Messy data i praksis: Oprydning i tekst

Social Data Science I

19. oktober 2023

Kristian Gade Kjelmann
kgk@socsci.aau.dk

Inst. for Sociologi og Socialt Arbejde
Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Dagens program

Eksempel med tekst pre-processing og klyngeanalyse (live-coding)

Tekst som data - Udfordringer

Tekst som tal
- Tekst som vektorer
- Tekster som matrix
- Tekster som ordtabel
- Tekster som graf/netværk
- Overvejeser med tekst som data

Fra tekst til datastruktur
- Bag-of-words repræsentation
- Tf-idf vægtning

Vectorizers og tokenizers i Python (live-coding)

Eksempel med tekst pre-processing og klyngeanalyse (live-coding)

AALBORG UNIVERSITY
DENMARK

Tekst som data - Udfordringer

Det skrevne sprog er ikke entydigt!

Jeg elsker politik

Jeg elsker ikke politik

Jeg elsker politik, når folk råber i munden på hinanden

Jeg elsker politik, når folk råber i munden på hinanden. Først da kan jeg mærke, hvor passionerede politikerne er.

Tekst som data - Udfordringer

Det skrevne sprog er ikke entydigt!

Tim valgte kort før Allan.

- Valgte Tim kort i et spil, før Allan valgte kort?
- Valgte Tim et eller andet lige inden, at Allan valgte noget?
- Valgte Tim kort inden han valgte Allan?

Tekst som data - Udfordringer

Tekstdata har altid *høj dimensionalitet* (mange variable)

- Høj dimensionalitet gør det vanskeligere at opsummere data simpelt, og mange modeller har svært ved at håndtere det.
- Høj dimensionalitet bevirker typisk også høj *sparsity* (mange 0-tællinger), som også er en udfordring for mange modeller og metoder.
- Ikke blot ordene i teksten, men også sætningskonstruktion (syntaks) og ordenes betydning (semantik) er nødvendige for at kunne opsummere teksten fyldestgørende
- Derudover kan der være faktorer, som går ud over teksten i sig selv, der kan være relevante for, hvordan den skal forstås (historisk kontekst, forfatter, medie osv.)

Tekst som data - Udfordringer

Tekstdata er vanskelige at standardisere

- Gradbøjning - vælge, vælger, valgte
- Ens stavemåder - en vælger (navneord), han vælger (verbum)
- Alternative stavemåder - ressource, resurse
- Synonymer - stille, sagte
- Forskellig semantisk "vægt" - hvor relevant er ordet for teksten?

Tekst som data - Udfordringer

Tekstdata har ikke en given datastruktur

- Opgjort på ord, sætninger, afsnit?
- Tællinger? Vægtning?
- Sammenlignes tekster, ord, ordforbindelser, kontekst?

Tekst som tal

- Ord og tekster kan repræsenteres numerisk på forskellig vis.
- Computeren har brug for numerisk repræsentation for at kunne foretage beregninger.
- Udfordring med tekst: Høj dimensionalitet og høj *sparsity* ("tomme"observationer).
- Teoretisk udfordring: Hvordan bevarer vi kontekst for et ord?
- Teknisk udfordring: Hvordan reduceres dimensionalitet uden at miste information om ordets kontekst?

  Man anvender termet “text vectorization” til at referere til teknikker, der konverterer tekster til matematiske repræsentationer (datastrukturer, som man kan foretage beregninger på)

Document-term matrix

“En kat spiser en mus”

	en	kat	mus	ost	spiser
0	2	1	1	0	1
1	1	0	1	1	1

- Én række per tekst, én kolonne per unikt ord/lemma/token (“types”).
- Hver række en vektor: Tal der afspejler sammenfaldende ord (co-occurrence).
- Dimensionalitet svarende til antal types (“vocabulary”).
- Høj *sparsity* (mange 0’er) - et problem for mange beregningsmetoder/modeller.

Co-occurrence matrix

"En kat spiser en mus"

	en	kat	mus	ost	spiser
en	0	2	3	1	3
kat	2	0	1	0	1
mus	3	1	0	1	2
ost	1	0	1	0	1
spiser	3	1	2	1	0

- Rækker og kolonner afspejler unikt ord/lemma/token.
- En hel række/kolonne beskriver de ord, som optræder i kontekst af ordet.
- Høj *sparsity* (mange 0'er).

Tekster og datastrukturer

Datastrukturer for tekster ikke givet.

Datastruktur afhænger både af, hvordan tekst er konverteret til tokens samt af, hvordan ordene skal opgøres (tælles, vægtes eller andet).

Lad os se på forskellige måder, at repræsentere nedenstående sætninger som datastrukturer:

'Vi er utroligt beærede'

'Vi vil gerne dele prisen med alle'

'Det skal vi have gjort op med.'

Tekster som matrix (document-term matrix)

- En række per tekst, en kolonne per ord, en celle per tælling
- (hvis kolonner som ord og rækker som tekst: term-document matrix)

	alle	beærede	dele	det	er	gerne	gjort	have	med	op	prisen	skal	utroligt	vi
0	0	1	0	0	1	0	0	0	0	0	0	0	1	1
1	1	0	1	0	0	1	0	0	1	0	1	0	0	1
2	0	0	0	1	0	0	1	1	1	1	0	1	0	1

Tekster som ordtabel

Tekster og datastrukturer

ID	Word	POS	Order	ID
0	Vi	PRON	1	1
1	er	AUX	2	1
2	utroligt	ADV	3	1
3	beærede	VERB	4	1
4	Vi	PRON	1	2
5	vil	AUX	2	2
6	gerne	ADV	3	2
7	dele	VERB	4	2
8	prisen	NOUN	5	2
9	med	ADP	6	2

Tekster som graf/netværk

Tekster og datastrukturer

- Hver række udgør en ordforbindelse; kolonne for hver "ende" af forbindelsen

ID	From	To
0	Vi	er
0	Vi	utroligt
0	Vi	beærede
0	er	Vi
0	er	utroligt
0	er	beærede
0	utroligt	Vi
0	utroligt	er
0	utroligt	beærede
0	beærede	Vi

Tekst som data - overvejelser

Tekster og datastrukturer

1. Hvordan skal tekst *bearbejdes*? (standardisering, sortering, udvælgelse)
2. Hvordan skal tekst *repræsenteres*? (datastruktur)
3. Hvordan skal tekst *analyseres*? (anvendte teknikker, metoder, modeller)

2 og 3 hænger ofte sammen (repræsentation afhænger af analysen, som skal foretages).

Fra tekst til analyseenheder
Fra tekst til datastruktur

Politiet har givet borgerne råd til,

hvordan man kan forsøge at jage prærieulvene på flugt

med larm, vandpistoler og peberspray.

Bag-of-words repræsentation

Fra tekst til datastruktur

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

15

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Bag-of-words repræsentation

Fra tekst til datastruktur

Mange modeller bygger på “bag-of-words” repræsentationer af tekst (bag-of-words modeller).

Optales bag-of-words, da denne struktur ikke tager højde for sætningskonstruktion (syntaks) eller semantik.

Eksempel på bag-of-words struktur: Document-term matrix

ID	alle	beærede	dele	det	er	gerne	gjort	have	med	op	prisen	skal	utroligt	vi
0	0	1	0	0	1	0	0	0	0	0	0	0	1	1
1	1	0	1	0	0	1	0	0	1	0	1	0	0	1
2	0	0	0	1	0	0	1	1	1	1	0	1	0	1

Bag-of-words repræsentation
Fra tekst til datastruktur

Udfordringer med bag-of-words repræsentation

- Tager ikke højde for sætningskonstruktion og semantik
- Giver ofte datastrukturer med høj "sparsity" (mange tomme celler).

Ordtællinger

Fra tekst til datastruktur

Simpleste måde at opgøre ord (eller tokens) i tekst: hvor mange gange er ordet nævnt i teksten?

ID	am	and	anywhere	do	eggs	green	ham	here	like	not	or	sam	so	thank
0	1	2	0	3	2	2	2	0	3	2	0	1	0	0
1	0	0	1	0	0	0	0	2	3	2	2	0	0	0
2	1	1	0	1	1	1	1	0	1	0	0	1	1	2

Binær tælling
Fra tekst til datastruktur

Binær tælling: 1 hvis term indgår i tekst, ellers 0.

ID	am	and	anywhere	do	eggs	green	ham	here	like	not	or	sam	so	thank
0	1	1	0	1	1	1	1	0	1	1	0	1	0	0
1	0	0	1	0	0	0	0	1	1	1	1	0	0	0
2	1	1	0	1	1	1	1	0	1	0	0	1	1	1

Tf-idf vægtning af ord

Fra tekst til datastruktur

En udbredt måde at opgør ord/tokens er med tf-idf: *term frequency-inverse document frequency*.

Dette mål udregner, hvor ofte ordet fremgår i en tekst set i forhold til, hvor mange tekster ordet fremgår i.

I udregningen vægtes ord, som fremgår i få tekster, op. Dette ud fra en antagelse om, at ord, der fremgår i få tekster, er mere sigende for indholdet af de tekster, som ordet indgår i.

Tf-idf vægtning af ord

Fra tekst til datastruktur

tf-idf udregnes som:

$tfidf = tf(t, d) * idf(t, D)$

hvor:

$tf(t, d)$: antal gange term t er nævnt i dokument d

$idf(t, D) = log(\frac{D}{|\{d \in D : t \in d\}|})$: logaritmen af antal dokumenter i alt (D) divideret med antal dokumenter, der indeholder termet ($\{d \in D : t \in d\}$)

Obs: Der kan være variationer i, hvordan tf og idf udregnes samt forskelle i brug af standardisering mellem funktioner

Tf-idf vægtning af ord

Fra tekst til datastruktur

ID	am	and	anywhere	do	eggs	green	ham	here	like	not	or	sam	so
0	0.16	0.32	0.00	0.47	0.32	0.32	0.32	0.00	0.38	0.32	0.00	0.16	0.00
1	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.35	0.31	0.27	0.35	0.00	0.00
2	0.23	0.23	0.00	0.23	0.23	0.23	0.23	0.00	0.18	0.00	0.00	0.23	0.30

Vectorizers tokenizers i Python (live-coding)

AALBORG UNIVERSITY
DENMARK