningar?

- visualisering av vektorrummet

  kräver dimensionalitetsreduktion (PCA, t-SNE, UMAP)



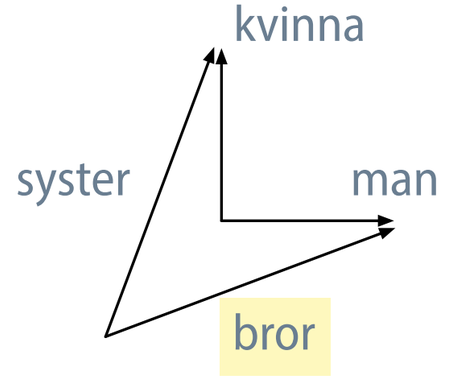
- beräkna relativa likheter

- utvärderingsprov som bygger på likhet

  det som inte passar in: *frukost lunch middag födelsedag*

- utvärderingsprov som bygger på analogier

  kvinna förhåller sig till *man* som *syster* till *?*



Begränsningar av ordinbäddningar

- Det finns många olika facetter av ”semantisk likhet”.

  Är en *katt* mer lik en *hund* eller en *tiger*?

- Det finns många semantiska egenskaper som vi inte skriver om.

  Det kan finnas fler *svarta får* än *vita får* i texten.

- Ordinbäddningar reflekterar biaser i träningsdatan.

  t.ex. relaterade till kön, etnisk ursprung, social ställning

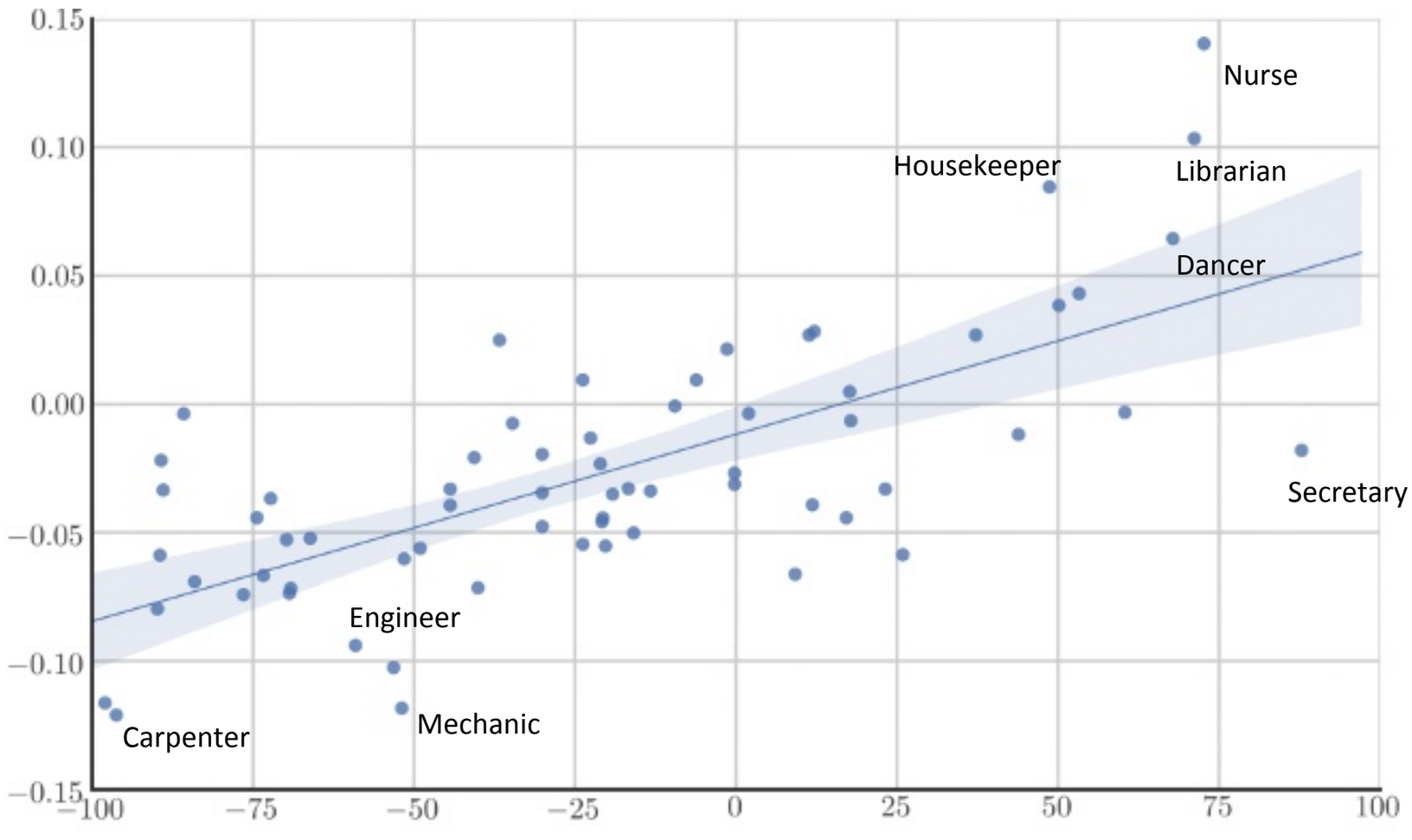
Goldberg (2017); Caliskan et al. (2017); Garg et al. (2018)

corned
potatoes
sauerkraut
beans
vegetables
mangoes
artichokes
cauliflower
nematodes
brassicaceae
ovaries
carrots
cabbage
delincee
cubby
connery
broccoli
saltzman
moonraker

word2vec All, 200d, 20 neighbours, UMAP

Source | Fir0002, GFDL 1.2

Inbäddningsbias och yrkesgrupper



Garg et al. (2018)

AI för naturligt språk

Kontextualiserade inbäddningar

Marco Kuhlmann
Institutionen för datavetenskap

LINKÖPINGS UNIVERSITET

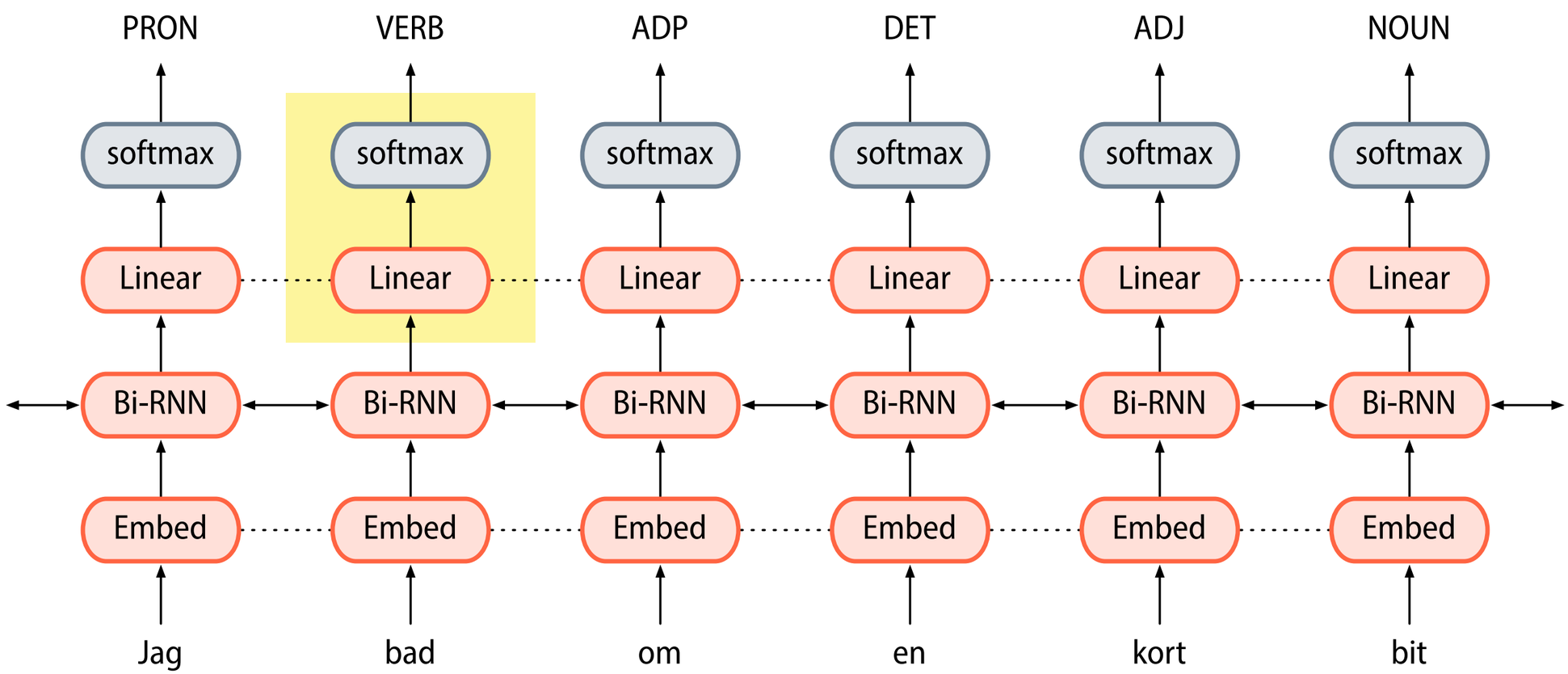
Kontextualiserade inbäddningar

- I vanliga inbäddningar tilldelas varje ord en enda vektor, oberoende av ordets sammanhang.
- En sådan modell kan inte hantera faktumet att ett och samma ord kan ha flera olika betydelser.

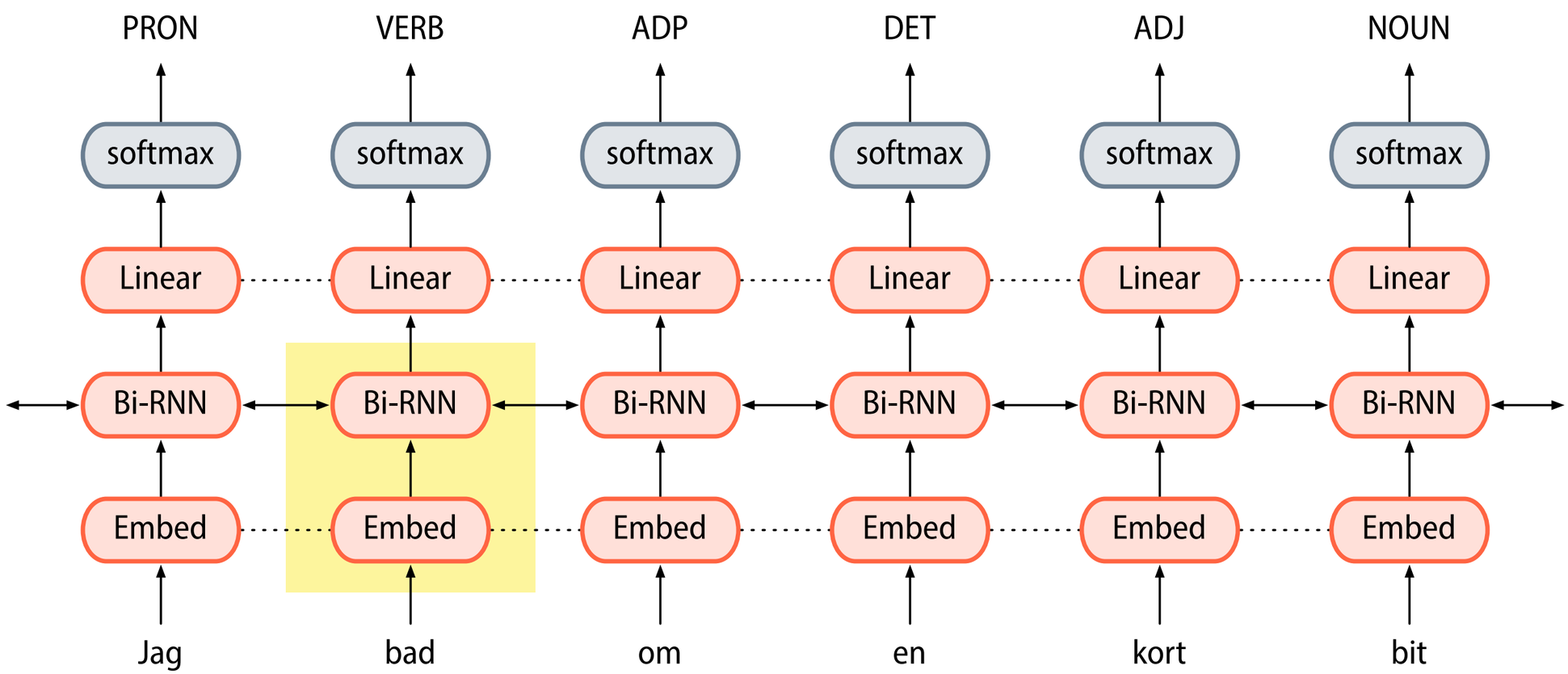
  Kim tog ett *bad*. De *bad* om att få återkomma senare.

- I en **kontextualiserad inbäddning** tilldelas varje ord en vektor som är beroende av hela meningen som ordet står i.

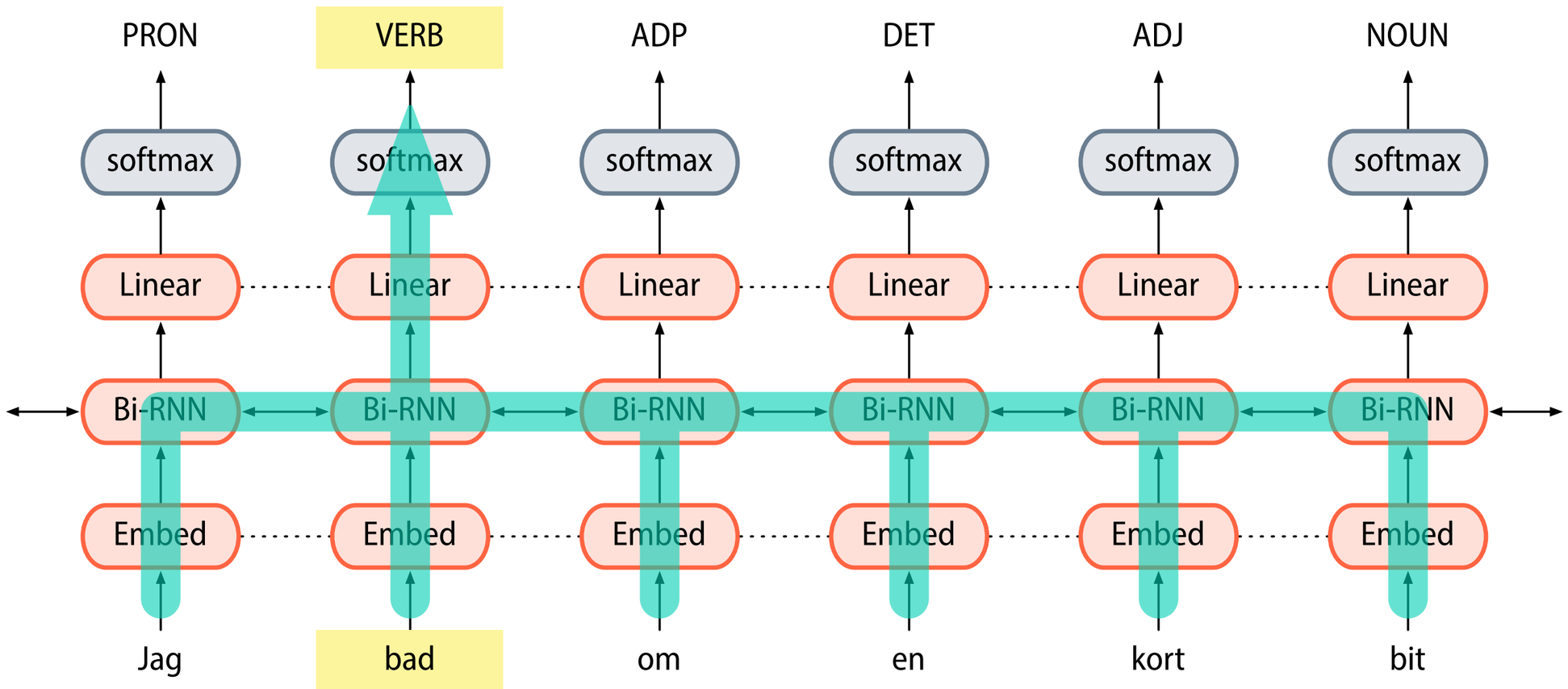
Ordklasstaggning med ett rekurrent neuronnät



Ordklasstaggning med ett rekurrent neuronnät



Ordklasstaggning med ett rekurrent neuronnät



Moderna inbäddningsmodeller

- De senaste åren har sett utvecklingen av ett stort antal mycket kraftfulla inbäddningsmetoder, såsom ELMo och BERT.
- Dessa modeller förtränas på mycket stora textmängder och finjusteras sedan för olika uppgifter.
- De definierar ”state of the art” för många uppgifter inom naturligt språk-behandling som tidigare krävt specialiserade modeller.

Källa: The Muppet Wiki