Projekt om vektorer i sprogteknologi

Sprogteknologi

En computer forstår umiddelbart ikke de sprog vi mennesker taler og skriver. Inden for sprogteknologien (på engelsk: Natural Language Processing eller NLP), der er en gren af kunstig intelligens, beskæftiger man sig med teknikker der netop gør dette muligt.

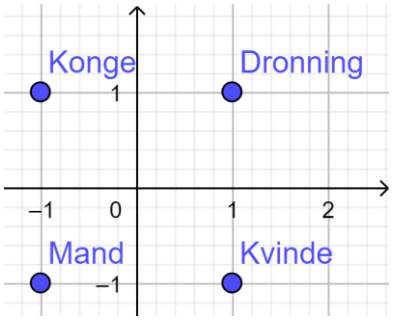
a. Find ud af mere om sprogteknologi på internettet. Hvad er nogle typiske anvendelser? Hvor mange af dem bruger du i din hverdag?

Word embeddings

Som sagt er en computer ikke særligt god til at forstå sprog som dansk, tysk og engelsk. Den er meget bedre til at regne med tal. Derfor er det ofte praktisk at lave ord om til tal, så computeren bedre kan håndtere dem. Et simpelt eksempel kunne være følgende:

Ord	Køn	Royalitet
Mand	-1	-1
Kvinde	1	-1
Konge	-1	1
Dronning	1	1

Her betyder en kønsværdi på -1 maskulin, og 1 feminin. Og en royalitets-værdi på -1 betyder almindelig borger, mens 1 svarer til at være adelig.

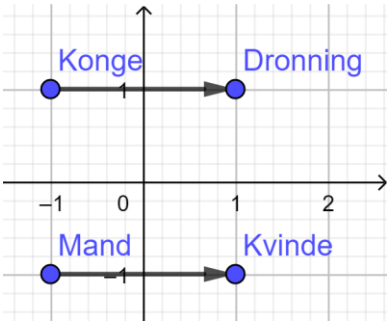


En sådan tildeling af værdier til forskellige ord kaldes en *word embedding* (der er så vidt jeg ved ikke et godt dansk ord for dette). Man kan vælge at tænke på værdierne der svarer til hvert ord som et punkt i planen – se figuren til venstre. Eller alternativt som den tilhørende stedvektor. Man vil derfor for eksempel skrive:

$$\overrightarrow{\text{Mand}} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

b. Hvad er vektorerne svarende til de tre andre ord?

I praktiske anvendelser er der selvfølgelig behov for mange flere værdier til at beskrive et ord – ofte flere hundrede – men her vil vi altså nøjes med at tænke i to dimensioner for nemheds skyld.



c. Beregn vektorerne $\overrightarrow{\text{Kvinde}} - \overrightarrow{\text{Mand}}$ og $\overrightarrow{\text{Dronning}} - \overrightarrow{\text{Konge}}$. Bemærker du noget påfaldende?

LIKE

Analogier mellem ord

At forskellen mellem to sæt af vektorer er ens (eller næsten ens) udtrykker en form for *analogi*. Du har måske set opgaver hvor man skal indsætte et ord der passer. F.eks.:

"*Danmark* forholder sig til *København* som *Rusland* forholder sig til _____"

Her kan de fleste nok gætte, at det manglende ord er *Moskva*, idet sammenhængen handler om lande og deres hovedstæder. Den type forhold mellem ord kaldes en analogi. Med word embeddings kan man formulere en analogi på vektorform:

"*Mand* forholder sig til *Kvinde*, som *Konge* forholder sig til *Dronning*"

$$\Leftrightarrow$$

$$\overrightarrow{\text{Kvinde}} - \overrightarrow{\text{Mand}} \approx \overrightarrow{\text{Dronning}} - \overrightarrow{\text{Konge}}$$

d. Beregn vektorerne $\overrightarrow{\text{Dronning}} - \overrightarrow{\text{Kvinde}}$ og $\overrightarrow{\text{Konge}} - \overrightarrow{\text{Mand}}$.
e. Udtryk konklusionen man kan drage af spørgsmål c som en analogi.

På hjemmesiden http://labs.statsbiblioteket.dk/dsc/ kan du under "Analogy" finde ud af hvilke ord der passer bedst ind i en analogi. Husk at vælge "Danish Newspapers 1900-2016" under "Select corpus" for at få danske ord (Du vil lære mere om hvad ordet "corpus" betyder i næste afsnit).

f. Hvilke tre ord passer ifølge siden bedst ind i analogien: "*Hammer* forholder sig til *Søm*, som *Sav* forholder sig til _____"
g. Find selv at finde på nogle andre analogier, og skriv et par af de bedste (eller måske værste – modellen er ikke altid lige imponerende) ned.

Mangel på analogi

Ovenstående viser hvordan man kan konstatere en analogi mellem ord. Tilsvarende vil man kunne vise en mangel på analogi hvis de to relevante vektorer er (meget) forskellige.

h. Undersøg om følgende analogi holder i vores oprindelige eksempel: "*Mand* forholder sig til *Dronning*, som *Konge* forholder sig til *Kvinde*".

Hvordan finder man word embeddings? Algoritmer!

I praksis kan man ikke sidde og lave embeddings for alle ord i ordbogen i hånden. I stedet bruger man forskellige *algoritmer*, altså en slags "opskrifter" for computere. Algoritmerne virker ved at tage en stor mængde tekster og kigge på hvordan de forskellige ord forekommer i forhold til hinanden.

En sådan mængde af tekster kaldes et *korpus* (engelsk: *corpus* – flertal: *corpora*). På hjemmesiden ovenfor var der mulighed for at vælge mellem modeller trænet på tre forskellige korpusser. Algoritmen der er benyttet der hedder *word2vec*, men det er ikke så vigtigt for os.

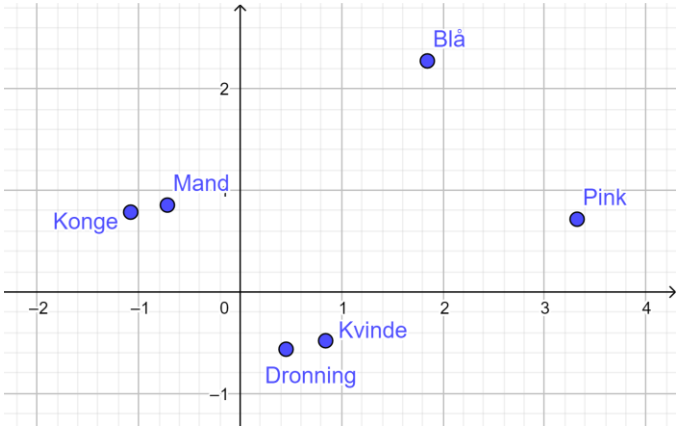
Endnu et simpelt eksempel

Man er dog sjældent så heldig, at værdierne er så nemme at fortolke som ovenfor; der vil sjældent være et enkelt tal der beskriver køn, f.eks. Det gør det sværere for et menneske at overskue hvad der egentlig foregår, og hvad der betyder hvad, mens computeren intet problem har.

Vi forestiller os nu vi har sat en algoritme til at træne på en mængde tekster. Tabellen nedenfor viser dens to-dimensionale embeddings for seks forskellige ord:

Ord	x	y
Mand	-0,72	0,85
Kvinde	0,84	-0,48
Konge	-1,08	0,79
Dronning	0,45	-0,57
Blå	1,84	2,27
Pink	3,32	0,72

Læg igen mærke til, at vi umiddelbart ikke ved hvad de to koordinater betyder, så derfor har vi blot kaldt dem x og y. Figuren til højre viser hvordan de seks ords embeddings ligger i planen.



i. Overvej hvorfor nogle ord ligger tæt på hinanden, mens andre ikke gør.

Ligheder mellem ord

En måde at give et mål for hvor ens eller forskellige to ord er fra hinanden, bruger man ofte det der kaldes *cosinus-similaritet* (engelsk: *cosine similarity*). Det er ikke så indviklet som det lyder: Man bruger den almindelige formel for vinklen mellem to vektorer, men i stedet for at beregne selve vinklen med $\cos^{-1}$ til sidst, bruger vi $\cos\theta$ som afstand.

Hvis vi f.eks. vil beregne cosinus-similariteten mellem *Mand* og *Konge* kan vi gøre det på følgende måde. Formlen for vinklen mellem to vektorer er:

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

I dette tilfælde betyder det:

$$\cos\theta = \frac{\begin{pmatrix}-0{,}72\\0{,}85\end{pmatrix}\cdot\begin{pmatrix}-1{,}08\\0{,}79\end{pmatrix}}{\left|\begin{pmatrix}-0{,}72\\0{,}85\end{pmatrix}\right|\left|\begin{pmatrix}-1{,}08\\0{,}79\end{pmatrix}\right|} = \frac{-0{,}72\cdot(-1{,}08)+0{,}85\cdot 0{,}79}{\sqrt{(-0{,}72)^2+0{,}85^2}\sqrt{(-1{,}08)^2+0{,}79^2}} = 0{,}97$$

I stedet for at beregne vinklen θ bruger vi nu simpelthen tallet 0,97 som et mål for lighed. Dette er cosinus-similariteten mellem *Mand* og *Konge*.

LIKE

j. Udfyld de manglende celler i følgende tabel:

Ord 1	Ord 2	Cosinus-similaritet
Mand	Konge	0,97
Mand	Kvinde	
Mand	Blå	
Dronning	Pink	

k. Mellem hvilke to tal kan værdien af cosinus-similariteter ligge? (Hint: Hvordan ser grafen for cosinus ud? Hvad er ekstremumsværdierne?)
l. Hvad betyder det, at to ord har en cosinus-similaritet tæt på 1?
m. Hvad betyder det, at to ord har en cosinus-similaritet tæt på -1?
n. Hvad betyder det, at to ord har en cosinus-similaritet tæt på 0?

Teknisk bemærkning: Dette ligheds-mål kommer ikke helt til sin ret her, da antallet af dimensioner er så lavt (2). I højere dimensioner vil langt de fleste par af ord have cosinus-similariteter tæt på nul, da der i en vis forstand er "mere plads" i rum med høj dimension. I vores eksempel får urelaterede ordpar som *Dronning* og *Pink* en kunstigt høj cosinus-similaritet af denne grund. Ordparret *Mand* og *Kvinde*, der her ser ud til at være modsætninger, ville tilsvarende have en cosinus-similaritet tæt på 1, da de begge er navneord, angiver personer osv. De adskiller sig faktisk kun på ét punkt: køn.

Hvilke ord minder mest om hinanden?

Hvis man har et givet ord kan man være interesseret i hvilke andre ord der ligner det mest, altså har egenskaber der er tættest på. Hvis man har en lang række embeddings for forskellige ord, kan man sammenligne cosinus-similariteterne mellem det givne ord og alle de andre ord i ordbogen. Herefter udvælger man det/de ord der havde højest cosinus-similaritet.

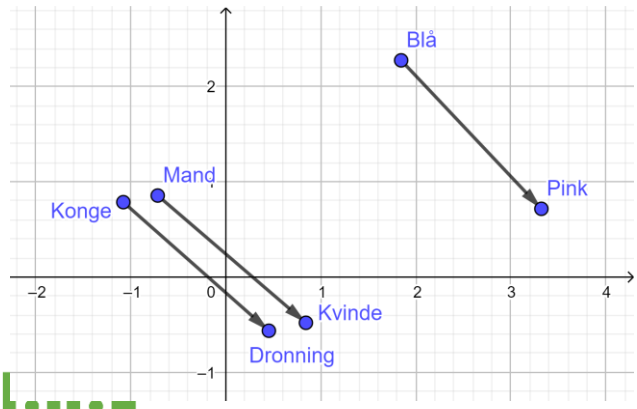
o. Hvilket ord minder mest om *Mand* i eksemplet ovenfor? *Konge, Kvinde* eller *Blå?*

Hjemmesiden vi så tidligere - http://labs.statsbiblioteket.dk/dsc/ - har også mulighed for at lave en sådan søgning. Dette foregår i feltet "Nearest words".

p. Hvilke tre ord er tættest på *Hest* ifølge modellen på siden?
q. Find selv på flere ord at prøve med, og skriv de bedste/sjoveste resultater ned.

Analogier i eksemplet

Vi vender nu tilbage til eksemplet fra før, og ønsker at finde analogier mellem ord. Undersøg ved hjælp af word embeddings om følgende analogier gælder, og kommenter resultaterne:



DUDE

r. ”*Mand* forholder sig til *Kvinde*, som *Konge* forholder sig til *Dronning*”.
s. ”*Mand* forholder sig til *Kvinde*, som *Konge* forholder sig til *Blå*”.
t. ”*Mand* forholder sig til *Kvinde*, som *Blå* forholder sig til *Pink*”.

Bias skaber problemer!

Den sidste analogi ser potentielt problematisk ud! I en moderne, ligestillet verden skulle mænd og kvinder helst ikke identificeres med forskellige farvekoder. Dette er et eksempel på *bias* i modellen, altså en type *forudindtagethed*.

Dette kan synes harmløst når det handler om farver, men det kan gå grueligt galt hvis der er andre biases i modellen:

u. Research historien om Microsofts Twitter-chatbot Tay, og forklar hvad der gik galt.

Hvor kommer bias fra?

Som du måske fandt ud af i historien om Tay, er sprogteknologiske modeller kun så gode som de data man baserer dem på. Hvis det korpus man bruger til at træne modellen er racistisk eller sexistisk vil det også afspejle sig i de word embeddings den producerer. Dette kan illustreres gennem talemåden ”Garbage in – garbage out”.

Problemet kunne derfor løses ved at bruge et andet korpus der ikke indeholder de biases man ønsker at undgå. I praksis kan det dog være svært eller kostbart at finde sådan et korpus. Men der er også andre muligheder som vi skal se i næste afsnit.

De-biasing

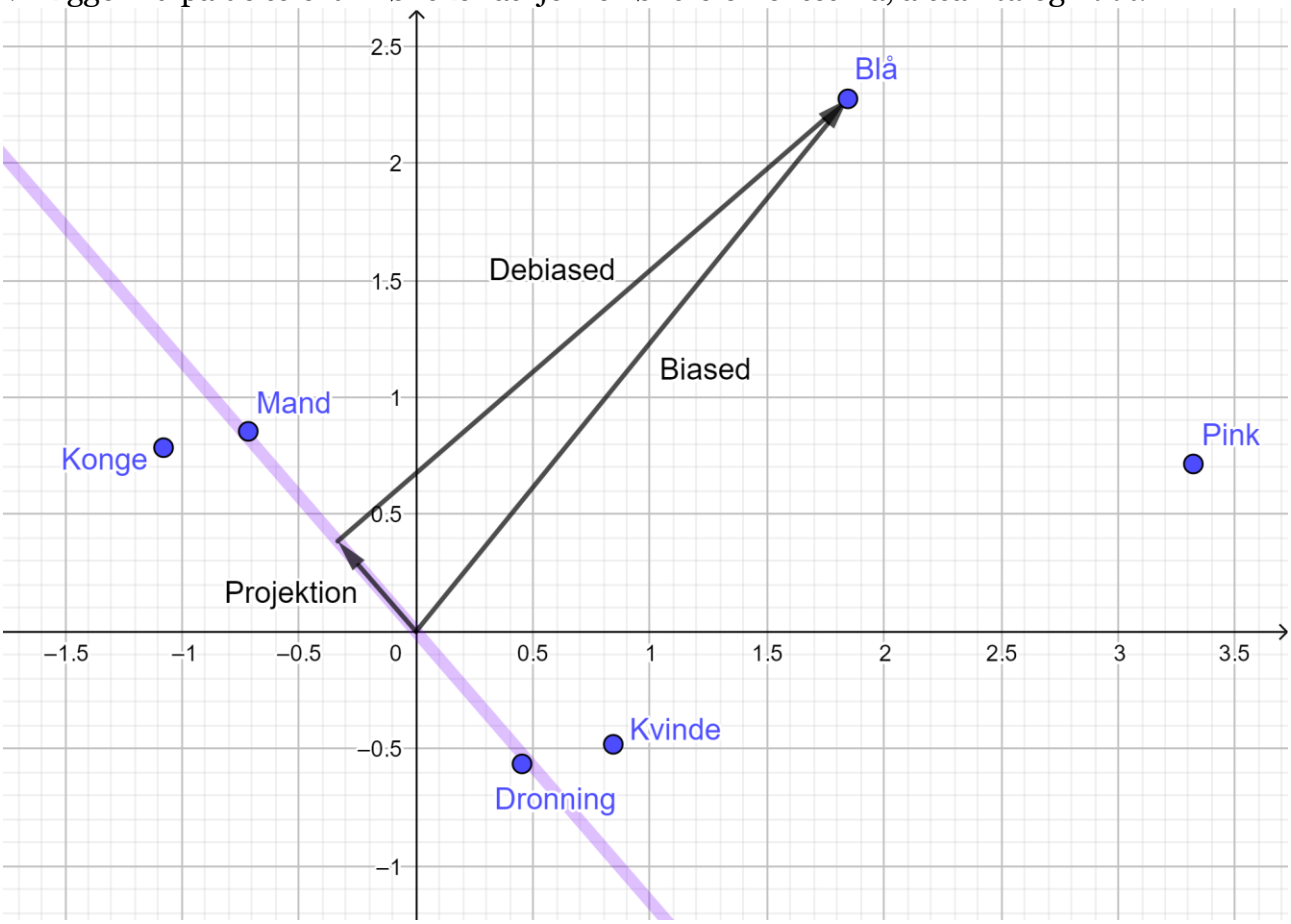
Nu hvor vi ved der er et problem med analogien mellem køn og farve i vores model, kan vi aktivt forsøge at fjerne denne sammenhæng. Denne proces kaldes *de-biasing*, idet vi prøver at reducere bias i modellen.

Først spørger vi os selv: Hvordan udtrykkes køn i modellen? Når vi kigger på de fire ord der har et tydeligt køn ser vi, at de i store træk ligger på en lige linje. I modsætning til vores første legetøjseksempel, hvor køn kunne aflæses på x-aksen er køns-elementet her tilsyneladende udtrykt ved hvor vi befinder os langs denne linje.

v. Find en retningsvektor $\vec{v}_{\text{køn}}$ for denne linje ved at tage gennemsnittet af de fire kønnede ords embeddings. Du skal altså lægge de fire vektorer sammen, og gange resultatet med $\frac{1}{4}$.

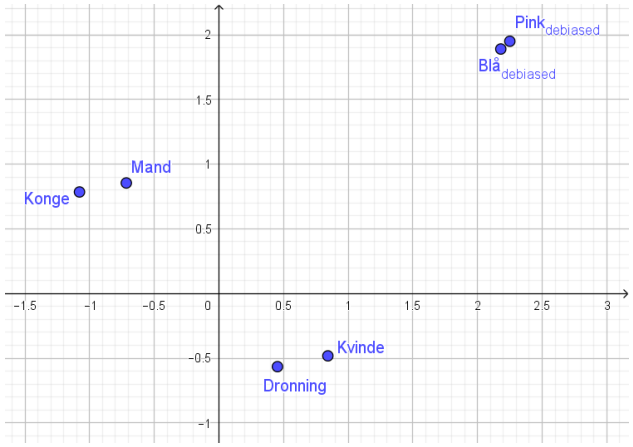
På figuren nedenfor er linjen tegnet med en fed, halvgennemsigtig lilla.

Vi kigger nu på de to ord vi ønsker at fjerne køns-elementet fra, altså *Blå* og *Pink*.



Hvilken del af disse ords embedding er parallel med $\vec{v}_{\text{køn}}$? Svaret er, at det er *projektionen* på $\vec{v}_{\text{køn}}$, som vist på figuren ovenfor.

w. Beregn $\overrightarrow{\text{Blå}}_{\vec{v}_{\text{køn}}}$, altså projektionen af $\overrightarrow{\text{Blå}}$ på $\vec{v}_{\text{køn}}$.



Det er denne del af *Blå*'s embedding vi ønsker at slippe af med. Derfor kan vi finde den *de-biasede* embedding ved at trække projektionen fra:

x. Beregn $\overrightarrow{\text{Blå}}_{\text{debiased}} = \overrightarrow{\text{Blå}} - \overrightarrow{\text{Blå}}_{\vec{v}_{\text{køn}}}$
y. Lav en skitse af den tilsvarende geometri for *Pink*. Beregn den de-biasede embedding $\overrightarrow{\text{Pink}}_{\text{debiased}}$
z. Hvad betyder det, at de to debiasede embeddings nu er tæt på hinanden?
æ. Undersøg analogien "*Mand* forholder sig til *Kvinde*, som *Blå* forholder sig til *Pink*" med de nye, de-biasede embeddings.

Klap dig selv på skulderen ☺ Du er nået til slutningen af projektet ☺