Kirstine Boas, Inger Schoonderbeek Hansen, Tina Thode Hougaard
& Ea Lindhardt Overgaard (udg.):
19. Møde om Udforskningen af Dansk Sprog, Århus 2022

1500-talssalmer og danske sprogmodeller

Ea Lindhardt Overgaard
Aarhus Universitet

Indledning

Danske sprogmodeller til *Natural Language Processing* (NLP) er i de seneste år blevet forbedret i takt med at der er udviklet flere ressourcer inden for dansk sprogteknologi. Det viser sig blandt andet ved større og bedre korpora samt nye forbedrede sprogmodeller (Enevoldsen, Johansen m.fl 2021, Strømberg-Derczynski, Ciosici m.fl. 2021, sprogteknologi.dk 2023). I takt med udviklingen af danskprogede ressourcer er opmærksomheden på sprogmodellernes potentiale og inddragelsen af digitale værktøjer i undervisnings- og forskningsmiljøer også steget (Ondelli 2018). Trods den positive udvikling for danske sprogmodeller er der stadig et stykke vej endnu før modellerne uden videre kan benyttes på domænespecifikke områder, f.eks. salmelitteratur, og historiske datasæt, med tilfredsstillende resultater (Enevoldsen, Johansen m.fl. 2021). Det er et velkendt problem at sprogmodellerne præsterer dårligere på sådanne data fordi sprogmodellerne er udviklet og trænet på moderne dansk og hovedsageligt prosaisk tekstmateriale. Det er altså en udfordring at få sprogmodellerne til at opnå en høj nok grad af præcision når de benyttes på tekstsamlinger der afviger en moderne retskrivningsnorm.

For at undersøge hvordan man kan forbedre de danske sprogmodellers præcision på normafvigende korpora, præsenterer jeg i denne artikel forskellige strategier for at optimere sprogmodellernes præstation gennem normalisering af en lille samling danske reformationssalmer. Med afsæt i ti salmer[1] fra Thomissøns salmebog (1569) undersøger jeg hvorvidt normalisering af teksterne påvirker NLP-værktøjers præstation. Til at illustrere modellernes præstation tester jeg præcisionen af *Part of speech-tagging* (POS) for de to danske sprogmodeller *DaCy* og *SpaCy*, der begge er frit tilgængelige værktøjer til *Natural Language Processing* i programmeringssproget *Python* (Honnibal og Montani 2017, Enevoldsen, Johansen m.fl. 2021). Sprogmodellernes præstation testes på tre forskellige grader af normalisering af salmeteksterne. Denne undersøgelse indeholder kun 10 salmer, men hvis tallet steg bare til 100 sal-

[1] Titler i moderniseret form: *Aleneste Gud i Himmerig*; *Behold os Herre, ved dit ord*; *Det hellige kors, vor Herre selv bar*; *Et barn er født i Betlehem*; *Et lidet barn så lysteligt*; *Hvad kan os komme til for nød*; *Jeg vil mig Herren love*; *Krist stod op af døde*; *O Guds Lam Uskyldig* og *Vor Gud han er så fast en borg*.

mer, ville en manuel annotering være enormt ressourcekrævende. Af samme grund er det ofte nødvendigt at benytte digitale sprogmodeller når man arbejder med *big data*-analyser. Artiklen fungerer derfor som et eksempel på hvordan man kan håndtere normaliseringen og optimere brugen af sprogmodeller på tekster i meget større korpora.

I artiklen redegør jeg først for undersøgelsens overordnede design. Herefter vil jeg gennem et eksempel med POS-tagging argumentere for nødvendigheden af at normalisere sine tekstdata når man benytter sprogmodeller på ældre tekst. Efter eksemplet forklarer jeg hvordan undersøgelsens ti salmer er blevet normaliseret, og hvordan præcisionen af sprogmodellernes POS-tagging evalueres. Slutteligt diskuterer jeg normaliseringens indvirkning på sprogmodellernes POS-tagging af salmerne.

Undersøgelsens design

Undersøgelsens data består af digitale udgaver af de ti ovennævnte salmer fra Thomissøns salmebog 1569. Salmeteksterne er digitaliseret til et rent tekstformat (.txt-filer). I undersøgelsen bruger jeg programmeringssproget *Python* (Rossum og Drake 2009) hvor jeg, foruden de maskinlæringsbaserede sprogmodeller *SpaCy* og *DaCy,* benytter mig af basismoduler og basisfunktionalitet indbygget i *Python*. Jeg bruger særligt datatypen *dictionaries* til at opbygge en ordbogsressource for salmeteksterne. Her knytter jeg et opslagsord, *key*, med en eller flere variationsformer for ordet, betegnet som *values*. Det er muligt at knytte flere *values* til hver enkelt *key*. Nedenfor ses i eksempel (1) en dictionary med key *vederkvæger* og den tilhørende value *vederqueger*, samt i eksempel (2) en dictionary med key *hand* og de tilhørende values *han* og *hånd*.

```
(1)   Ordbog_1 = {'vederkvæger': ['vederqueger']}
(2)   Ordbog_2 = {'hand': ['han', 'hånd']}
```

Ud fra et opslagsord får jeg altså automatisk præsenteret dets mulige variationsformer, præcis som hvis jeg slog ordet op i en almindelig ordbog. Ordbøgerne fungerer begge veje så jeg i ovenstående eksempel kan udtrække både *vederkvæger* ved at søge *vederqueger* frem, og omvendt. For ord der i salmerne har en ældre stavemåde end den moderne retstavning foreskriver, oprettes indgangene *key: value* hvor den ældre stavemåde knyttes til en stavemåde der følger moderne retstavning. Ordbogen er på den måde et redskab til at normalisere de ældre salmetekster til en mere moderne retstavning. De moderne retstavningsformer er hovedsageligt baseret på Dansk Sprog- og Litteraturselskabs omfattende ordbogsressource knyttet til danske reformationssalmer. Ressourcen giver adgang til en lang række leksikografiske værker på én

gang. For ét opslag præsenteres man for opslagsartikler der matcher det søgte ord på tværs af forskellige ordbogsressourcer, og opslagene indeholder links til de originale opslagsartikler i de enkelte ordbøger (Det Danske Sprog- og Litteraturselskab 2023).

Normaliseringen af ordene i de ældre salmer er i mange tilfælde ikke entydigt bestemt. For ord med flere mulige *values* knyttet til hver *key* kræver det en manuel vurdering af hvilket ord der er det rette i den givne kontekst. Jeg har derfor opdelt normaliseringsprocessen i undersøgelsen i tre dele for at tage forbehold for de ord der kræver en manuel vurdering. I afsnittet *Normalisering af tekstdata* uddyber jeg denne opdeling samt hvordan de forskellige *dictionaries* bruges.

Som nævnt i indledningen tester jeg efter hver del af normaliseringsprocessen præcisionen af de danske sprogmodellers POS-tagging. Ved POS-tagging annoteres hvert enkelt ord i en tekst med et *part of speech*-tag, der angiver ordklassen for det givne ord. Ordklassenotationen følger de universelle POS-tags (Universal Dependencies (UD) 2014-2022). Forud for normaliseringsprocessen har jeg testet præcisionen for POS-tagging på de originale tekster. Til at evaluere præcisionen har jeg lavet en manuel POS-opmærkning for salmeteksterne, en såkaldt *golden standard*, som kan holdes op mod sprogmodellernes resultater. Dette er standard evalueringspraksis (Enevoldsen, Johansen m.fk. 2021). Præcisionen af sprogmodellerne måles ud fra hvor mange procent af POS-taggene sprogmodellerne har angivet korrekt.

Er normalisering nødvendig?

For at sprogmodellerne kan præstere godt nok på små domænespecifikke korpora til at man kan opnå brugbare resultater, er det nødvendigt at tilpasse sit korpus så det ligner modellens træningsdata mest muligt. Det er som nævnt et velkendt problem inden for sprogteknologi at sprogmodellernes præcision er for lav hvis man anvender dem på korpora der ikke følger moderne retskrivningsnormer (Bigi 2014). For at illustrere problematikken viser Tabel 1 et udsnit af en automatiseret POS-opmærkning af den originale version af salmen *Alleniste Gud i Himmerig* fra Thomissøns salmebog (1569) og af den moderne salme *Aleneste Gud i Himmerig* fra Den Danske salmebog (2003)[2]. Fejlklassificerede POS-tags er markeret med grå, og POS-taggingen er lavet med *SpaCy*:

[2] En liste over de forskellige POS-tags findes til sidst i artiklen.

Tabel 1

Thomissøn 1569	POS	DDS 2002	POS
Alleniste	Propn	Aleneste	Propn
Gud	Noun	Gud	Noun
i	Adp	i	Adp
Himmerig	Noun	Himmerig	Noun
være	Aux	ske	Verb
loff	Adj	lov	Noun
oc	X	og	Konj
priss	Noun	pris	Noun
for	Adp	for	Adp
alle	Adj	sin	Det
sin	Det	nåde	Noun
naade	Noun	som	Pron
der	Pron	han	Pron
hand	X	har	Aux
haffuer	Propn	os	Pron
giort	Verb	skænket	Verb
i	Adp	faderlig	Adj
Jorderig	Noun	at	Part
J	Propn	fri	Adj
disse	Det	os	Pron
samme	Adj	af	Adp
naadelige	Adj	syndens	Noun
dage.	noun	våde	Adj

I skemaet kan man se at der forekommer flere fejlklassificeringer for det ældre teksteksempel end for eksemplet med moderne retstavning. Hvis man kigger på den procentvise fejlklassificering for salmeteksterne i deres fulde længde, har *Alleniste Gud i Himmerig* 44,8% forkerte POS-tags mens *Aleneste Gud i Himmerig* har 11,7%. Der er altså en tydelig forskel på sprogmodellernes præstation fra det ældre teksteksempel til det nyere. Det skal bemærkes at det moderne teksteksempel ikke er en normalisering af den ældre salme, men er den moderne udgave som den fremgår i den nugældende salmebog. Den moderne salme har altså, foruden at være tilpasset gældende retskrivningsnormer, været udsat for flere gendigtninger og revisioner gennem tiden. Jeg benytter begrebet *normalisering* om dét at tilpasse de enkelte ord til en gældende norm – i dette tilfælde retskrivningsnormen – i det omfang det er muligt uden at foretage drastiske indholdsmæssige ændringer ift. den originale tekst.

For salmen *Alleniste Gud i Himmerig* er fejltyperne på de forkerte POS-tags varierende mellem proprier, adjektiver og X (andet). Disse fejltyper er hyppige gennem hele salmen, men ikke begrænset til disse ordklasser da der forekommer fejlklassificeringer for samtlige ordklas-

ser. Den hyppigste fejltype er dog at SpaCy klassificerer noget som et proprium der ikke er det. Dette sker 30 gange bare i denne salme og udgør godt en tredjedel af alle fejlklassificeringerne.

I salmen *Aleneste Gud i Himmerig* ses samme fejltype, som i det ældre eksempel, hvor SpaCy klassificerer noget som et proprium der ikke er det. Ellers er den hyppigste fejltype her at ord klassificeres som substantiver og adjektiver når de ikke er det. Her er det særligt imperativformer af verber, som *fri* i eksemplet fra skemaet, samt interjektioner der bliver fejlklassificeret. Ordene *fri* og *våde* er begge homografer med et adjektiv hvor man kan forestille sig at den adjektiviske udgave er den hyppigste variant af ordet i normaliseret brug. Fejlklassificeringen af disse ord i eksemplet er her højst sandsynligt baseret på at sprogmodellens træningsdata har en højere frekvens af *fri* og *våde* som adjektiver end som verbum og substantiv.

Selvom der forekommer fejl i begge ovenstående eksempler, så er fejlene meget hyppigere for den ældre tekst end for den moderne. Fejlklassificeringerne synes oftest kategoriske for den moderne salme hvor de for den ældre salme både består af kategoriske fejl, men også har en høj grad af tilfældighed. Selvom fejlmarginen er betydeligt lavere for den moderne salme end for den ældre, så ligger mængden af fejlklassificeringer for den moderne salme på grænsen af hvad man kan acceptere som brugbart. De bedste danske sprogmodeller kan efterhånden klassificere med over 90 % nøjagtighed, og dette er inden for digital humaniora et præcisionsniveau der normalt accepteres (Enevoldsen, Johansen m.fl. 2021). En måde at reducere fejlmarginen for POS-taggingen af de ældre salmer er at normalisere salmeteksterne i en sådan grad at ordenes ortografi ligner det data, som sprogmodellen er trænet på, så meget som muligt. Dette behandles i følgende afsnit.

Normalisering af tekstdata

I arbejdet med tekst som data er man nødt til at klargøre sit datamateriale på flere niveauer for at gøre teksten maskinlæsbar (Bigi 2014, Ondelli 2018). Et af disse niveauer omhandler hvordan man skal opdele teksten i tokens, dvs. i enheder af ord, tegnsætning, mellemrum - små betydningsenheder man gerne vil kunne identificere maskinelt. I denne proces ensretter og normaliserer man ofte sit korpus så eksempelvis tal altid skrives ud i bogstaver, forkortelser skrives i deres fulde længde mm. Dette gør disse størrelser let genkendelige i korpus. I denne undersøgelse drejer spørgsmålet om normalisering sig om hvordan ældre ordformer og varianter gøres genkendelige for moderne sprogmodeller. Normaliseringsaspektet rejser for salmerne i Thomissøns salmebog flere filologiske spørgsmål, herunder hvor meget og hvordan teksterne skal normaliseres uden at normaliseringen skaber indholdsmæssige uoverensstemmelser mellem den originale og den normaliserede udgave af en salmetekst. Jeg ønsker at bringe

salmeteksterne så meget i overensstemmelse med den gældende retskrivningsnorm som muligt uden at ændre indholdsmæssigt på salmeteksterne. Hvis man eksempelvis vil undersøge forhold omkring et ord som *navn*, kan man så uproblematisk slå formerne *Naffn, Navn,* og *navn* sammen under én form: *navn*? Det er langt fra altid muligt at reducere et ords varianter til én form. Ord, der ikke entydigt henviser til samme lemma, er eksempelvis problematiske: Vil et ord som *ere* eksempelvis kunne udbyttes med *ære* eller *er*? Mange ord har flere mulige normaliserings-muligheder, og man skal vurdere normaliseringen for disse ord i dets kontekst. I det følgende viser jeg et eksempel på hvordan de forskellige normaliseringsaspekter kan behandles.

Eksempel på normalisering af salmetekster

Normaliseringen af de ældre salmer er foretaget i tre trin. Første trin af processen er automatisk at ensrette de ord der uden problemer kan reduceres fra mange variantformer til én. Første trin ensretter altså ortografien for ord der gennem tiden har ændret stavemåder og samtidig ikke har sammenfald i stavemåde med andre ord. Eksempler herpå kunne være *nåde* > *naade* og *vilje* > *villie*. Denne ortografiske normaliseringsproces er automatiseret ved at jeg har lavet en *dictionary* hvor variationsformerne for et ord er knyttet sammen med den form af ordet som jeg vil normalisere med. Jeg har herefter lavet en funktion der for hver forekomst af variationsformerne i salmerne udskifter ordet med dens normaliserede form. Nedenfor i eksempel (3) ses eksempler på opslag i denne ordbog hvor den ønskede normaliserede form af ordet står først og den ældre variant af ordet følger efter:

(3)

```
ordbog = {'lignes': ['lignis'],
          'gjort': ['giort'],
          'nådelig': ['naadelig'],
          'yndest': ['yndist'],
          'vilje': ['villie'],
          'vederkvæger': ['vederqueger']}
```

I andet trin af normaliseringen har jeg lavet en funktion hvor homografer der senere har fået to forskellige stavemåder, identificeres i teksten, og den korrekte form for ordet udvælges manuelt ved at vælge mellem en række mulige normaliseringsmuligheder. Koden for funktionen er vist i nedenstående billede:

```
def moderne_2(tekst): #ordsplittet liste
    tekst_lav = tekst.lower()
    ordopdelt = rens_ord(tekst_lav)
    moderniseret = []
    for token in ordopdelt:
        val = 0
        for key, list_of_values in ordbog_2.items():

            if token == key:
                print('Opslag: ', key)
                print('list_of_values: ', list_of_values)
                valg = int(input('Skriv nummer på korrekt opslag (start ved indeks 0).'))

                moderniseret.append(list_of_values[valg])
                val = 1
                continue

        if val == 0:
            moderniseret.append(token)
            continue

    return " ".join(moderniseret)

moderne_2(tekst)
```

```
Opslag:  loff
list_of_values:  ['lov', 'løv', 'lo', 'løb', 'loft']

Skriv nummer på korrekt opslag (start ved indeks 0).
```

Her ses det at funktionen har identificeret *loff* som et ord der skal tages stilling til i sætningen "være loff oc priss for alle sin naade". Efter opslagsordet følger en *list_of_values* med en række ord der kan vælges imellem. I den sorte bjælke kan man skrive indeks for det ord man ønsker at udskifte *loff* med. Herefter retter funktionen automatisk ordet i salmeteksten til den valgte form. Dette trin er altså en semi-automatiseret proces der giver mulighed for en menneskelig vurdering af ordets betydning i dets kontekst.

Det sidste trin af normaliseringen omhandler brugen af versaler. Jeg har valgt at inddrage dette aspekt i normaliseringen af salmeteksterne da jeg fra eksemplet i Tabel 1 ved at mange ord bliver fejlklassificeret som proprier. Det sker ofte de steder hvor ord starter med versal hvor vi efter moderne retskrivning ikke ville bruge versaler. I 1569 herskede ingen officiel dansk retskrivning (Jacobsen 2010, Det Danske Sprog- og Litteraturselskab 2023), og brugen af versaler for substantiver benyttes inkonsekvent i Thomissøns salmebog. Den anderledes brug af versaler ift. den moderne retskrivningsnorm synes at påvirke sprogmodellernes præcision af POS-tagging i en sådan grad at det er væsentligt at inddrage dette aspekt i normaliseringen for at reducere fejlmarginen. Salmetekster har desuden en særlig brug af versaler da ord som *Gud, Herre* og *Himmerig* starter med versaler og kan anses som proprier i salmekonteksten. Jeg har derfor lavet en vurdering af hvilke ord der i salmekontekst skal bevare startversalerne ud fra et diakront blik på salmelitteraturen. Jeg bevarer derfor versalbrugen ved ord hvor denne versalbrug gør sig gældende gennem hele salmehistorien. Til dette tredje trin af normaliseringen har

jeg igen lavet en funktion der sørger for at bevare versaler for de ord jeg har gemt som versal-bevarende ord.

Nedenfor vises et eksempel fra salmen "Det Hellige Kaarss" fra Thomissøns salmebog (1569), og de forskellige trin teksten har gennemgået i normaliseringsprocessen. Først vises den originale tekst a). Herefter er ændringerne for 1. modernisering markeret med fed i b), 2. modernisering er markeret med understregning i c) og 3. modernisering er markeret med gennemstregning i d):

(4)

a) Det hellige Kaarss vor Herre selff bar, met blodige vunder oc dødelig Saar, for oss hand plict oc bod fuldgiorde, at wi skuld oss der til forlade, wi vorder ej salig i andre maade.

b) Det hellige Kaarss vor Herre ***selv*** *bar, met blodige vunder oc dødelig* ***sår****, for oss hand plict oc bod* ***fuldgjorde****, at wi* ***skulle*** *oss der til forlade, wi vorder ej salig i andre* ***måde****.*

c) Det hellige kors vor Herre ***selv*** *bar, med blodige vunder og dødelig* ***sår****, for os han pligt og bod* ***fuldgjorde****, at vi* ***skulle*** *os der til forlade, vi vorder ej salig i andre* ***måde****.*

d) ~~*d*~~*et hellige kors vor* ~~*H*~~*erre* ***selv*** *bar, med blodige vunder og dødelig* ***sår****, for os han pligt og bod* ***fuldgjorde****, at vi* ***skulle*** *os der til forlade, vi vorder ej salig i andre* ***måde****.*

Det er som nævnt den moderne retstavning der har styret graden af normalisering, samtidig med at der tages forbehold for at ændringerne har så lidt indflydelse på den oprindelige tekst som muligt. Syntaktiske mønstre er ikke ændret da moderne salmer også har en vekslende syntaks. Ældre ordformer (*vunder*) er heller ikke ændret til mere moderne synonymer (*sår*) da sådanne ændringer ville lægge sig tæt op ad nyfortolkninger af salmen hvilket i højere grad ændrer betydningen af den oprindelige tekst. Teksten er gennem normaliseringsprocessen blevet forberedt til NLP-værktøjer der er trænet på moderne dansksproget materiale.

Resultater

De ti salmer i denne undersøgelse har alle gennemgået de tre niveauer af normalisering. Jeg evaluerer sprogmodellerne på alle de tre niveauer af normalisering ved at teste modellernes klassificeringsevner mod en manuel POS-opmærkning af salmerne. Forud for moderniserings-

processen har jeg målt den procentvise del af korrekte POS-klassificeringer, og efter hver del af normaliseringsprocessen har jeg igen målt den procentvise andel af korrekte klassificeringer. Det er på den måde muligt at vurdere det enkelte moderniseringstrin op mod sprogmodellens evne til at klassificere korrekt. I undersøgelsen testes både sprogmodellen SpaCy der, blandt mange mulige sprog, indeholder en model til dansk sprog (Honnibal og Montani 2017), og DaCy, en sprogmodel baseret på SpaCy som er blevet forbedret til dansksproget materiale. Blandt andet er DaCy trænet på en mængde data der kan mime slå- og stavefejl i tekst samt historisk ortografisk variation (Enevoldsen, Johansen m.fl. 2021). DaCy kan derfor antages at være mere robust over for variationer i korpora end SpaCy.

Nedenfor ses *Tabel 2* med resultater for undersøgelsen af hhv. spaCy og DaCy's POS-tagger. Der er for hver salme angivet de procentvise korrekte klassificeringer for hhv. spaCy og DaCy til hver moderniseringsgrad af salmerne.

Tabel 2

Normaliseringsgrad	Original		1. normalisering		2. normalisering		3. normalisering	
Sprogmodul	spaCy	DaCy	spaCy	DaCy	spaCy	DaCy	spaCy	DaCy
Aleneste Gud	56,7	72,4	66,2	80,0	86,7	89,6	90,5	88,6
Behold os Herre	66,9	83,5	73,6	86,0	82,6	90,9	84,3	90,9
Det hellige kors	50,7	75,4	63,0	83,3	84,1	84,8	84,8	86,6
Et barn er født i Betlehem	53,5	58,0	64,3	74,5	80,3	84,1	84,7	86,6
Et lidet barn så lysteligt	46,4	63,1	51,4	70,3	74,8	77,5	74,8	79,7
Hvad kan os komme til for nød	57,1	75,8	67,1	83,9	84,5	89,7	84,5	89,7
Jeg vil mig Herren love	59,0	71,9	75,2	81,3	84,1	87,6	84,5	87,8
Krist stod op af døde	40,0	72,0	36,0	72,0	68,0	64,0	72,0	72,0
O Guds lam uskyldig	33,3	69,8	57,3	78,1	71,9	83,3	75,0	83,3
Vor Gud han er så fast en borg	46,8	68,9	61,2	81,1	81,6	87,9	83,2	88,4

For de originale salmetekster ligger andelen af korrekt klassificerede ord på mellem 33,3% og 66,9% for SpaCy og på mellem 58,0% og 83,5% for DaCy. Trods store udsving i begge modellers succesrater kan det konkluderes at DaCy præsterer væsentligt bedre på den originale tekst end SpaCy. Ingen af modellerne nærmer sig dog et niveau hvor opmærkningen vil kunne benyttes i et videre analytisk arbejde. Det er også værd at bemærke at de laveste succesrater for

SpaCy ikke nødvendigvis medfører en tilsvarende lav succesrate for DaCy. Det kan blandt andet ses ved "O Guds lam uskyldig", der har succesraterne 33,3% og 69,8%. Samtidig præsterer de to modeller næsten på samme niveau for salmen "Et barn er født i Betlehem" med succesrater på 53,5% og 58,0%.

I tabellen kan man se at sprogmodellerne præsterer bedre i takt med at salmeteksterne moderniseres. Dette gælder i alle tilfælde bortset fra "Krist stod op af døde" hvor både SpaCy og DaCy falder i succesrate fra hhv. originalteksten til udgaven med 1. modernisering for SpaCy og fra 1. modernisering til 2. modernisering med DaCy. Dette resultat skyldes formentlig at salmen kun indeholder 25 ord og én korrekt klassificering fra eller til derfor påvirker den procentvise succesrate med 4 %.

Den gennemsnitlige andel af korrekt klassificerede ord til de forskellige normaliseringsgrader ses i Tabel 3:

Tabel 3

Normaliseringsgrad	Original		1. normalisering		2. normalisering		3. normalisering	
Sprogmodul	spaCy	DaCy	spaCy	DaCy	spaCy	DaCy	spaCy	DaCy
Gennemsnit	51,04	71,08	61,53	79,05	79,86	83,94	81,83	85,36

Her kan man se at SpaCy præsterer langt dårligere end DaCy jo lavere normaliseringsgrad af teksten man evaluerer modellen på. Dog sker der væsentlige forbedringer ved højere normaliseringsgrad og særligt 1. til 2. normaliseringsgrad har betydning for SpaCys evne til at klassificere korrekt. En grund til DaCys markant bedre evne til at klassificere korrekt for de lavere normaliseringsgrader kan skyldes at modellen netop er trænet på data, der gør modellen mere robust for mindre ortografiske forskelle.

Både 1. og 2. grad af normalisering har stor indflydelse på succesraten for sprogmodellerne hvor den 3. normaliseringsgrad påvirker resultatet i mindre grad. Det skyldes blandt andet at proprier ikke er en så udbredt ordklasse sammenlignet med andre ordklasser og en 100 % korrekt klassificering af proprier derfor stadig kun vil bidrage med en forholdsvis lav procentvis forbedring i den samlede succesrate af ordklassificeringer.

De normaliseringstrin der er benyttet i denne undersøgelse, har gennemsnitligt forbedret andelen af korrekt klassificerede ord fra 51,04 % til 81,83 % for SpaCy og fra 71,08 % til 85,36 % for DaCy. Det er, trods den store forbedring, stadig ikke nok til at komme over de gyldne 90 % som inden for digital humaniora kan antages som den accepterede grænse. Normaliseringen medvirker dog til at man når et langt stykke af vejen til at nå målet ved blot at normalisere

teksterne. Det diskuteres i følgende afsnit hvordan man videre vil kunne tilpasse sprogmodellerne for at nå et acceptabelt klassificeringsniveau.

Diskussion

Jeg har gennem artiklen præsenteret forskellige normaliseringsgrader for ældre salmetekster og evalueret to danske sprogmodellers succesrate for POS-tagging på hvert niveau af normaliseringsgraderne. Resultaterne viser at normalisering af ældre salmetekst forbedrer sprogmodellernes evne til POS-tagging væsentligt. Samtidig kan det konkluderes at normalisering af ældre salmetekst alene ikke kan medvirke til at sprogmodellernes procentvise succesrate kan nå et højt nok niveau til at de lever op til en tilfredsstillende standard. Ud fra denne mindre undersøgelse tyder det på at modellen DaCy virker mere robust for lavere normaliseringsgrader af dansksproget tekst end modellen SpaCy. Dette forhold udligner sig jo højere grad af normalisering teksten har gennemgået.

Selvom normalisering og ensretning af tekst altid er et spørgsmål af relevans i tilrettelæggelsen af et korpus, så skal man altid forholde sig til konsekvenserne ved en sådan ensretning af tekstkorpus. Er alle dele nødvendige? Og hvorfor? Og desuden, hvilke konsekvenser medfører det at modernisere versus hvis man ikke gør det? Det vil under alle omstændigheder altid være en konsekvens at videre undersøgelser af tekstdata i et korpus baserer sig på den måde teksten er repræsenteret i korpusset.

Domænespecifikke karakteristika i tekstkorpora kan også påvirke succesraten for sprogmodellernes klassificering. I en salmekontekst drejer det sig blandt andet om den særlige brug af versaler, der er blevet kommenteret tidligere. Desuden medvirker salmernes strofiske form og sangbarhed til at den syntaktiske struktur er afvigende fra en 'normal' sætningssyntaks, brugen af tegnsætning er friere end retskrivningens normer foreskriver – punktum afgrænser ikke nødvendigvis sætninger, og kommatering indikerer nogle gange linjeskift frem for syntaktisk placerede kommaer. Salmer har også en særlig brug af interjektioner og imperativer. Frekvensen af disse ordklasser i salmernes tekstdomæne kan altså variere meget fra den frekvens ordklasserne har i det træningsdatasæt som sprogmodellerne er trænet på, da salmer eller lignende tekstdata ikke er repræsenteret i tilstrækkelig grad i disse datasæt (Enevoldsen, Johansen m.fl. 2021).

En løsning på de domænespecifikke karakteristikas indvirken på sprogmodellernes klassificering af tekstdata kan være at træne sprogmodellerne yderligere på en mængde salmer der er manuelt annoterede med korrekte klassificeringer. Det er imidlertid et ressourcekrævende og omstændigt arbejde. Der er desuden også en begrænset mængde salmedata tilgængelig, og

det ville give dårligt mening at annotere en tilstrækkelig mængde salmedata til at optræne en model hvis der ikke ligger en tilsvarende stor mængde salmer tilbage som man kan benytte modellen på efterfølgende.

En måde at optimere sprogpakkerne til salmerne er at man manuelt går ind og retter sprogværktøjerne for specifikke situationer. Hyppige årsager til fejlklassificeringer i salmerne forekommer ved interjektionerne *O* og *Amen*. Disse optræder forholdsvis hyppigt i salmerne, og det vil være en mindre opgave at justere sprogpakkerne til altid at kategorisere disse ord som interjektioner frem for konjunktioner eller substantiver som de hyppigt fejlklassificeres som. En nærmere undersøgelse af strukturen for imperativer i salmerne vil på samme måde højst sandsynligt kunne afsløre strukturer for brugen af imperativer i salmerne. Hvis mønstre kan identificeres, kan de oversættes til regelbaserede rettelser af klassificeringen som kan implementeres i sprogmodellerne.

Litteratur

Bigi, Brigitte (2014). A Multilingual Text Normalization Approach. *Human Language and Technology Challenges for Computer Science and Linguistics*, *8387:*515-526.

Den danske salmebog. (2003). Det Kgl. Vajsenshus.

Det Danske Sprog- og Litteraturselskab. (2023). *Vejledning: Sådan bruger du Danske Reformationssalmer. Melodier og tekster 1529-1573.* Det danske Sprog- og Litteraturselskab. Sidst tilgået den 29.01.2023 på https://salmer.dsl.dk/guidelines

Enevoldsen, Kenneth; Johansen, Lasse & Nielbo, Kristoffer Laigaard (2021). DaCy: A Unified Framework for Danish NLP. *CEUR Workshop Proceedings*, *2989:*206-216.

Honnibal, Matthew & Montani, Ines (2017). spaCy 2: Natural language understanding with Bloom embeddings, convulational neural networks and incremental parsing. Sidst tilgået 02.02.2023 på https://spacy.io/

Jacobsen, Henrik Galberg (2010). *Ret og Skrift: Officiel dansk retskrivning 1739-2005. Bind 1. Direktiver. Aktører. Normer.* Syddansk Universitetsforlag.

Ondelli, Stefano (2018). Treat Texts as Data but Remember They Are Made of Words: Compiling and Preprocessing Corpora. I A. Tuzzi (Red.), *Tracing the Life Cycle of Ideas in the Humanities and Social Sciences* Springer International Publishing:133-150.

Rossum, Guido van, & Drake, Fred L. (2009). Python 3 Reference Manual.

sprogteknologi.dk. (2023). *sprogteknologi.dk*. Digitaliseringsstyrelsen. www.sprogteknologi.dk.

Strømberg-Derczynski, Leon; Ciosici, Manuel R.; Christiansen, Morten H.; Baglini, Rebekah; Dalsgaard, Jacob Aarups; Fusaroli, Riccardo; Henrichsen, Peter Juel; Hvingelby, Rasmus; Kirkedal, Andreas & Kjeldsen, Alex Speed (2021). The Danish Gigawords Corpus. I *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Sverige. University Electronic Press.

Thomissøn, Hans (1569). *Den danske P s a l m e b o g / met mange Christelige Psalmer / Ordentlig tilsammenset formeret oc forbedret. Aff Hans Thomissøn. Prentet i Kiøbenhaffn / aff Laurentz Benedicht. Cum gratia et Privilegio Serenissimæ Regiæ Maiestatis.* Laurentz Benedicht.

Universal Dependencies (UD). (2014-2022). *Universal POS tags*. Universal Dependencies contributers. Sidst tilgået den 31.01.2023 på https://universaldependencies.org/u/pos/all.html