Kandidatspeciale

Titel: Opinion Mining

Forfatter: Nicolai Nyströmer

Vejledere: Bolette Sandford Pedersen, Center for Sprogteknologi, KU
Daniel Hardt, Department of IT Management, CBS

Dato: 29. juli 2013

Institutnavn: Institut for Nordiske Studier og Sprogvidenskab
Center for Sprogteknologi

Emne: Specialets overordnede emne er opinion mining. Dette udgøres af elementer fra maskinlæring og datalingvistik.

Tegn: 133.228 = 55.5 normalsider

Opinion mining

1 Indhold

2 Abstract ..... - 4 -

3 Indledning ..... - 5 -

3.1 Telenor brænder nallerne ..... - 5 -

3.2 Et stærkt værktøj ..... - 5 -

3.3 Store mængder data ..... - 5 -

3.4 Projektets mål og rammer ..... - 6 -

3.4.1 Opbygning af opgaven ..... - 6 -

4 Data Mining ..... - 8 -

4.1 Supervised vs unsupervised learning ..... - 8 -

4.1.1 Superviseret maskinlæring ..... - 8 -

4.2 Eksempler ..... - 8 -

4.3 Data på passende form ..... - 9 -

4.3.1 Repræsentation af instanserne ..... - 9 -

4.4 Maskinlæringsalgoritmer ..... - 10 -

4.4.1 Instance-based representation ..... - 10 -

4.4.2 Naive Bayes Classifier ..... - 11 -

4.4.3 Support Vector Machine ..... - 12 -

4.4.4 Soft Margin SVM ..... - 15 -

4.5 Videre behandling af data ..... - 17 -

4.5.1 Dimensionalitetens forbandelse ..... - 17 -

4.5.2 Eksponentiel forøgelse af hypoteserummet ..... - 17 -

4.5.3 Reduktion af dimensionaliteten ..... - 18 -

5 Udfordringer ved opinion mining ..... - 20 -

5.1 En svær definition ..... - 20 -

5.2 Korpusbaserede metoder ..... - 20 -

5.2.1 Kategorisering på baggrund af ordlister ..... - 20 -

5.3 Semantisk orientering ud fra semantisk association ..... - 21 -

5.3.1 Paradigmer af semantisk orientering ..... - 22 -

5.3.2 PMI-IR ..... - 22 -

5.3.3 Test af algoritmen ... - 23 -
5.4 Informationsudtrækning og subjektivitetsanalyse ... - 24 -
5.4.1 Subjektivitetsanalyse forbedret gennem IE ... - 24 -
5.4.2 IE forbedret gennem subjektivitetsanalyse ... - 31 -
6 Data ... - 35 -
6.1 Nykredit-korpusset ... - 35 -
6.2 Preprocessing ... - 36 -
6.2.1 Dubletter og neutrale artikler bortfiltreres ... - 36 -
6.2.2 Semidubletter ... - 36 -
6.3 Kategorisering af artikler ... - 39 -
6.4 Kvaliteten af data ... - 39 -
6.4.1 Intercoder agreement ... - 39 -
6.4.2 Annotering af Nykredit-korpusset ... - 44 -
6.4.3 Clear Cases ... - 45 -
6.5 Validering af datasættet ... - 46 -
6.5.1 Analyse af forsøget ... - 47 -
6.5.2 Datas velegnethed ... - 49 -
6.6 Opinion mining på journalistiske artikler ... - 49 -
7 Implementering af program ... - 51 -
7.1 Python ... - 51 -
7.2 Natural Language Tool Kit ... - 51 -
7.3 Scikit-Learn ... - 51 -
7.3.1 Træning og test på et mindre datasæt ... - 51 -
7.4 Beskrivelse af program ... - 52 -
7.4.1 pre_processing.py ... - 52 -
7.4.2 opinion_mining.py ... - 53 -
7.4.3 Træning og test med Scikit Learn ... - 53 -
7.5 Alternativ til min implementering - Majority Voting på sætningsniveau ... - 54 -
7.5.1 Bag of bag of words ... - 54 -
7.5.2 Sætningsbaseret frem for dokumentbaseret ... - 54 -
8 Afslutning ... - 56 -
9 Litteraturliste ... - 57 -

2 Abstract

Opinion mining is a subfield of the more general area of data mining which deals with identifying patterns and extracting information from unstructured data. In this thesis I am giving an introduction to opinion mining and I'll be going through some of the problems that make this task quite challenging.

Some machine learning algorithms will be described (Naïve Bayes and Support Vector Machine) and I will be giving examples of how different kinds of opinion mining have been conducted in previous experiments.

A part of the project is to produce a program that is able to perform opinion mining on a collection of Danish, journalistic articles regarding Nykredit and other actors related to Nykredit. The program is trained and tested on different splits of the articles. The data however turned out to be too invalid to be of any use for this task, and the overall score of the program is pretty disappointing (<76%).

3 Indledning

I et forsøg på at bestemme en forfatters holdning til det emne, en given tekst omhandler, kan man gøre brug af opinion mining eller sentiment analysis – to begreber, der dækker over samme indhold. Det er et forholdsvist nyt område inden for sprogteknologien, og det er et område, der er yderst relevant for blandt andet virksomheder og andre organisationer, som fx overvejer at bevæge sig ud på nettets mange sociale medier som en del af deres onlinestrategi.

3.1 Telenor brænder nallerne

At opinion mining overhovedet er en relevant disciplin, kan man med al ønskelig tydelighed se på den storm af dårlig omtale, teleselskabet Telenor bliver ramt af i august 2012[1]. En utilfreds Telenor-kunde opsiger sit abonnement hos selskabet, men det bliver ikke en opsigelse i det stille. Som det er blevet kutyme for nærmest alle virksomheder, har Telenor en Facebook-side, hvor enhver med en Facebook-profil kan lave opslag på deres væg – og det gør den snart forhenværende kunde hos Telenor. I et velskrevet og stærkt kritiserende opslag argumenterer han for sin store utilfredshed med selskabet, og på få timer tilkendegiver flere tusinde folk, at de "synes godt om" den verbale bredside, der rammer Telenor.

En utilfreds kunde, som på Telenors egen side skyder med skarpt efter selskabet, er i sig selv slemt nok, men det største problem for Telenor er, at de ikke reagerer på opslaget før næste dag, så op mod 30.000 personer i mellemtiden når at tilkendegive deres enighed i kritikken. Opslaget bliver nemlig postet uden for det tidsrum, hvor Telenors medarbejdere besvarer indlæg på siden, så de har på ingen måde mulighed for at lave "damage control" og styre den effekt, opslaget kan have. Derudover underbygger Telenors manglende deltagelse i tråden, der følger opslaget, jo et af kritikpunkterne, som netop går på en inkompetent kundeservice.

3.2 Et stærkt værktøj

Et opinion mining-værktøj kunne have tjekket opslaget, efter det var blevet postet. Det ville have klassificeret opslaget som værende "negativt" – måske endda "stærkt negativt" - og det ville straks have tændt for den røde, roterende lampe hos Telenors vagthavende medarbejder, der omgående ville besvare opslaget med et løfte om, at de straks ville tage sig af sagen, eller i det mindste give et passende svar. Et opinion mining-værktøj ville have havde forbedret Telenors muligheder for at kontrollere effekten af opslaget, før det eksploderede i en mediestorm af dårlig omtale. Og det ville have givet et billede af, at man hos Telenor lytter til sine kunder.

3.3 Store mængder data

Ligesom Telenor har rigtig mange virksomheder interesse i at vide, hvordan de og deres produkter bliver omtalt, og derfor tilbyder firmaer som bl.a. Infomedia og NewsWatch/Retriever at lave opinion mining for dem. Det fungerer i praksis ved, at et hold medarbejdere læser artikler og lignende igennem for herefter at bestemme, hvorvidt de er positive eller negative. Problemet med dette er naturligvis, at de enorme mængder data, der er tilgængelige på nettet, gør dette til et ganske omfattende arbejde.

[1] http://politiken.dk/tjek/digitalt/telefoni/ECE1712571/social-mediestorm-skubber-telenor-ud-i-imagekrise/

3.4 Projektets mål og rammer

Der eksisterer allerede en stor mængde resultater fra arbejdet med opinion mining, men langt størstedelen omhandler engelsksproget data, mens der ikke eksisterer tilgængelige resultater baseret på dansk. Traditionelt står arbejdet – og de deraf følgende resultater - på et lavressource-sprog som dansk tilbage for det ditto på engelsk. Og det er klart, at også feltets state-of-the-art-løsninger er tilpasset det engelske sprog.

Lige så klart er det dog, at der er brug for denne type teknologi rettet mod det danske sprog. For det første fordi der er et marked af aftagere, som er stærkt interesserede i dette, og ovennævnte virksomheder, som arbejder med at klippe nyheder, vil i højeste grad kunne drage fordel af denne type værktøjer. For det andet – og dette er måske af større vigtighed – er det nødvendigt, at den danske sprogteknologi også på dette område kommer op i omdrejninger og på højde med andre og mere udbredte sprog, hvad udvikling af sprogteknologi angår. Ellers risikerer vi simpelthen, at et sprog som dansk med et relativt lille antal sprogbrugere ikke kan overleve i det lange løb. Det talte hverdagssprog skal nok overleve, men hvad med områder - fx politik, lovgivning og handel - hvor det vil være aldeles brugbart at kunne forlade sig på sprogteknologisk funderede værktøjer? Her vil det blive svært at argumentere imod en overgang til engelsk, hvis løsninger, der på tilfredsstillende vis kan håndtere det danske sprog, bare ikke findes eller ikke er på højde med state of the art på området (Pedersen et al., s. 2).

3.4.1 Opbygning af opgaven

Jeg vil i dette speciale se på hvorledes et opinion mining-værktøj kan konstrueres. Jeg vil undersøge hvilke maskinlæringsteknologier, der er velegnet til denne opgave, og jeg vil gøre rede for nogle af de lingvistiske og statistiske forhold, der ligger bag. Endvidere vil jeg diskutere, hvad opinion mining eller sentimentanalyse er en analyse af.

Jeg vil forsøge mig med at lave et basalt program, der er i stand til at klassificere tekster ud fra, om de er positive eller negative. Programmet bliver trænet og testet på et dansksproget tekstkorpus, bestående af 2146 tekster, som i preprocessing-fasen bliver barberet ned til 1309 tekster. Teksterne er opmærket hos Infomedia, og Ankiro har leveret data. Ankiro er en dansk it-virksomhed, der siden slutningen af 1990'erne har arbejdet med sprogteknologi. Specialet er et samarbejde med Ankiro.

Preprocessing af data bliver lavet i Python – bl.a. ved hjælp af frameworket NLTK[2] - og træning og klassificering af data foregår med Scikit-Learn[3]. Gennemgang af lingvistiske begreber sker på baggrund af bl.a. Jurafsky & Martins *Speech and Language Processing*. Gennemgang af begreber omhandlende maskinlæring tager udgangspunkt i en række artikler samt *Introduction to Machine Learning* af Ethem Alpaydin.

I kapitel 2 introduceres data mining og maskinlæring. Jeg ser på, hvordan man kan repræsentere data og jeg gennemgår nogle relevante maskinlæringsalgoritmer. Kapitel 3 omhandler de grundlæggende udfordringer ved opinion mining, og der gives eksempler på, hvordan man rent praktisk har grebet forskellige problemstillinger an. I kapitel 4 ser jeg nærmere på de data, jeg har fået udleveret. Kvaliteten af

[2] Natural Language Tool Kit - http://nltk.org/

[3] http://scikit-learn.org/stable/

data bliver diskuteret, og jeg ser nærmere på, hvordan man i videnskabelige sammenhænge opmærker data, så dennes validitet sikres. Jeg har som skrevet lavet et ganske basalt program, og beskrivelse af dette findes i kapitel 5. Endelig afrundes opgaven i kapitel 6.

Formel hovedvejleder på specialet er professor Bolette Sandford Pedersen, Center for Sprogteknologi, KU, og ekstern vejleder er lektor Daniel Hardt, Department of IT Management, CBS.

4 Data Mining

Opinion mining er et underområde af det mere generelle *data mining*, som handler om at identificere mønstre og udtrække informationer fra store mængder ustrukturerede data. Formålet er at finde brugbare informationer og i vores tilfælde – klassifikation af tekster som værende enten positive eller negative - mønstre, der er så markante, at det gør os i stand til at lave præcise forudsigelser og klassifikationer af nye, ukendte data (Witten & Frank, p. xxiii).

4.1 Supervised vs unsupervised learning

Hvor data mining er disciplinen, er maskinlæring værktøjerne og det tekniske fundament, der muliggør denne. Man skelner overordnet mellem to typer maskinlæringsprocesser:

- I *supervised learning* træner man en maskinlæringsalgoritme på et datasæt, hvor datapunkterne er opmærket i forhold til deres klasse.
- I *unsupervised learning* forsøger den givne maskinlæringsalgoritme derimod at uddrage strukturer og mønstre i data, som ikke er opmærket. Dette kan for eksempel være at opdele et korpus af tekster i forhold til det emne, de omhandler.

Vi vil i det følgende kun beskæftige os med supervised learning.

4.1.1 Superviseret maskinlæring

Superviseret maskinlæring kan kort beskrives som processen at fremstille algoritmer, som tager ved lære af træningsdata eller tidligere erfaringer. Det gør det muligt at håndtere visse problemer, som ikke umiddelbart kan løses af manuelt fremstillede programmer, da der simpelthen ikke findes matematiske modeller af de givne problemer. I stedet kan de kun løses ved netop at gøre brug af eksempeldata og tidligere erfaringer (Alpaydin, s. Xxxi). Eksempelvis vides det ikke, hvordan man i hånden skriver et program, der kan udføre håndskrift-genkendelse, da der simpelthen ikke findes en matematisk model af, hvordan håndskrift bliver genkendt, mens det er relativt simpelt gennem metoder fra maskinlæring at konstruere sådan et program (Christianini & Shawe-Taylor, s. 1).

4.2 Eksempler

Et andet eksempel på anvendelse af maskinlæring er kreditvurdering. En kunde i et pengeinstitut ønsker at optage et lån, og til at afgøre, om vedkommende skal bevilliges dette, kan man gøre brug af en maskinlæringsalgoritme. Et program baseret herpå kan på baggrund af erfaringer med tidligere låntagere vurdere sandsynligheden for, at kunden vil opfylde sine forpligtelser over for pengeinstituttet. Det sker ved at se på en lang række af parametre som fx kundens formue, årsindkomst, sociale forhold osv.

Et tredje eksempel, hvor maskinlæring bliver benyttet, er bortfiltrering af spam fra ikke-spam i en mailboks – et typisk såkaldt klassifikationsproblem. Et program, der kan gøre dette, bliver i input givet en tekst, som den gør ”et eller andet” ved. Herefter tilskriver programmet teksten én af de to klasser ”spam” eller ”ikke-spam”.

Men hvordan kommer man fra input til det konkluderende output? Problemet er, at vi ikke umiddelbart besidder en algoritme, som kan udføre den ønskede klassifikation. Men hvad vi ikke har af algoritmer og anvendt viden, kan vi altså kompensere for med store mængder data, og vi kan relativt let indsamle mange

tusinde emails, som vi ved er spam. Det, vi ønsker, er at den algoritme, som udgør spamfilteret, skal lære hvilke elementer en mail skal indeholde, for at den kan blive klassificeret som spam.

Lige som spamfiltrering er opinion mining et klassifikationsproblem[4] – tilhører teksten klassen "positiv" eller "negativ" – og den største del af udfordringen består lige præcis i at finde de parametre, der er afgørende for, om teksten tilhører den ene eller anden klasse.

Inden for data mining og maskinlæring kaldes de parametre, man vurderer en tekst (eller hvad det nu må være, man ønsker behandlet) ud fra, *features*. Jeg vil i det følgende kort beskrive, hvordan en algoritme, der kan klassificere tekster – en såkaldt *classifier* – er bygget op.

4.3 Data på passende form

Der findes en mængde forskellige typer maskinlæringsalgoritmer, men før man kommer så langt, at man kan træne sin algoritme på træningsdata, skal disse forberedes og repræsenteres på en form, som gør det muligt at behandle dem matematisk. Man kan ikke bare slippe en klassifikationsalgoritme løs på et råt datasæt – data skal bearbejdes og forberedes til den givne opgave.

4.3.1 Repræsentation af instanserne

Hver artikel udgør en såkaldt instans, og der skal findes en passende form, hvorpå hver instans skal repræsenteres. En almen praksis i arbejdet med maskinel behandling af natursprog er at lade en tekst blive repræsenteret ved en featurevektor. Dette er en såkaldt vektorrum-model (Jurafsky & Martin, s. 673) – og vi lader indgangene i disse vektorer være binære. Hver indgang i vektoren repræsenterer en term i korpussets ordforråd, og afhængigt af, om termen optræder i teksten, vektoren repræsenterer, optræder et 0 eller 1.[5]

På baggrund af det samlede ordforråd (leksikonnet, som består af *D* ord), der bliver benyttet i korpusset, bliver den D-dimensionelle featurevektor altså konstrueret således, at hver indgang svarer til en term i ordforrådet:

$$\boldsymbol{d_j} = (t_1, t_2, \dots, t_D)$$

4.3.1.1 Bag of words

Denne repræsentation af en tekst kaldes en *bag of words*-model. Som i en pose ligger ordene hulter til bulter, og alt hvad der hedder kontekst og syntaktiske relationer går fuldstændig fløjten, når denne metode tages i brug. Til trods for det, der synes som en temmelig grov handling – at smide en sammenhængende tekst ned i en pose for blot at ryste den – er det alligevel en metode, som kan give rigtig gode resultater. Man skal dog være bevidst om, at der også går en masse informationer tabt. Eksempelvis mister negationer

[4] Opinion mining kan også være et såkaldt regressionsproblem, hvor output i stedet for at være en bestemt klasse er en værdi på en kontinuert skala (Alpaydin, s. 9). Man kan fx definere en skala fra 0 (negativ) til 10 (positiv), som vurderingen af en tekst bliver udregnet i forhold til.

[5] Man kan også vælge at lade featurevektorens indgange indikere hvor mange gange de enkelte termer optræder i dokumentet, hvilket inden for emneklassifikation kan forbedre klassifikationsresultaterne. Ifølge [Pang et al (1988), s. 6] gør det stik modsatte sig dog gældende inden for opinion mining, hvor der opnås bedre resultater ved kun at notere tilstedeværelse frem for frekvens for de enkelte termer. Dette kan skyldes, at en teksts emne kan blive understreget gennem gentagelse af visse nøgleord, mens en teksts orientering, hvad positivitet og negativitet angår, ikke nødvendigvis bliver forstærket ved gentagelse af de samme termer (Pang & Lee, s. 21).

deres betydning: ”Filmen er slet ikke dårlig.” Trækkes *dårlig* op af posen, kan det give negative associationer, selvom det jo faktisk ikke er tilfældet. Dette kan man råde bod på ved at skelne mellem ord i deres isolerede form og ord, der bliver negeret i teksten. Man kan fx føje en underscore til *dårlig*, inden man dumper det ned i posen som *dårlig_*. Således vil det være klart, at når man trækker et ord op af posen, som har en underscore til sidst, så er det ordet i dets negerede form, der er tale om.

En anden type information, der går tabt, er den et idiom bringer med sig: ”Filmen var ikke lige min kop the.” Udsagnet har lige så lidt med the, som det har med en kop at gøre – men det spiller bare ingen rolle, når alle ordene ligger og roder rundt nede i den pose, der repræsenterer teksten. En løsning på dette problem kunne være at tjekke en tekst igennem for idiomer, for derefter at vælte hele idiomet som en samlet enhed ned i posen og ikke blot som enkeltord. Ligeledes kunne det begrænse tabet af informationer, hvis man lod teksten parse, før den røg i posen, således at det ikke var enkeltord, men fraser, man sendte videre.

4.4 Maskinlæringsalgoritmer

Når man for hver instans i datasættet har en featurevektor og den tilhørende klasse, kan man træne en maskinlæringsalgoritme på dette datasæt. Det er desuden værd at notere sig, at foruden at være i stand til at forudsige klassen af fremtidige, ukendte instanser, udgør en maskinlæringsalgoritme også en form for repræsentation af det datasæt, den er blevet trænet på. For hvis den i en klassifikationsopgave klassificerer ukendte instanser korrekt, må den jo nødvendigvis have formået at trække de elementer eller features ud af teksterne, som ligger til grund for, at træningsdata blev klassificeret, som de nu engang blev af de annotører, der opmærkede data. Kompleksiteten af disse repræsentationer varierer fra algoritme til algoritme, men lad os starte med at se på den simpleste af disse.

4.4.1 Instance-based representation

Den simpleste måde at repræsentere et datasæt på er simpelthen datasættet - eller de instanser, der udgør sættet – selv (Witten & Frank, s. 76). Denne repræsentation kaldes instansbaseret. Træningen af en classifier, som er bygget på dette princip, sker ved blot at indlæse træningssættets instanser i hukommelsen. Når classifieren så står over for en ny ukendt instans (en testinstans), bliver klassen af denne testinstans bestemt ud fra klassen af den instans i træningssættet, som ligner testinstansen mest. Dette kaldes *nearest-neigbour*-klassifikation. Similariteten instanserne imellem afgøres af en metrik, der beregner afstandene mellem instanserne i det vektorrum, de befinder sig i.

I stedet for blot at lade den træningsinstans, der ligger tættest på testinstansen bestemme klassen, kan man også vælge at lade den klasse, som størstedelen af de k nærmeste træningsinstanser har, bestemme klassen. Dette kaldes *k-nearest-neighbour*-klassifikation.

Ved instansbaseret maskinlæring foregår læringsdelen først, når nye, ukendte instanser skal klassificeres – det beregningsmæssige arbejde finder altså ikke sted, idet træningsdata bliver processeret. Instansbaseret læring er på den måde ”doven”, da den udskyder det reelle arbejde længst muligt, mens andre maskinlæringsalgoritmer er mere ”ivrige” og producerer en model (dvs. classifier), allerede så snart den bliver gjort bekendt med træningsdata. Vi skal nu se eksempler på to algoritmer af denne type – Naive Bayes og Support Vector Machine.

4.4.2 Naive Bayes Classifier

En Naive Bayes Classifier er en statistisk funderet generativ classifier, og den beregner af en række klasser, hvilken klasse, en tekst sandsynligvis tilhører. Lad fx C betegne en af de klasser, en tekst kan tilhøre, og lad b$\boldsymbol{x} = (f_1, \dots, f_m)$ betegne featurevektoren for den givne tekst. Indgang f_j i $\boldsymbol{x}$ antager værdien 1, hvis termen f_j optræder i teksten $\boldsymbol{x}$**.** For hver klasse skal det beregnes, hvad sandsynligheden er for, at $\boldsymbol{x}$ tilhører klassen C, dvs. $P(C|\boldsymbol{x})$. Hvis vi benytter Bayes' Formel, kan dette skrives således:

$$P(C|\boldsymbol{x}) = \frac{P(C)P(\boldsymbol{x}|C)}{P(\boldsymbol{x})}$$

$P(C|\boldsymbol{x})$ bliver altså beregnet på baggrund af $P(C)$ (sandsynligheden for, at en vilkårlig tekst er af klassen C), $P(\boldsymbol{x}|C)$ (sandsynligheden for, at en tekst er lig med $\boldsymbol{x}$, når det er givet, at den tilhører klassen C) og $P(\boldsymbol{x})$ (sandsynligheden for, at en vilkårlig tekst er lig med $\boldsymbol{x}$).

Antag, at der er n klasser C_1 til C_n. Vi skal nu bestemme, hvilken klasse teksten $\boldsymbol{x}$ sandsynligvis tilhører. Dette gøres ved at beregne $P(C_i|\boldsymbol{x})$ for hver klasse C_i. Bayes' Formel giver os følgende:

$$P(C_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|C_i)P(C_i)}{P(\boldsymbol{x})}$$

$P(\boldsymbol{x}|C_i)$ er sandsynligheden for, at en instans, givet at den tilhører klassen C_i, indeholder de ord, featurevektoren $\boldsymbol{x}$ indikerer. $P(C_i)$ er sandsynligheden for, at en instans uden at tage indholdet i betragtning tilhører klassen C_i. Indeholder træningsdata eksempelvis 1000 instanser, hvoraf 150 tilhører klassen C_i, er $P(C_i) = 0.15$.

Vi skal altså finde det i, som maksimerer $P(C_i|\boldsymbol{x})$. Eftersom $P(\boldsymbol{x})$ ikke varierer over i og udelukkende fungerer som en skalar for de forskellige værdier af $P(C_i|\boldsymbol{x})$, kan vi nøjes med at regne tælleren ud for at finde $\max_i P(C_i|\boldsymbol{x})$. Teksten $\boldsymbol{x} = (f_1, \dots, f_m)$ består af m features, som hver repræsenterer en term i tekstsamlingen. Vi står nu over for det skridt, som har givet anledning til benævnelsen Naive Bayes Classifier, for vi laver nemlig den naive antagelse, at de m features er uafhængige. Dette svarer til, at vi antager, at ordene i teksten ingen relation har til hinanden (bag-of-words som beskrevet i 4.3.1.1), hvilket naturligvis ikke er tilfældet, men det har alligevel vist sig at give gode resultater inden for netop tekstklassifikation (Zhang, s. 1).

Antagelsen af stokastisk uafhængighed giver os med definitionen af denne (definition 1.5.4 i Sørensen) følgende:

$$P(\boldsymbol{x}|C_i) = P(f_1, \dots, f_m|C_i) = \prod_{j=1}^{m} P(f_j|C_i)$$

$P(f_j|C_i)$ er sandsynligheden for, at termen f_j optræder i en tekst af klassen C_i. I praksis beregnes dette således:

$$P(f_j|C_i) = \frac{\text{antal tekster af klassen } C_i\text{, som indeholder } f_j}{\text{antal tekster af klassen } C_i}$$

Således tilskriver Naive Bayes Classifieren $\boldsymbol{x}$ den klasse C_i, for hvilken i opfylder følgende:

$$\max_i \prod_{j=1}^{m} P(f_j|C_i)\, P(C_i)$$

4.4.3 Support Vector Machine

En Support Vector Machine (SVM) er en såkaldt diskriminativ classifier, som kort fortalt etablerer en skillelinje, som - baseret på de instanser, den trænes på - adskiller instanserne klassevist. En SVM er en meget mere avanceret classifier, som til forskel fra Naive Bayes er en ikke-probabilistisk classifier, men i stedet defineret geometrisk.

Lad datasættets N instanser være defineret således:

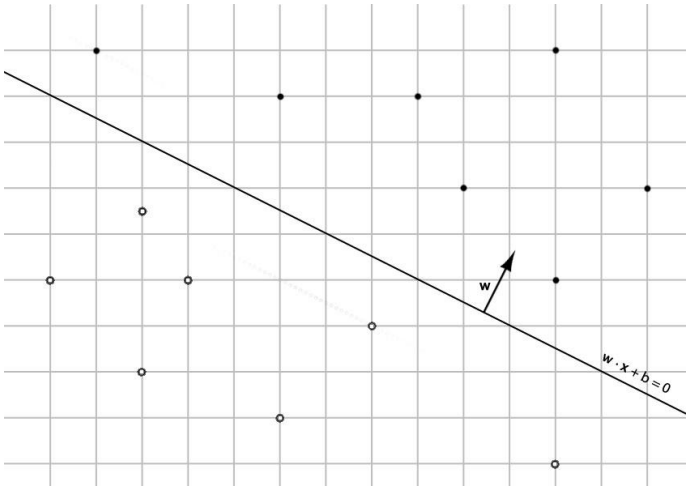
$$\{(x_i, y_i) \mid i = 1, \dots, N, x_i \in \mathbb{R}^D, y_i \in \{-1, 1\}\}$$

x_i udgør altså featurevektorerne (som er D-dimensionelle), mens y_i er de tilhørende klasser.

Vi antager, at instanserne i datasættet er lineært separable, dvs. at instanserne tilhørende de to klasser kan adskilles af et hyperplan. Ethvert hyperplan kan beskrives som de punkter $x \in \mathbb{R}^D$ repræsenteret ved den tilhørende positionsvektor, som opfylder

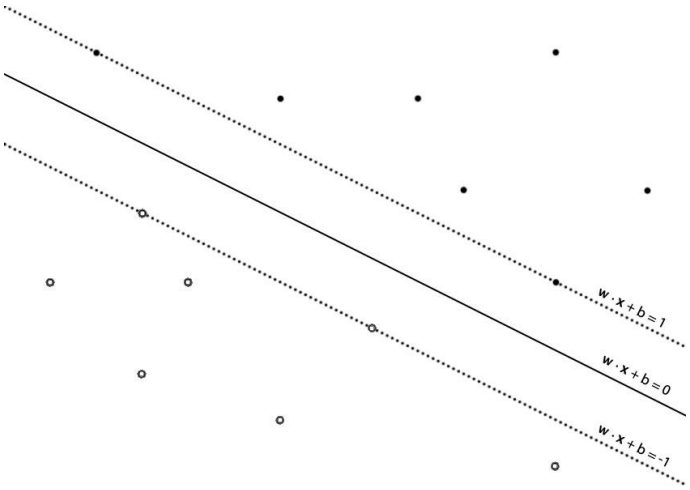
$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

hvor $\boldsymbol{w}$ er normalvektor til hyperplanet, dvs. den står vinkelret på planet:



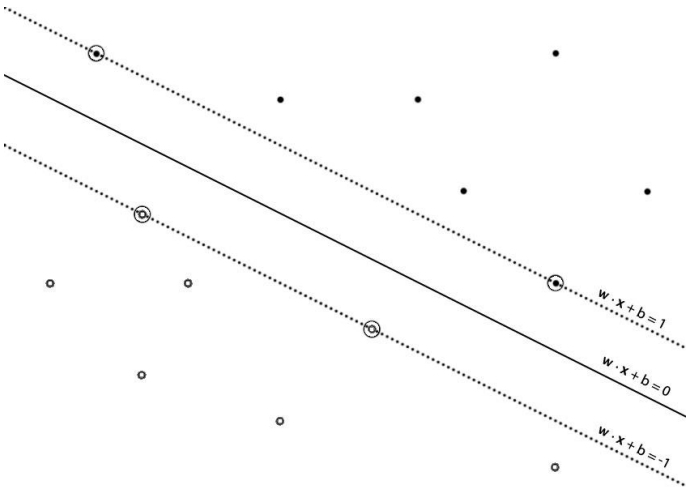
Figur 4.1

Vi ønsker at finde det optimalt adskillende hyperplan, dvs. det hyperplan, der har størst mulig afstand eller margen til instanserne af hver klasse. Dette skyldes, at classifieren dermed vil være mindre følsom over for støj blandt fremtidige instanser, der skal klassificeres, og den vil generelt lave færre fejl, jo større margenen er (Alpaydin, s. 311). Margenen er afgrænset af to parallelle hyperplaner, som hver kan skrives $\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$ og $\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$:



Figur 4.2

Instanserne (eller rettere de tilknyttede repræsentationsvektorer) tættest på det separerende hyperplan benævnes *supportvektorer* – det er de indcirklede punkter:



Figur 4.3

Målet for en Support Vector Machine er at orientere det separerende hyperplan, så det er så langt væk som muligt fra supportvektorerne af begge klasser (Fletcher, s. 2). Først og fremmest skal $\boldsymbol{w}$ og b vælges, således at datasættet kan beskrives ved

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq 1 \text{ for } y_i = 1$$
$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 \text{ for } y_i = -1$$

Disse uligheder kan samles som

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geq 0, \forall i$$

De to planer, der ligger på randen af margenen og på hvilke supportvektorerne ligger, benævnes H_1 og H_2. De kan skrives således:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b = 1 \text{ for } H_1$$
$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b = -1 \text{ for } H_2$$

Da afstanden fra hyperplanet til margenen jo skal være så stor som muligt, og da afstanden fra hyperplanet til H_1 og H_2 er $\frac{1}{\|\boldsymbol{w}\|}$, maksimeres afstanden, jo mindre $\|\boldsymbol{w}\|$ er. Vi står således over for optimeringsproblemet at minimere $\|\boldsymbol{w}\|$ under bibetingelsen $y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geq 0, \forall i$.

Dette optimeringsproblem afhænger af b og $\|\boldsymbol{w}\|$, normen af $\boldsymbol{w}$, som bl.a. omfatter en kvadratrod. Dette gør det vanskeligt at løse problemet, men at minimere $\|\boldsymbol{w}\|$ er ækvivalent med at minimere $\frac{1}{2}\|\boldsymbol{w}\|^2$. Vi skal således finde

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 \text{ under bibetingelsen } y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geq 0, \forall i$$

Vi skal bl.a. benytte Lagranges Metode (Solovej, s. 32) til at finde løsningen. Denne metode gør det i korte træk muligt på relativ enkel vis at finde ekstremum af en funktion $f(x)$ af flere variable under bibetingelsen $g_i(x) = 0, i = 1, \dots N$. (Da bibetingelsen i vores optimeringsproblem ikke er en lighed, men en ulighed, skal problemet opfylde de såkaldte Karush-Kuhn-Tucker-betingelser. Disse gør det muligt at benytte uligheder som bibetingelser.) Det kan ofte være svært at finde løsningen til bibetingelsen for derefter at finde ekstremum for f inden for denne løsningsmængde. Lagranges Metode går i stedet ud på at finde ekstremum for den lagranske hjælpefunktion $L(x, \lambda) = f(x) - \lambda_i g_i(x)$ under bibetingelsen $\lambda_i \geq 0, i = 1, \dots, N$. λ_i er såkaldte Lagrangemultiplikatorer, og $f(x)$ kaldes objektfunktionen. Hvis gradienten[6] af L er lig 0 i punktet (symboliseret ved $\nabla L(x_0) = 0$), så udgør x_0 et ekstremum for L (Klein, s. 4).

Vores lagranske hjælpefunktion (i såkaldt *primal* form) afhænger af $\boldsymbol{w}$, b og λ og ser således ud:

$$L_P(\boldsymbol{w}, b, \lambda) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \lambda[y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1]$$
$$= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \lambda_i[y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1]$$
$$= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \lambda_i y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \lambda_i$$

Målet er af finde det $\boldsymbol{w}$ og b, som minimerer L_P, og de λ_i, som maksimerer den. Dette punkt udgør et saddelpunkt, og da både objektfunktionen og de lineære funktioner givet ved bibetingelsen er konvekse, er også L_P konveks. Således er Karush-Kuhn-Tucker-betingelserne opfyldt, og vi kan løse dette såkaldt duale problem. Det duale problem er at maksimere L_P med hensyn til λ_i med bibetingelserne $\frac{\partial L_P}{\partial \boldsymbol{w}} = 0$, $\frac{\partial L_P}{\partial b} = 0$ og $\lambda_i \geq 0$ (Alpaydin, s. 312). L_P differentieret med hensyn til $\boldsymbol{w}$ og b og sat lig 0 giver følgende:

---

[6] Lad *U* være en åben delmængde af $\mathbb{R}^n$ og $f: U \to \mathbb{R}$ en funktion. alle de partielt afledte, $\frac{\partial f}{\partial x_i}(x), i = 1, \dots, n$ af f eksisterer i punktet $x \in U$, kaldes vektoren

$$\left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \frac{\partial f}{\partial x_3}(x), \dots, \frac{\partial f}{\partial x_n}(x)\right) \in \mathbb{R}^n$$

for gradienten af *f* i punktet *x* (Thue Poulsen, s. 210)

$$\frac{\partial L_P}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i y_i = 0$$

Således har vi fundet det $\boldsymbol{w}$, vi skal bruge, og hvis vi erstatter $\boldsymbol{w}$ og b i den lagranske hjælpefunktion med de to ovenstående udtryk, får vi den duale funktion L_D, som kun er afhængig af λ_i:

$$L_D(\lambda_i) = \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} - \boldsymbol{w} \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} - \boldsymbol{w} \cdot \boldsymbol{w} - b \cdot 0 + \sum_{i=1}^{N} \lambda_i$$

$$= -\frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} + \sum_{i=1}^{N} \lambda_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N} \lambda_i$$

Vi maksimerer denne funktion med hensyn til λ_i under bibetingelserne $\sum_{i=1}^{N} \lambda_i y_i = 0$ og $\lambda_i \geq 0, \forall i$ – et problem, der kan løses med metoder fra kvadratisk programmering. Der er N forskellige λ_i'er, men for langt de fleste gælder, at $\lambda_i = 0$ og kun ganske få λ_i'er er større end 0. Mængden af $\boldsymbol{x}_i$'er, for hvilke $\boldsymbol{x}_i > 0$ er supportvektorerne, og da $\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i$ ses det, at $\boldsymbol{w}$ udgøres af en vægtet sum af de træningsinstanser, der er valgt som supportvektorer. Det er de vektorer, som ligger på margenen og opfylder

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) = 1 \Leftrightarrow \boldsymbol{x}_i \cdot \boldsymbol{w} + b = \frac{1}{y_i} \Leftrightarrow \boldsymbol{x}_i \cdot \boldsymbol{w} + b = y_i \Leftrightarrow b = y_i - \boldsymbol{x}_i \cdot \boldsymbol{w}$$

Vi kan nu for hver supportvektor beregne b, og for at sikre en vis numerisk stabilitet anbefales det netop at beregne alle disse b'er og herefter at tage gennemsnittet:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} y_i - \boldsymbol{x}_i \cdot \boldsymbol{w}$$

hvor N_{SV} er antallet af supportvektorer. Vi har nu størrelserne $\boldsymbol{w}$ og b, som definerer det separerende hyperplans optimale orientering, og dermed har vi vores Support Vector Machine.

4.4.4 Soft Margin SVM

Problemet med standard SVM-algoritmen beskrevet ovenfor er, at den kun fungerer, hvis datasættet er lineært separabelt. Hvis to klasser ikke er lineært separable, dvs. at der ikke findes et hyperplan, der adskiller dem fuldstændigt, må vi i stedet finde det hyperplan, som forårsager mindst fejl. I arbejdet på

dette indføres såkaldte *slack*-variable $\xi_i \geq 0, i = 1, \ldots, N$, som varierer alt efter hvor meget, $\boldsymbol{x}_i$ bliver fejlklassificeret i forhold til det separerende hyperplan (Alpaydin, s. 315). Instanserne i datasættet skal derfor ikke længere opfylde disse uligheder:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq 1 \text{ for } y_i = 1$$
$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 \text{ for } y_i = -1$$

De skal i stedet blot opfylde disse:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq 1 - \xi_i \text{ for } y_i = 1$$
$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 + \xi_i \text{ for } y_i = -1$$

som kan samles som

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 + \xi_i \geq 0$$

Der mindes om, at $\boldsymbol{w}$ og b er de størrelser, som definerer hyperplanet, og ξ_i antager præcis den værdi, der gør, at den givne ulighed er opfyldt. Således gælder, at hvis $\xi_i = 0$, klassificerer den fremkomne SVM $\boldsymbol{x}_i$ korrekt. Hvis $0 < \xi_i < 1$, er $\boldsymbol{x}_i$ ligeledes korrekt klassificeret, men placeret i margenen omkring hyperplanet. Og hvis $\xi_i \geq 1$, er $\boldsymbol{x}_i$ forkert klassificeret, dvs. den befinder sig på den helt forkerte side af hyperplanet i forhold til dens klasse.

Vi står som i foregående sektion over for et optimeringsproblem, der løses som før. Objektfunktionen $\frac{1}{2}\|\boldsymbol{w}\|^2$ bliver nu udvidet med endnu et led, som tildeler en straf, hvis $\xi_i > 0$:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i \text{ under bibetingelsen } y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 + \xi_i \geq 0, \forall i$$

Parameteren C kontrollerer en trade-off mellem hvor meget slack-variablen straffer fejlklassificeringer, og størrelsen af margenen (Fletcher, s. 7). Den tilhørende lagranske hjælpefunktion skal minimeres med hensyn til $\boldsymbol{w}$ og b og maksimeres med hensyn til λ, og den ser således ud:

$$L_P(\boldsymbol{w}, b, \lambda) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i[y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 + \xi_i] - \sum_{i=1}^{N}\mu_i\xi_i$$

$\mu_i \geq 0$ er den nye Lagrange-multiplikator, som skal sikre, at ξ_i er positiv (Alpaydin, s. 315). Som før differentierer vi med hensyn til $\boldsymbol{w}$, b og ξ_i og sætter de afledte lig 0:

$$\frac{\partial L_P}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N}\lambda_i y_i = 0$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 \Rightarrow C = \lambda_i + \mu_i$$

Eftersom $\mu_i \geq 0$ og $C = \lambda_i + \mu_i$, er $0 \leq \lambda_i \leq C$, og indsætter vi disse i L_P, finder vi frem til L_D, der har samme form som før, men med en ændret bibetingelse:

$$L_D(\lambda_i) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N}\lambda_i$$

Igen skal L_D maskimeres med hensyn til λ_i med bibetingelserne $\sum_{i=1}^{N}\lambda_i y_i = 0$, men λ_i skal nu ikke længere bare være større eller lig 0, men opfylde $0 \leq \lambda_i \leq C, \forall i$. For de instanser, der ligger på den rigtige side af hyperplanet og med tilstrækkelig afstand hertil (dvs. $\xi_i = 0$), er $\lambda_i = 0$. Supportvektorerne udgøres af alle de instanser, der befinder sig i margenen omkring hyperplanet. For de instanser, som ligger i margenen eller er forkert klassificeret, er $\lambda_i = C$. For alle supportvektorer er $\lambda_i > 0$, og de definerer $\boldsymbol{w} = \sum_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i$.

De supportvektorer for hvilke $\lambda_i < C$ ligger på randen af margenen, og de benyttes til at beregne b. For disse gælder, at $\xi_i = 0$, og de opfylder $y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) = 1$. Som i det første separable tilfælde anbefales det at beregne b for alle supporvektorer og derefter tage gennemsnittet.

4.5 Videre behandling af data

Vi har altså nu i 4.3.1 defineret en passende form at repræsentere de enkelte dokumenter i datasættet på, og i 4.4 blev det beskrevet, hvordan et par forskellige maskinlæringsalgoritmer fungerer. Der er imidlertid et problem – eller i hvert fald en betydelig udfordring – ved den måde, dokumenterne bliver repræsenteret på i 4.3.1.

Som udgangspunkt er dimensionaliteten af featurevektoren for de enkelte dokumenter nemlig lig størrelsen af antallet af de termer, der samlet bliver benyttet i alle dokumenterne i korpusset. Og i vores tilfælde med mere et korpus bestående af mere end 1.300 dokumenter er det klart, at dimensionaliteten er enorm (47.185 for at være præcis). Dette udgør et problem af flere årsager, og det har bl.a. relation til begrebet *curse of dimensionality* – dimensionalitetens forbandelse.

4.5.1 Dimensionalitetens forbandelse

Vi betragter et fikseret datasæt. Som dimensionaliteten stiger (dvs. antallet af features forøges), bliver datapunkterne spredt ud over et større rum, så afstandene punkterne imellem kommer til at ligne hinanden. Statistisk tæthed er et udtryk, der dækker over hvor hyppigt et eller flere udfald (dvs. instanser eller datapunkter) forekommer i en observation, eller for hvordan et antal udfald fordeler sig i en observation. I dette eksempel falder datasættets statistiske tæthed altså, når dimensionaliteten stiger, så for at sikre samme statistiske tæthed, kræves således flere datapunkter. Tætheden er proportional til $M^{\frac{1}{N}}$, hvor *M* er antallet af datapunkter og *N* er antallet af dimensioner/features (Liu & Motoda, s. 20).

4.5.2 Eksponentiel forøgelse af hypoteserummet

En anden grund til, at et stort antal features udgør et problem, er den simple grund, at der dermed opnås et enormt og enormt beregningskrævende hypoteserum. En hypotese er et mønster eller en funktion (dvs. en maskinlæringsalgoritme, i vores tilfælde en classifier), som bestemmer en instans' klasse på baggrund af dennes featurevektor. Jo flere features, desto større hypoteserum, dvs. desto flere mulige hypoteser findes der. En lineær stigning i antallet af features afføder sågar en eksponentiel forøgelse af hypoteserummet, og

for et datasæt af instanser repræsenteret af *N* binære features og to mulige klasser er hypoteserummet så stort som 2^{2^N} (Liu & Motoda, s. 4).

4.5.3 Reduktion af dimensionaliteten

Der findes imidlertid metoder til at reducere featurevektorens størrelse.

4.5.3.1 Trimming på baggrund af frekvensdistributionen

Den første og simpleste reduktion kan bestå i at fjerne de features – dvs. termer – fra featurevektoren, som optræder færrest og flest gange i korpusset. Idéen er, at de termer, som forekommer hyppigst, formentlig er at finde i langt størstedelen af alle dokumenterne i korpusset. Således fungerer disse termer dårligt som kriterie for, om et dokument tilhører én klasse frem for en anden. Omvendt findes de termer, som forekommer sjældent, kun i ganske få dokumenter. Derfor fungerer de heller ikke som diskriminative størrelser, da en klassifikation på baggrund af disse termer kun vil være brugbar for de dokumenter, de figurerer i. Dette vurderes altså at være så forsvindende få dokumenter, at termerne i stedet bliver fjernet for at reduceret dimensionen af featurevektorerne (Guyon, p.1158).

4.5.3.2 Stemming

Langt størstedelen af ord kan skrives på flere forskellige måder afhængigt af hvilken del, de udgør i den kontekst, de optræder. I korpussets ordforråd optræder eksempelvis termer som *økonomi, økonomis* og *økonomiers*, som umiddelbart bliver betragtet og behandlet som værende tre lige så forskellige termer som *automobil, finanskrise* og *golfbane* er det. Dette er naturligvis ikke i overensstemmelse med ordenes betydning, da alle tre ord jo er afledt af den samme ordstamme *økonomi*. Således bliver der i optællingen af forekomster af de forskellige ord i korpusset lavet en separat optælling for hver af de tre afledninger af ordet. Dette problem kan til dels afhjælpes af en stemmer (Jurafsky & Martin, s. 680).

En stemmer er en funktion, som på baggrund af en række regler fjerner et ords suffiks, så man står tilbage med en form for grundstamme af ordet. En regel kan for eksempel være at fjerne *–er* fra termernes endelser, så man i tilfældet *økonomier* efter stemming står tilbage med ordet *økonomi*. Således kan man med en stemmer slå flere forskellige termer sammen til ét, som har den samme betydning, og frekvenserne for disse bliver derfor lagt sammen til et enkelt højere tal. Dette kan have betydning for nogle termer, om de bliver medtaget i den endelige featurevektor på grund af ændringer i frekvensdistributionen.

4.5.3.3 Recursive Feature Elimination

Ovenstående metoder er effektive og ikke mindst intuitivt klare, når det kommer til at begrænse antallet af de features, en tekst skal klassificeres på baggrund af. Men til trods for, at de reducerer dimensionaliteten af featurerummet drastisk, ender man stadig gerne op med et featurerum af en betragtelig størrelse, når træningen af en given classifier skal i gang. Efter behandling og klargøring af data fra Nykredit-korpusset i afsnit 7 består featurevektorerne af ikke mindre end 991 features.

4.5.3.3.1 Overfitting og regularization

Når dimensionaliteten af featurerummet reduceres, sker det ikke kun på grund af en besparelse i beregningerne, men også for at undgå at komme til at overfitte maskinlæringsalgoritmen. Overfitting opstår, når antallet af features er relativt stort i forhold til antallet af træningsinstanser (Guyon et all, s. 391). I denne type situationer er det nemlig nemt at finde en funktion, der separerer træningsdata klassevist, men når der skal trænes på ukendte testdata, er der stor chance for at lave fejl.

For at undgå overfitting gør visse algoritmer brug af *regularization* under træningen. Kort forklaret er regularization en metode til at begrænse den kompleksitet, en algoritme antager under træning på træningsdata. Som skrevet kan man ofte relativt nemt konstruere en algoritme, der givet en vis kompleksitet scorer rigtig højt på træningsdata, men når der introduceres nye, ukendte datapunkter, falder algoritmens ydeevne drastisk. Dette kan ofte være et produkt af overfitting, og med regularization kan man opnå bedre resultater på nye data mod et fald i præcision på træningsdata. Og skønt Support Vector Machine gør brug af regularisering (i form af *C*-parameteren i 4.4.4), er der alligevel meget at hente ved reduktion af dimensionaliteten af featurerummet.

Det er muligt at foretage en udtømmende afprøvning af hvilken af samtlige delmængder af de oprindelige features, der giver den bedste mulige score, men i praksis er det en skidt idé, da det afføder en uhyrligt stor mængde beregninger. *Recursive Feature Elimination* (RFE) er en metode, der griber tingene mere praktisk an.

Ved RFE starter man ud med samtlige features, der umiddelbart er udvalgt til at repræsentere teksterne i et givent korpus. Classifieren bliver trænet, og efter endt træning, bliver samtlige features på baggrund af testdata rangeret af en dertil hørende algoritme. Den lavest rangerede feature bliver herefter fjernet, og processen gentages, indtil det ønskede antal features er opnået. Man kan vælge at fjerne flere af de dårligst rangerede features ved slutningen af hver iteration for at begrænse beregningerne.

Det er i dette kapitel blevet beskrevet, hvordan et datasæt kan repræsenteres, så det bliver muligt at arbejde med det i forbindelse med maskinlæring. Endvidere det gennemgået, hvorledes tre maskinlæringsalgoritmer – primært Naive Bayes og Support Vector Machines – fungerer, og hvordan data kan behandles, så det bliver mere håndterbart. Jeg vil nu diskutere hvilke udfordringer, der generelt er forbundet med opinion mining.

5 Udfordringer ved opinion mining

Opinion mining er videnskaben eller måske kunsten og den i hvert fald overhovedet ikke trivielle opgave at bestemme, hvordan en tekst forholder sig til det emne, den omhandler.

Men hvad er det, der gør, at man kan føle, at en tekst udtrykker et positivt eller negativt budskab? Og hvad griber man fat i, når man vil undersøge dette? Som skrevet er det et område, mange virksomheder har stor interesse i, og allerede i dag eksisterer der firmaer i Danmark, som tilbyder at udføre opinion mining. De sidder og laver arbejdet i hånden – det vil sige, at de har medarbejdere siddende, som manuelt (så at sige) læser tekster igennem for herefter at kategorisere dem. Det giver sig selv, at det ville være smart at få en computer til at udføre dette kategoriseringsarbejde. Rationalet er tid, penge og et overblik over hvad der sker på et ellers uoverskueligt internet og hvad der dertil hører.

5.1 En svær definition

En helt grundlæggende udfordring ved opgaven er, at det er svært præcist at definere, hvad der gør en tekst positiv eller negativ. Hvis en forfatter fx benytter sig af ironi i en tekst, vil den ironiske tone gå hen over hovederne på nogle af læserne, mens andre opfatter den klart. Således vil det for disse læsere umiddelbart være svært at blive enige om, hvad tekstens forhold til emnet er.

Lad os begynde med at indføre begrebet *polaritet*. En teksts polaritet dækker netop over, hvordan forfatteren af teksten (og dermed også teksten selv – jeg skelner ikke mellem disse to størrelser) forholder sig til tekstens emne.

5.2 Korpusbaserede metoder

Det må være klart, at det er i tekstens ord og sammensætningen af disse, at vi skal søge efter de bestanddele, som konstituerer tekstens polaritet. Se fx en tekst som denne:

> *Bilen er grim, og den er frygtelig at køre i*

Hvis *bilen* angiver emnet for teksten, giver *grim* og *frygtelig* anledning til at mene, at teksten udtrykker et negativt budskab herom. Omvendt ville ordene *flot* og *behagelig* være med til at give følgende tekst et positivt udtryk:

> *Bilen er flot, og den er behagelig at køre i*

Således vil det umiddelbart være en oplagt tanke, at man blot på baggrund af en række nøgleord kan putte en tekst i en af kategorierne positiv eller negativ.

5.2.1 Kategorisering på baggrund af ordlister

Der er dog et problem tilknyttet denne idé. Det er nemlig langt fra en let opgave at konstruere de lister, der skulle ligge til grund for sådan en kategorisering – i hvert fald ikke hvis man søger en løsning, der kunne være den optimale. I et eksperiment satte Pang et. al (2002, s. 3) netop to personer til hver at konstruere to lister bestående af ord, som personerne formodede ville fungere som gode indikatorer for, om en tekst fra en samling af engelsksprogede filmanmeldelser var positiv eller negativ. Datasættet bestod af 700 positive filmanmeldelser og 700 negative:

Tabel 5.1

	Foreslåede ordlister	Korrekt	Uafgjort
Person 1	Positive ord: *dazzling, brilliant, phenomenal, excellent, fantastic* Negative ord: *suck, terrible, awful, unwatchable, hideous*	58%	75%
Person 2	Positive ord: *gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting* Negative ord: *bad, clichéd, sucks, boring, stupid, slow*	64%	39%

Som man ser på ovenstående lister, giver deres valg af ord intuitivt ganske fin mening. I hver anmeldelse blev ordene fra ordlisterne talt op, og afhængig af, om der var flest ord fra den positive eller negative ordliste, blev anmeldelserne kategoriseret herefter. I henholdsvis 75% og 39% af tilfældene var det ikke muligt at foretage en kategorisering, da disse tekster indeholdt lige mange positive og negative ord. I de tilfælde, hvor det var muligt at træffe en afgørelse, var henholdsvis 58% og 64% korrekt kategoriseret.

Forskerne bag forsøget lavede dernæst deres egne lister, som skulle bruges til kategoriseringen. De gjorde dette ved kort at kigge på frekvensdistributionen, og herefter valgte de syv positive og syv negative ord (herunder tegn) ud. Disse så således ud:

Tabel 5.2

	Foreslåede ordlister	Korrekt	Uafgjort
Person 1	Positive ord: *love, wonderful, best, great, superb, still, beautiful* Negative ord: *bad, worst, stupid, waste, boring, ?, !*	69%	16%

Som det ses, var det kun i 16% af tilfældene ikke muligt at træffe en afgørelse på baggrund af de udvalgte ord, og den overordnede testscore på 69% var også højere end testpersonernes. At andelen af ikke-kategoriserede anmeldelser for testpersonerne var så høj, skyldtes, at deres udvalgte ord simpelthen ikke var til stede i teksterne – altså situationer, der blev uafgjort med 0-0 positive og negative ord. Dette undgik forskerne netop ved at kigge på frekvensdistributionen. Det gjorde det muligt for dem at vælge ord, der forekom mange gange, hvilket sandsynliggjorde, at disse ord var repræsenteret i mange af anmeldelserne. Dermed undgik de i højere grad 0-0-situationer. At ordene skulle forekomme mange gange var dog ikke eneste kriterium, og de har givetvis sorteret såkaldte stopord fra. Mere om disse senere.

Hvad intuition angår, er der næppe nogen, der umiddelbart ville udvælge termer som *still, ?* og *!* til at udgøre et fundament for denne form for kategorisering af tekster, men resultaterne lyver altså ikke – de fungerer som indikatorer for disse filmanmeldelsers polaritet.

Konklusionen på dette lille eksperiment er, at det giver rigtig god mening at inddrage korpusbaserede teknikker, når det kommer til at udvælge gode indikatorer i forbindelse med opinion mining.

5.3 Semantisk orientering ud fra semantisk association

Konstruktion af ordlister på baggrund af frekvensdistributioner og korpusdata virker altså, og der findes metoder, som er langt mere avancerede, end det var tilfældet i det beskrevede forsøg. I et forsøg af Turney

& Littman (2002) bestemmer de en række ords semantiske orientering – er ordet positivt eller negativt - ud fra ordenes semantiske association, altså ud fra hvilke ord, de gerne optræder sammen med.

5.3.1 Paradigmer af semantisk orientering

Der bliver taget udgangspunkt i to lister, som fungerer som paradigmer af positiv og negativ semantisk orientering. Hver liste indeholder syv ord, og for hvert ord i den positive liste findes et modsvarende ord i den negative: *good/bad, nice/nasty, excellent/poor,* etc.:

Positiv: {*good, nice, excellent, positive, fortunate, correct, superior*}
Negativ: {*bad, nasty, poor, negative, unfortunate, wrong, inferior*}

Disse lister er fremstillet på baggrund af forskernes intuition. Når et givent ords semantiske orientering skal bestemmes, bliver der beregnet en værdi, som indikerer hvor stærkt ordet er associeret med ordene i den positive liste. Fra dette tal trækkes den beregnede værdi af, hvor stærkt ordet er associeret med ordene i den negative liste. Dette tal beskriver således både ordets semantiske orientering og hvor stærkt, det er orienteret.

Det er altså ud fra ordenes semantiske associering, at den semantiske orientering skal beregnes. Der findes forskellige metrikker, når semantiske associering skal beregnes – en af disse er PMI-IR, som står for *Pointwise Mutual Information and Information Retrieval.*

5.3.2 PMI-IR

Vi ser på PMI – Pointwise Mutual Information. To ords PMI er defineret således:

$$\text{PMI}(ord_1, ord_2) = \log_2 \frac{\text{p}(ord_1 \cap ord_2)}{\text{p}(ord_1)\text{p}(ord_2)}$$

$\text{p}(ord_1 \cap ord_2)$ er sandsynligheden for, at ord_1 og ord_2 optræder sammen. Hvis ord_1 og ord_2 er statistisk uafhængige[7], er sandsynligheden for, at de optræder sammen lig $\text{p}(ord_1)\text{p}(ord_2)$. Forholdet mellem $\text{p}(ord_1 \cap ord_2)$ og $\text{p}(ord_1)\text{p}(ord_2)$ er således et mål for hvor statistisk afhængige, de to ord er af hinanden. $\log_2$ af dette forhold er den værdi, vi tilskriver informationen, vi får om tilstedeværelsen af det ene ord, når vi observerer det andet.

Funktionen SO-PMI-IR , som beregner den semantiske orientering af ordet *ORD,* er defineret således:

$$\text{SO-PMI-IR}(ORD) = \text{PMI}(ORD, \{\text{positiv-listen}\}) - \text{PMI}(ORD, \{\text{negativ-listen}\})$$

PMI-IR estimerer PMI ved at sende forespørgsler til en søgemaskine, hvilket er anledningen til IR-delen af PMI-IR. Antallet af søgeresultater – dvs. dokumenter, der matcher søgestrengen – bliver herefter gemt. Forskerne benyttede sig af den på daværende tidspunkt foretrukne søgemaskine AltaVista, hvilket gjorde det muligt at benytte dennes NEAR-operator i forespørgslerne. Brugen af NEAR-operatoren restringerer søgeresultaterne til sider, som indeholder søgeordene i en maksimal afstand af 10 ord fra hinanden. Forskerne lader dette udgøre det kriterium, som bestemmer, om to ord optræder sammen.

[7] Statistisk uafhængighed er defineret således: Lad P være et sandsynlighedsmål. Hvis $\text{P}(\text{A}|\text{B}) = \text{P}(\text{A})$, er hændelsen A uafhængig af, om hændelsen B har indtruffet. Det omvendte gør sig naturligvis også gældende, og at to hændelser – i vores tilfælde ord – er uafhængige, kan skrives således: $\text{P}(\text{A} \cap \text{B}) = \text{P}(\text{A}) \cdot \text{P}(\text{B})$ (Sørensen, definition. 1.5.1)

Funktionen hits(*forespørgsel*) er antallet af returnerede resultater af søgestrengen *forespørgsel*. På baggrund af ovenstående ligninger for PMI(ord_1, ord_2) og SO-PMI-IR(ORD) kan vi nu omskrive SO-PMI-IR(ORD) således:

$$\text{SO-PMI-IR}(ORD) = \log_2 \frac{\text{hits}(ORD \text{ NEAR } p\_forespørgsel)\text{hits}(n\_forespørgsel)}{\text{hits}(ORD \text{ NEAR n}\_forespørgsel)\text{hits(p}\_forespørgsel)}$$

hvor

$$p\_forespørgsel = (good \vee nice \vee \ldots \vee superior)$$

$$n\_forespørgsel = (bad \vee nasty \vee \ldots \vee inferior)$$

hits(ORD NEAR $p\_forespørgsel$) er antallet af sider, hvor *ORD* optræder sammen med *good* eller *nice* eller *excellent* osv. Tilsvarende gør sig naturligvis gældende for ORD NEAR $n\_forespørgsel$.

For at undgå division med 0 lægges 0.01 til antallet af alle hits. Grundet $\log_2$ kan vi tillade os at tilskrive ordet *ORD* en positiv semantisk orientering, hvis SO-PMI-IR(*ORD*) er positiv og en negativ semantisk orientering, hvis SO-PMI-IR(*ORD*) er negativ. Den numeriske værdi er udtryk for hvor stærkt orienteret ordet *ORD* er mod den pågældende semantiske retning.

Forskerne benyttede som sagt algoritmen på søgemaskinen Alta Vista, som på daværende tidspunkt havde indekseret omkring 350 millioner engelsksprogede websider. De anslog – lidt forsigt – at hver side indeholdt 300 ord, hvilket betød, at de i praksis kørte algoritmen på et korpus af størrelsen 100 milliarder ord.

5.3.3 Test af algoritmen

Testen af SO-PMI-IR-algoritmen gjorde brug af et leksikon bestående af 3596 ord. 1614 af disse ord var klassificeret som positive, mens 1982 ord var klassificeret negative.

Tabel 5.3 Eksempler på de positive og negative ord, der blev testet på

Positive ord		Negative ord	
Abide	*Absolve*	*Abandon*	*Abhor*
Ability	*Absorbent*	*Abandonment*	*Abject*
Able	*Absorption*	*Abate*	*Abnormal*
Abound	*Abundance*	*Abdicate*	*Abolish*

Ordenes semantiske orientering blev beregnet af SO-PMI-IR-algoritmen, og ordene blev sorteret i faldende orden i forhold til graden af deres semantiske orientering. Resultatet var som følger:

Tabel 5.4 Testresultater fra kørsel af SO-PMI-IR på et 100 milliarder ord stort korpus

Topandel af leksikonnet	Antal ord	Andel korrekte klassifikationer
100%	3596	79.70%
75%	2697	86.43%
50%	1798	90.04%
25%	899	92.21%

Med ”Topandel af leksikonnet” menes den del af leksikonnets ord, som bliver rangeret højest i forhold til den numeriske værdi af deres semantiske orientering. Tabel 5.4 viser, at når testen blev udført på det fulde leksikon, tilskrev SO-PMI-IR-algoritmen ordets dets korrekte semantiske orientering (i forhold til den orientering, der står listet i leksikonnet) i 79.70% af tilfældene. Testede man kun på de 25% højst rangerede ord – dvs. de ord, som var allermest subjektive og længst væk fra at være neutrale - og så bort fra resten, var hele 92.21% klassificeret korrekt.

Resultatet af testen viser altså, at ikke kun er det muligt på mere eller mindre automatisk vis at bestemme et ords semantiske orientering – udelukkende på baggrund af 14 ord, hvis semantiske orienteringer, man antager for kendte og klart entydige - det er tilmed muligt at bestemme *hvor* kraftig denne semantiske orientering er.

5.4 Informationsudtrækning og subjektivitetsanalyse

Omend opinion mining eller subjektivitetsanalyse er en form for *information extraction* (IE) – informationsudtrækning – skelner man også gerne mellem disse to begreber. For hvor subjektivitetsanalyse søger at afdække og udtrække informationer, der har relation til holdninger, meninger og følelsesmæssige forhold – subjektive størrelser - søger informationsudtrækning at udtrække faktuelle informationer. Der er altså fokus på henholdsvis subjektive og objektive udtryk inden for de to områder.

Områderne er dog ikke fjernere fra hinanden, end at resultater og teknikker inden for det ene område med fordel kan benyttes inden for det andet. Wiebe & Riloff viste i et studie overordnet to ting (Wiebe & Riloff, s. 1):

1. IE-teknikker kan benyttes til at lære og genkende subjektivt sprog.
2. Ved at benytte subjektivitetsanalyse kan man forbedre resultaterne inden for faktabaseret IE.

5.4.1 Subjektivitetsanalyse forbedret gennem IE

Dette foregår gennem brug af let superviserede IE-læringsalgoritmer, som automatisk generer lister af subjektive termer og udtryk fra ikke-annoterede tekster. Fokus er på substantiver, der bærer på subjektive konnotationer samt flerords-udtryk, der udtrykker subjektivitet.

Wiebe og Riloff mener, at opinion mining kan forbedres, hvis man inddrager et system, der kan skelne mellem subjektive og objektive sætninger. For det første kan udtryk være subjektive uden at have nogen semantisk orientering. For det andet har det vist sig, at det er vanskeligere at skelne mellem subjektive og objektive udtryk, end det er efterfølgende at bestemme den semantiske orientering.

5.4.1.1 Baggrunden for studiet

Af de forskningsstudier, der inspirerede til Wiebes og Riloffs eksperimenter, var Hatzivassiloglou & McKeowns (1997) de første. De udnyttede den semantiske afhængighed, som eksisterer blandt adjektiver brugt i forbindelse med konjunktioner. Med afhængighed menes, at hvis <adjektiv_1> i konstruktionen "<adjektiv_1> og <adjektiv_2>" er positivt orienteret, så er <adjektiv_2> det nødvendigvis også. Således kunne de gøre brug af "adjektiv-kendt-som-positiv og <ukendt-adjektiv>" som udtrækningsmønster til at finde positive adjektiver, og det samme gør sig selvfølgelig gældende for negative adjektiver. Senere fulgte Turney (Turney 2002), Gamon & Aue (2005) og Kanayama & Nasukawa (2006) op med arbejde relateret hertil.

5.4.1.2 Udgangspunkt for forsøget

Wiebe og Riloff tager udgangspunkt i et subjektivitetsleksikon – dvs. en liste af ord, der alle udtrykker subjektivitet i forskellige grader – og en samling af udtræksmønstre. Herudover kræves en smule menneskelig supervision og gennemgang af fremkomne ordlister etc.

5.4.1.2.1 Udtrækningsmønstre

En vigtig del af eksperimentet er udtrækningsmønstre. Et udtrækningsmønster er et leksikalsk-syntaktisk mønster, som repræsenterer et eller flere ord i en specifik syntantisk kontekst. Et eksempel herpå er "<adjektiv_1> og <adjektiv_2>" som beskrevet ovenfor. Teksten, der skal tjekkes efter for matches med et givent udtrækningsmønster, parses og mærkes op med hensyn til syntaktiske konstituenter (dvs. NP'er, VP'er, PP'er, etc.). Således vil udtrækningsmønsteret "<adjektiv> og sjov" finde et match i teksten "hun er smuk og sjov", mens det ikke matcher noget i teksten "der var popcorn og sjov og ballade", da *popcorn* ikke lever op til udtrækningsmønsterets krav om at være et adjektiv.

Wiebe og Riloff gør brug af følgende syntaktiske skabeloner, som bruges til at konstruere udtrækningsmønstre:

Tabel 5.5 Syntaktiske skabeloner

<subj> passive-verb <subj> active-verb <subj> active-verb dobj <subj> verb infinitive <subj> aux noun
active-verb <dobj> infinitive <dobj> verb infinitive <dobj> noun aux <dobj>
noun prep <np> active-verb prep <np> passive-verb prep <np> infinitive prep <np>

5.4.1.3 Konstruktion af subjektive ord samt udtrækningsmønstre

Arbejdet består nu af to dele:

1. Gennem udnyttelse af udtrækningsmønstre findes gennem bootstrapping kontekster, der ofte indeholder subjektive substantiver. Derigennem konstrueres en list af substantiver, der gerne bruges subjektivt.
2. På baggrund af ovenstående liste konstrueres en liste af udtrækningsmønstre, som er korreleret med subjektivitet i et korpus.

Del 1 gør brug af et allerede eksisterende subjektivitetsleksikon (dette er de såkaldte *seed words* eller seed-ord) og de syntaktiske skabeloner i Tabel 5.5. To bootstrapping-algoritmer (Meta-Bootstrapping og Basilisk) løber et ikke-annoteret korpus (benævnt bootstrapping-korpusset) igennem og tjekker hvilke mønstre, seed-ordene optræder i. Hypotesen er nu, at ord, som optræder i de samme mønstre som seed-ordene, tilhører den samme semantiske kategori.

5.4.1.4 Subjektivitetsleksikonnet

For at konstruere det subjektivitetsleksikon, som udgør seed-ordene, bootstrapping-algoritmerne gør brug af, bliver der til at starte med fundet 850 substantiver, der er korreleret med subjektive sætninger, i et andet datasæt, der på sætningsniveau er opmærket, hvad subjektivitet angår. For hvert af de nævnte 850 substantiver bliver det tjekket, om det pågældende ord optræder i bootstrapping-korpusset. Hvis det er tilfældet, bliver det føjet til en liste, som efterfølgende bliver ordnet efter hvor hyppigt ordene optræder i bootstrapping-korpusset, og manuelt udvælger forskerne 20 hyppigt forekommende ord, som de mener er stærkt subjektive.

5.4.1.5 Bootstrapping

Hver bootstrapping-algoritme løber korpusset igennem 400 gange, og producerer i alt 3996 ord (en del af disse ord er ens, da begge algoritmer producerer dem). Disse ord bliver af en menneskelig supervisor tjekket efter og klassificeret som *StrongSubjective, WeakSubjective* eller *Objective*. I alt producerer de to algoritmer 1052 subjektive ord.

For at måle kvaliteten af de 1052 ord og hvor effektive de to bootstrapping-algoritmer er, bliver ordene holdt op mod MPQA-korpusset[8], der på sætningsniveau er opmærket, hvad subjektivitet angår. Dette sker ved at evaluere en simpel classifier, som består af den ene regel, at hvis en sætning indeholder et af de 1052 ord, så bliver sætningen klassificeret subjektivt. Classifieren bliver evalueret på præcision (SubjPrec) og recall (SubjRecall), og resultaterne er som følger:

Tabel 5.6

	SubjRecall	SubjPrec
Basilisk StrongSubj	2.4	86.8
Basilisk WeakSubj	4.6	75.9
Meta-Bootstrap StrongSubj	13.9	83.1
Meta-Bootstrap WeakSubj	36.9	72.6

Evalueringsdata består af 9732 sætninger, hvoraf 5380 (55%) er subjektive. Præcisionen er således ganske høj for alle fire sæt af subjektive ord.

5.4.1.6 Leksikalsk-syntaktiske mønstre som repræsentanter for subjektive udtryk

De 1052 subjektivitetsbærende ord fra forrige afsnit indgår i konstruktionen af to high-performance classifiers: HP-Subj og HP-Obj. Disse classifiers skal bruges til at producere to datasæt – ét indeholdende subjektive sætninger og ét indeholdende objektive sætninger – ud fra et korpus af ikke-annoteret tekst. Datasættene skal udgøre træningsdata for maskinlæringsalgoritmen AutoSlog-TS (Riloff 1996), som automatisk kan identificere udtrækningsmønstre, der er korreleret med subjektiv tekst. For at træne AutoSlog-TS er der ikke brug for annoteret data, men den skal til gengæld fodres med relevante (subjektive) og irrelevante (objektive) teksteksempler. Det er disse teksteksempler, de to classifiers HP-Subj og HP-Obj skal udvælge.

En af Wiebe og Riloffs hypoteser er, at udtrækningsmønstrene vil kunne repræsentere subjektive udtryk, som har ikke-kompositionelle betydninger. Se fx udtrykket "drives (someone) up the wall", som udtrykker

[8] http://mpqa.cs.pitt.edu/

følelsen at blive generet af noget. Udtrykkets betydning adskiller sig meget fra de enkelte ords betydninger. Udtrykket "<x> drives <y> up the wall" er desuden en ganske fleksibel konstruktion, der kan indtage et utal af former, og det gør det svært at registrere disse udtryk via N-grammer: Fx vil "George drives me up the wall" og "the nosy old man drives his quiet neighbors up the wall" ikke blive fanget af det samme N-gram. Dette vil kunne lade sig gøre med udtrækningsmønstre.

5.4.1.7 High-performance classifiers

De to classifiers HP-Subj og HP-Obj gør brug af en række leksikalske udtryk, som er gode indikatorer for subjektivt sprogbrug – såkaldte *subjectivity clues* eller subjektivitetsclues. De forskellige clues består af ord og N-grammer, men ingen syntaktiske generaliseringer som udtrækningsmønstre. Listen af clues er sammensat af manuelt fremstillede kilder, bl.a. adjektiver, der er opmærkede, hvad polaritet angår, andre ord er trukket ud af forskellige korpora, herunder de 1052 ord i 5.4.1.5. De forskellige subjektivitetsclues bliver inddelt i *stærkt subjektive* og *svagt subjektive* clues.

5.4.1.7.1 Definition

HP-Subj og HP-Obj bliver nu defineret således:

- HP-Subj klassificerer en sætning som 'subjektiv', hvis sætningen indeholder to eller flere stærkt subjektive clues.
- HP-Obj klassificerer en sætning som 'objektiv', hvis sætningen ikke indeholder nogle stærkt subjektive clues. Desuden må sætningen selv og de to tilstødende sætninger maksimalt indeholde to svagt subjektive clues tilsammen.

Begge classifiers er altså defineret ud fra tilstedeværelsen af subjektive clues. Der findes clues, der helt klart er objektive, men de kan ikke bruges til at klassificere ud fra. For hvis der bare er et subjektivt clues i samme sætning, bliver hele sætningen ligeledes subjektiv.

5.4.1.8 Evaluering af HP-Subj og HP-Obj

HP-Subj og HP-Obj bliver evalueret på MPQA-korpusset, og de scorer som følger:

Tabel 5.7 Resultater med bootstrap-clues

	Precision	Recall	F-score
HP-Subj	91.7	30.9	46.2
HP-Obj	83	32.8	47.1

En del af formålet med forsøget er at teste hvor effektive HP-Subj og HP-Obj er, når de ikke gør brug af de 1052 ord, som blev fundet gennem bootstrap-processerne i 5.4.1.5. Når disse ord ikke medtages i den samling af subjektivitets-clues, de to classifiers gør brug af, er resultaterne således ud:

Tabel 5.8 Resultater uden bootstrap-clues

	Precision	Recall	F-score
HP-Subj	93.3	23.3	37.7
HP-Obj	73.6	43.3	54.5

Som det fremgår af Tabel 5.8, stiger precision for HP-Subj en anelse med 1.6 procentpoint. Til gengæld er der et betragteligt fald i recall med 7.6 procentpoint, hvilket viser, at de ord, bootstrapping-algoritmerne producerede, repræsenterer, bidrager med ny, brugbar viden.

For HP-Obj's vedkommende falder precision med hele 9.4 procentpoint, mens recall stiger med 10.5 procentpoint. Da formålet med de to classifiers jo er at konstruere træningsdata til AutoSlog-TS-algoritmen, og dette skal være af en vis kvalitet, ville HP-Obj således ikke kunne bruges, hvis de ord, bootstrap-algoritmen producerede, ikke indgik i samlingen af subjektivitets-clues.

5.4.1.9 Produktion af træningsdata

HP-Subj og HP-Obj bliver efter test sluppet løs på et ikke-annoteret korpus bestående af 298.954 sætninger. HP-Subj klassificerer 48.814 sætninger subjektivt, mens HP-Obj klassificerer 68.580 sætninger objektivt. Således bliver 117.394 sætninger - lige knap 40% - høstet, og de skal bruges til træning af AutoSlog-TS.

5.4.1.10 Træning af AutoSlog-TS

Når AutoSlog-TS skal trænes, skal den fodres med et sæt relevant og et sæt irrelevant data, dvs. hhv. subjektive og objektive sætninger. Træningen består nu af to trin:

5.4.1.10.1 Generering af udtrækningsmønstre

I det første trin bliver korpusset tjekket efter for hvilke tekststykker, der matcher de syntaktiske skabeloner i Tabel 5.5. Matcher skabelonen en tekst, bliver der på baggrund af dette genereret et udtrækningsmønster. Eksempelvis matcher skabelonen *<subj> passive-verb* teksten *"he was satisfied"*, hvilket genererer udtrækningsmønsteret *<subj> was satisfied*. Dette gøres for samtlige syntaktiske skabeloner.

5.4.1.10.2 Applicering af udtrækningsmønstre

I træningens andet trin bliver hvert genererede udtrækningsmønster appliceret på træningskorpusset i den forstand, at det bliver opgjort hvor mange gange hvert udtrækningsmønster optræder i en subjektiv versus en objektiv sætning. Normalt ville man i brugen af AutoSlog-TS-algoritmen nu gøre brug af en menneskelig supervisor til på baggrund af statistikken at afgøre, hvilke mønstre, der skal beholdes, og hvilke, der skal kasseres.

Men da Wiebe og Riloff ønsker, at processen skal være fuldautomatisk, benytter de sig ikke af dette menneskelige tjek. I stedet rangerer de udtrækningsmønstrene vha. et betinget sandsynlighedsmål, nemlig sandsynligheden for, at en sætning er subjektiv, givet at det pågældende udtrækningsmønster optræder i sætningen. Dette sker således:

$$P(\text{subjektiv\_sætning}|\text{mønster\_i}) = \frac{\text{subj\_frekvens(mønster\_i)}}{\text{frekvens(mønster\_i)}}$$

subj_frekvens(mønster_i) angiver hvor mange subjektive sætninger, **mønster_i** optræder i, mens **frekvens(mønster_i)** angiver hvor mange sætninger - både subjektive og objektive - **mønster_i** i alt optræder i.

θ-parametrene i det følgende kan varieres, og Wiebe og Riloff bestemmer, at hvis et mønster opfylder

$$\text{frekvens}(\text{mønster}) \geq \theta_1 = 5$$

og

$$P(\text{subjektiv\_sætning}|\text{mønster}) \geq \theta_2 = 0.95$$

er det stærkt associeret med subjektivitet i træningsdata - hvis ikke, bliver det kasseret. I alt bliver 8490 udtrækningsmønstre associeret med subjektivitet udvalgt med disse parametre.

For at finde udtrækningsmønstre, der er associeret med objektivitet, trækkes de mønstre ud, der er negativt korreleret med subjektive sætninger. Dette gøres ved at sætte $\theta_1 = 5$ og $\theta_2 = 0.15$. Dermed er de mønstre, der er korreleret med objektive sætninger, de mønstre for hvilke

$$\text{frekvens}(\text{mønster}) \geq \theta_1 = 5$$

og

$$P(\text{subjektiv\_sætning}|\text{mønster}) \leq \theta_2 = 0.15$$

Dette resulterer i 2910 udtrækningsmønstre, som er associeret med objektivitet.

5.4.1.11 Gode argumenter for et korpusbaseret system

Tabel 5.9 viser eksempler på nogle af de udtrækningsmønstre AutoSlog-TS har lært, hvor mange gange, de optræder, og i hvor stor en procentdel de optræder i subjektive sætninger.

Tabel 5.9

Udtrækningsmønster	Frekvens	% Subjektive sætninger
<subj> was asked	11	100
<subj> asked	128	63
<subj> is talk	5	100
talk of <np>	10	90
<subj> will talk	28	71
<subj> put an end	10	90
<subj> put	187	67
<subj> is going to be	11	82
<subj> is going	182	67
was expected from <np>	5	100
<subj> was expected	45	42
<subj> is fact	38	100
fact is <dobj>	12	100

Her kan man bl.a. aflæse, at når *asked* optræder i passivform, er det altid i en subjektiv sætning. Men når *asked* optræder i aktiv, er det kun i 63% af tilfældene i en subjektiv sætning. Og i de sidste to rækker ses det, at samtlige gange, hvor *fact* – som man kunne fristes til at tro var knyttet til objektive sammenhænge – optræder, er det i subjektive sætninger. Disse eksempler illustrerer fint, hvordan korpusbaserede læringsalgoritmer kan finde mønstre, som ikke umiddelbart intuitivt synes subjektive for mennesker, men ikke desto mindre er de gode, pålidelige indikatorer for subjektivitet.

5.4.1.12 Evaluering af udtrækningsmønstre

Evalueringen af de udtrækningsmønstre, AutoSlog-TS producerede, sker på baggrund af det manuelt annoterede MPQA-korpus. Af de sætninger i MPQA, som giver et match med de udtrækningsmønstre, som i 5.4.1.10.2 blev vurderet stærkt associeret med subjektivitet, er 80.7% annoteret som positive – dvs. precision er 80.7%, mens recall er 47.2%. F-score er 59.6%.

At udtrækningsmønstrene reelt bidrager med ny og brugbar viden, fremgår klart, hvis vi kigger på HP-Subj og HP-Obj igen. Hvis de af AutoSlog-TS' udtrækningsmønstre, der er associeret med subjektivitet føjes til listen over stærkt subjektive clues, falder precision for HP-Subj godt nok med 7.3 procentpoint til 84.4%, men recall stiger til gengæld med hele 19.8 procentpoint til 50.7%. For HP-Obj's vedkommende falder precision med 0.5 procentpoint til 82.5%, mens recall stiger med 4.9 procentpoint til 37.7%. Det kan altså kort siges, at ved at medtage udtrækningsmønstrene i listen over subjektivitets-clues, stiger recall for HP-Subj og HP-Obj mere, end precision falder.

5.4.1.13 Konstruktion af Naive Bayes-classifier

Wiebe og Riloff konstruerer en NB-classifier, som kan bruges til at klassificere sætninger efter subjektivitet, og denne classifier skal bruges til at forbedre IE-systemet længere fremme. Der bliver i dette arbejde gjort brug af de subjektivitets-clues og udtrækningsmønstre, som er blevet genereret i de just beskrevne afsnit.

En classifier, som bliver udviklet med maskinlæring, har gerne den fordel i forhold til de regelbaserede classifiers, der hidtil er blevet benyttet (HP-Subj og HP-Obj), at den producerer en højere F-score. Precision falder gerne noget, men recall er til gengæld så meget højere, og dette er særdeles kærkomment i forhold til HP-Subj og HP-Obj, da disse som vist i 5.4.1.9 kun var i stand til at klassificere knap 40% af data.

5.4.1.13.1 En mere omfattende featurevektor

NB-classifieren er mere nuanceret end de regelbaserede classifiers, og den gør brug af flere forskellige former for features. For hver af følgende mængder, er der en feature:

- stærkt subjektive clues
- svagt subjektive clues
- de subjektive udtrækningsmønstre, der blev genereret af AutoSlog-TS i 5.4.1.10.1
- de objektive udtrækningsmønstre, der blev genereret af AutoSlog-TS i 5.4.1.10.1
- en feature for hver af forskellige ordklasser: pronomener, modalverber ("will" undtaget), adjektiver, kardinaltal og adverbier ("not" undtaget)

Hver feature kan indtage værdien 0, 1 eller ≥ 2, alt efter hvor mange gange, den type feature optræder i sætningen. Ligeledes angives det med 0, 1 eller ≥ 2, alt efter hvor mange gange den type feature optræder i sætningen før eller efter.

NB-classifieren gør brug af flere forskellige typer features end de oprindelige HP-Subj og HP-Obj-classifiers, og som beskrevet i 4.4.2 klassificerer den ved hjælp af en probabilistisk model, der er baseret på en kombination af de forskellige features. Således er den potentielt i stand til at klassificere en større og mere divergerende mængde annoterede sætninger, end HP-Subj og HP-Obj er.

5.4.1.14 Applicering af NB-classifieren

NB-classifieren skal jo i IE-systemet bruges til at dømme sætninger, som er subjektive, ude. Men for ikke at fare for aggresivt frem og fjerne sætninger, der slet ikke burde fjernes, lader Wiebe og Riloff classifieren klassificere de 90% af sætningerne, som den er mest sikker på – dette gælder både subjektive og objektive sætninger. De sidste 10% undlader de at klassificere, hvorfor disse sætninger forbliver i teksten, hvorfra IE-systemet skal hive informationer ud af. Hvor sikker classifieren er (*measure of confidence*), måles således:

$$CM = \left| \log\big(P(subjektiv)\big) + \sum_i \log\big(P(f_i|subjektiv)\big) - \log\big(P(objektiv)\big) + \sum_i \log\big(P(f_i|objektiv)\big) \right|$$

5.4.1.15 Evaluering af NB-classifieren

Som ved tidligere evalueringer, bliver NB-classifieren evalueret ved hjælp af MPQA-korpusset. Classifieren opnår 72.7% subjektiv precision, 84.8% subjektiv recall og 78.3% subjektiv F-score. Dette er holdt op mod scoren for den regelbaserede classifier HP-Subj i 5.4.1.8:

Tabel 5.10

	Precision	Recall	F-score
Naive Bayes	72.7	84.8	78.3
HP-Subj	91.7	30.9	46.2

Et fald i precision, men en markant stigning i recall og F-score.

5.4.2 IE forbedret gennem subjektivitetsanalyse

Efter nu at have set, hvordan teknikker fra *information extraction* kan benyttes til at lære og genkende subjektivt sprog, skal vi nu den anden vej rundt – forbedring af IE-systemer gennem anvendelse af værktøjer fra subjektivitetsanalysen, herunder Naive Bayes-classifieren beskrevet i sidste afsnit.

5.4.2.1 Metaforisk sprog giver falske positiver

Et IE-system løber typisk gennem en teksts sætninger og søger efter informationer, der synes relevante i forhold til det pågældende domæne. En fejlkilde i dette setup kan være sætninger, der udtrykker en form for subjektivitet, og disse sætninger genererer ofte falske positiver, dvs. subjektive tekster, der anskues for relevante i forholdet til domænet, men i virkeligheden ikke er det. Dette kan bl.a. skyldes, at subjektivt sprog ofte gør brug af metaforiske udtryk. Et IE-system, som eksempelvis leder efter informationer relateret til terrorbomber, ville derfor nok bedømme sætningen ”*Manden eksploderede af raseri*” som relevant, selvom den som sådan intet har med terror at gøre. Weibe & Riloffs idé er benytte en classifier, som kan identificere og frasortere subjektive sætninger, hvorefter den faktuelle informationsudtrækning finder sted.

5.4.2.2 MUC-4 terror-korpusset

MUC-4 er et datasæt indeholdende 1700 artikler, hvoraf halvdelen omhandler terrorhandlinger som bomber, kidnapninger og mord. Den anden halvdel af artikler, omhandler ikke terrorhandlinger. Hver artikel har et metadokument tilknyttet, som angiver hvilke informationer relateret til terrorisme, der skal trækkes ud af artiklen. Således er metadokumenterne tilknyttet de irrelevante tekster tomme.

Metadokumenterne er alle opbygget på samme måde. Hver *event role*, IE-systemet forventes at kunne trække ud af teksten, har en plads i metadokumentet. Eksempelvis er der en plads til *PERPETRATORS, VICTIMS, PHYSICAL TARGETS* og *WEAPONS*.

Wiebe og Riloff benytter igen AutoSlog-TS-algoritmen til at generere udtrækningsmønstre for teksterne i MUC-4-datasættet. De relevante tekster, algoritmen bliver trænet på, er teksterne, der har en ikke-tom metatekst tilknyttet (dvs. de, der omhandler terror), mens de irrelevante er de tekster, har en tom meta-tekst tilknyttet. AutoSlog-TS rangerer udtrækningsmønstrene efter hvor brugbare de er i forhold til at trække informationerne ud af teksten. En menneskelig supervisor gennemgår de højest rangerede udtrækningsmønstre og tilskriver hver af disse en event role. Eksempelvis vil mønsteret ”<subject> was killed” trække information om ofre (*victims*) ud af teksten, og således vil indholdet af ”<subject>” blive tilskrevet *VICTIM*.

5.4.2.3 Flere angreb, flere svarark

IE-systemet hiver informationer ud af en given tekst og gemmer disse informationer i såkaldte *answer key templates* (svarark). Bliver flere terrorangreb omtalt i samme tekst, skal IE-systemet spytte et svarark ud for hvert angreb, som hver indeholder informationer om gerningsmænd, ofre, våben osv. relateret til det pågældende angreb. Det er klart, at dette langt fra er en triviel opgave, som bl.a. omfatter diskursanalyse for at kunne bestemme, hvor mange angreb, teksten omtaler, og hvilke informationer, der hører til hvilket angreb og dermed svarark. Og målet for Wiebe og Riloff er at foretage en subjektivitetsanalyse for at fjerne subjektive sætninger, hvilket skal medvirke til at lette diskursanalysen.

5.4.2.4 Evaluering

Der bliver testet og evalueret på de fire event roles i svararket, som er nævnt ovenfor: *perpetrators, victims, physical targets* og *weapons*.

5.4.2.4.1 IE - Baseline

Uden subjektivitetsanalyse opnår IE-systemet 52% recall med 42% precision, hvilket giver en F-score på 47%. Dette er baseline.

5.4.2.4.2 IE+Subj

Forskerne lægger ganske aggresivt ud med at frasortere alle informationer, der stammer fra sætninger, som NB-classifieren har vurderet til at være subjektive. 94 forkerte informationsudtrækninger bliver hermed fjernet, hvilket forårsager en forbedret precision på 2 procentpoint til 44%. Men da 48 korrekte udtrækninger også bliver fjernet, falder recall med 8 procentpoint til 44%, hvilket resulterer i en F-score på 44%. Forskernes idé om, at mange forkerte informationsudtrækninger finder sted i forbindelse med subjektive sætninger, ser ud til at holde, men da strategien med at fjerne samtlige subjektive sætninger alligevel synes lidt for aggresiv, ser de nærmere på, hvordan subjektive størrelser kan figurere i samme sætning som objektive og informationsbærende elementer.

5.4.2.4.3 IE+Subj_Attr

Wiebe og Riloff konstaterer på baggrund af første kørsel, at mange sætninger, hvor en kilde bliver angivet, gerne indeholder faktuelle informationer, men de bliver ofte klassificeret subjektivt. Det er sætninger som ”*The Associated Press reported...”* og ”*The President stated...”*, men foruden ofte at bære på faktuelle informationer, tilskriver de ofte en subjektiv ytring til nogen eller noget, fx ”*CNN reported that Bush praised*

his Attorney General for...". Men da en kildeangivelse altså ofte indeholder vigtige informationer, dvs. de informationer et IE-system skal trække ud, vælger Wiebe og Riloff at undtage visse subjektivt klassificerede sætninger. Disse skal opfylde følgende betingelser:

1. Sætningens CM (beskrevet i 5.4.1.14) skal være ≤ 25, dvs. sætningen er kun ganske svagt subjektiv.
2. Sætningen skal indeholde et af følgende verber: *affirm, announce, cite, confirm, convey, disclose, report, tell, say, state*.

Dette forbedrer recall med 2 procentpoint til 46%, precision forbliver på 44%, og F-score er 45%.

5.4.2.4.4 IE+Subj_Attr_Slct

For at lave en mere selektiv bortfiltrering gør Wiebe og Riloff brug af begreberne *indicator patterns* (indikatormønstre) og *nonindicator patterns* (nonindikatormønstre). I mange sætninger optræder fakta og subjektive ytringer sammen, så det går ikke an at se bort fra alle sætninger, der indeholder subjektivt sprog. Eksempelvis indeholder sætningen "*He was outraged by the terrorist attack on the World Trade Center*" , der indeholder den stærkt subjective term *outraged*. Men sætningen indeholder også den klare kendsgerning, at der har været et terrorangreb på WTC. Indikatormønstre repræsenterer udtryk, som næsten utvivlsomt indeholder informationer, der har relevans for IE-systemet. Eksempelvis vil indikatormønstrene "*murder of <NP>*" og "*<NP> was assassinated*" uafhængigt af konteksten næsten altid identificere mordofre. Omvendt afhænger relevansen af nonindikatormønstre af konteksten: "*<NP> was arrested*" og "*injured by <NP>*". Disse mønstre kan potentielt trække informationer om terrorister og gerningsmænd, men det afhænger altså af konteksten.

Wiebe og Riloff benytter igen AutoSlog-TS til at generere en liste over indikator- og nonindikatormønstre. Hvis de gør brug af indikatormønstre i forbindelse med filtreringen, falder recall med 12 procentpoint til 40% i forhold til baseline på 52%. Således må nonindikatormønstrene også hive en del relevante informationer ud af teksterne. Systemet bliver nu lavet om, så informationer fra indikatormønstre aldrig bliver bortfiltrerede, mens information fra nonindikatormønstre bliver fjernet, hvis de optræder i en subjektiv sætning. Dette giver en recall på 51% og en precision på 45%.

5.4.2.4.5 IE+Subj_Attr_Slct+SubjEP

Udtrækningsmønstrene, der ligger til grund for IE-systemet, er blevet gennemgået manuelt. Kvaliteten skulle således være høj, men det er svært at sige hvilke mønstre, der er gode til at identificere subjektive sætninger. Wiebe og Riloff løber træningsdata igennem og lader NB-classifieren (som vurderer, om en sætning er subjektiv eller objektiv) klassificere hver enkelt sætning. Dernæst tælles hvor mange gange nonindikatormønstrene optræder i subjektive vs objektive sætninger. Hvis et mønster opfylder P(subjektiv|mønster_i)>0.5 og det optræder mere end 10 gange i træningssættet, bliver det vurderet til at være subjektivt.

Næste og sidste inkarnation af IE-systemet består nu i også at ignorere alle informationer, som er trukket ud via disse subjektive mønstre. Dette giver en forbedring af precision på 4 procentpoint i forhold til baseline, så den ender på 46%, mens recall går ganske svagt ned med 1 procentpoint og ender på 51%.

5.4.2.5 IE forbedret via subjektivitetsanalyse

Skønt forbedringerne ikke er voldsomme, synes det alligevel som en god idé at gøre brug af subjektivitetsanalyse i forbindelse med informationsudtrækning, og omkostningerne er ganske små i forhold til gevinsten. Helt konkret resulterede subjektivitetsanalysen i, at 62 ud af 367 ukorrekte informationsudtrækninger blev fjernet, mens der kun blev fjernet 8 ud af 266 korrekte udtrækninger.

Efter i dette afsnit at have gennemgået forskellige metoder til subjektivitetsanalyse, vil jeg nu se nærmere på det datasæt, der ligger til grund for classifieren beskrevet i afsnit 7.

6 Data

Dansk er et sprog, der stort set kun bliver talt i Danmark, hvilken i sagens natur gør det til et relativt lille sprog. Dette er en udfordring i forbindelse med sprogteknologisk arbejde med dansksproget materiale. I modsætning til engelsk, hvor størstedelen af forskningen omkring opinion mining foregår, findes der nemlig ikke umiddelbart tilgængeligt annoteret dansk materiale og andre værktøjer, man kan gøre brug af.[9]

Man kan dog hurtigt opnå gode resultater gennem arbejde med annoteret data, men denne type data er også dyre at fremstille eller fremskaffe, og desuden findes der slet ikke annoteret data fra alle domæner.

6.1 Nykredit-korpusset

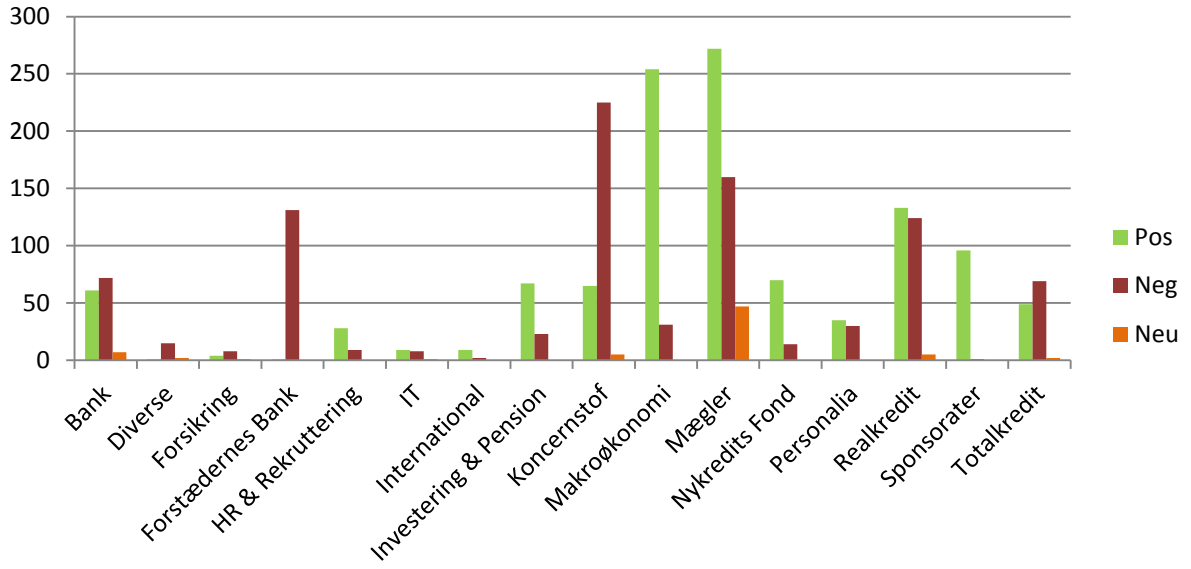
Data til dette projekt er leveret af Infomedia via Ankiro. Det består af et Excel-ark indeholdende 2146 tekster i xml-format, som alle omhandler forskellige aspekter af Nykredits virksomhed. Hver xml-træ indeholder en artikel og en mængde metadata.[10] Fra hver artikel benyttes brødteksten til klassificering, og 'QualitativeScore', der kan antage værdierne -1, 0 og 1, benyttes til evaluering af classifieren. Data er blevet manuelt evalueret af medarbejdere hos Infomedia, og 1147 tekster er klassificeret positivt, 929 er klassificeret negativt og 70 tekster er klassificeret neutralt. Teksterne er opdelt i 16 kategorier:

Tabel 6.1

Kategori	I alt	Pos	Neg	Neu
Bank	140	61	72	7
Diverse	18	1	15	2
Forsikring	13	4	8	1
Forstædernes Bank	132	1	131	0
HR & Rekruttering	37	28	9	0
IT	18	9	8	1
International	11	9	2	0
Investering & Pension	90	67	23	0
Koncernstof	295	65	225	5
Makroøkonomi	285	254	31	0
Mægler	479	272	160	47
Nykredits Fond	84	70	14	0
Personalia	65	35	30	0
Realkredit	262	133	124	5
Sponsorater	97	96	1	0
Totalkredit	120	49	69	2
I alt	**2146**	**1154**	**922**	**70**

[9] Der findes for så vidt data på dansk, der er annoteret, som vi har brug for i forhold til opinion mining – dvs. tekst med tilknyttet tag eller rating, fx i form af produktanmeldelser på sider som www.Trustpilot.dk og filmanmeldelser på sider som www.Scope.dk. Disse data er dog ikke validerede, og jeg har i dette projekt valgt at se bort fra dem.

[10] Se bilag A



Figur 6.1

6.2 Preprocessing

Datasættet indeholder dubletter i den forstand, at samme artikel er blevet bragt i flere medier og dermed talt med flere gange.

6.2.1 Dubletter og neutrale artikler bortfiltreres

119 artikler figurerer flere gange i datasættet - 70 klassificeret positive, 44 klassificeret negative og 5 klassificeret neutrale – og hver af disse optræder mellem 2 og 10 gange. Af de i alt 329 dubletter er 186 positive, 128 er negative og 15 er neutrale. Disse bliver alle fjernet, da termerne i disse artikler ellers ville blive talt med flere gange, hvilket ville være en kilde til støj i træningen af classifieren.

For at forenkle arbejdet fjerner jeg yderlige de 55 resterende artikler, der er klassificeret neutrale, og dette bringer antallet af artikler ned på 1762, hvoraf 961 er positive og 801 er negative.

6.2.2 Semidubletter

Foruden de tekster, som er fuldstændig identiske, optræder der også en lang række tekster, som *næsten* er identiske. Jeg benævner disse *semidubletter*. Følgende tekst handler i korte træk om, at forbrugerne går slukørede på juleferie, mens en økonom i Danske Bank oplyser, at det ikke bliver nogen jubeljul:

> *KØBENHAVN. Der er færre gaver under træet til jul hos hver femte familie i år.En ny meningsmåling foretaget af Greens Analyseinstitut for dagbladet Børsen viser, at hver femte familie regner med at bruge færre penge på julegaver i år sammenlignet med sidste år, som var en rigtig sparejul i kølvandet på finanskrisen.*
>
> *-Det kan umiddelbart overraske, at der bliver skåret lidt ned på julegaverne i år, for danskerne har ellers fået flere penge mellem hænderne i 2010. Men det tyder på en øget krisebevidsthed hos danskerne, siger cheføkonomen i Nykredit, John Madsen til Børsen.*

Krisebevidstheden kunne i går også tydeligt aflæses i de nye tal for forbrugertilliden i december fra Danmarks Statistik. Humøret faldt til under nulpunktet hos forbrugerne op til julehandlen -minus 0,7 mod 3,7 i november.

-Forbrugerforventningerne indikerer, hvordan det står til med privatforbruget, og i særlig grad med detailhandlen, og det er nedslående for privatforbruget at se, at forbrugerne går en smule slukørede på juleferie, vurderer Mira Lie Nielsen, økonom i Dansk Erhverv.

Den nye Greens-måling viser, at kun 15 procent af danskerne planlægger at bruge flere penge på julegaver. Et stort flertal af danskerne, 63 procent, forventer et uændret gavebudget. Undersøgelsen er foretaget blandt 941 repræsentativt udvalgte danskere.

-Det bliver ingen jubeljul, og vi kommer ikke i nærheden af julen 2007, siger Las Olesen, privatøkonom i Danske Bank.

/ ritzau/.[11]

Et andet sted i korpusset finder vi følgende sætning, som bortset fra den første, ekstra sætning og det fraværende, afsluttende "/ ritzau/." er lig ovenstående tekst:

Der er sparejul i hver femte danske familie, hvor der vil blive brugt færre penge end sidste år.Der er færre gaver under træet til jul hos hver femte familie i år.En ny meningsmåling foretaget af Greens Analyseinstitut for dagbladet Børsen viser, at hver femte familie regner med at bruge færre penge på julegaver i år sammenlignet med sidste år, som var en rigtig sparejul i kølvandet på finanskrisen.

- Det kan umiddelbart overraske, at der bliver skåret lidt ned på julegaverne i år, for danskerne har ellers fået flere penge mellem hænderne i 2010. Men det tyder på en øget krisebevidsthed hos danskerne, siger cheføkonomen i Nykredit, John Madsen til Børsen.

Krisebevidstheden kunne i går også tydeligt aflæses i de nye tal for forbrugertilliden i december fra Danmarks Statistik. Humøret faldt til under nulpunktet hos forbrugerne op til julehandlen - minus 0,7 mod 3,7 i november.

- Forbrugerforventningerne indikerer, hvordan det står til med privatforbruget, og i særlig grad med detailhandlen, og det er nedslående for privatforbruget at se, at forbrugerne går en smule slukørede på juleferie, vurderer Mira Lie Nielsen, økonom i Dansk Erhverv.

Den nye Greens-måling viser, at kun 15 procent af danskerne planlægger at bruge flere penge på julegaver. Et stort flertal af danskerne, 63 procent, forventer et uændret gavebudget. Undersøgelsen er foretaget blandt 941 repræsentativt udvalgte danskere.

- Det bliver ingen jubeljul, og vi kommer ikke i nærheden af julen 2007, siger Las Olesen, privatøkonom i Danske Bank.[12]

[11] Artikel fra bilag A med id-koden e25d2484

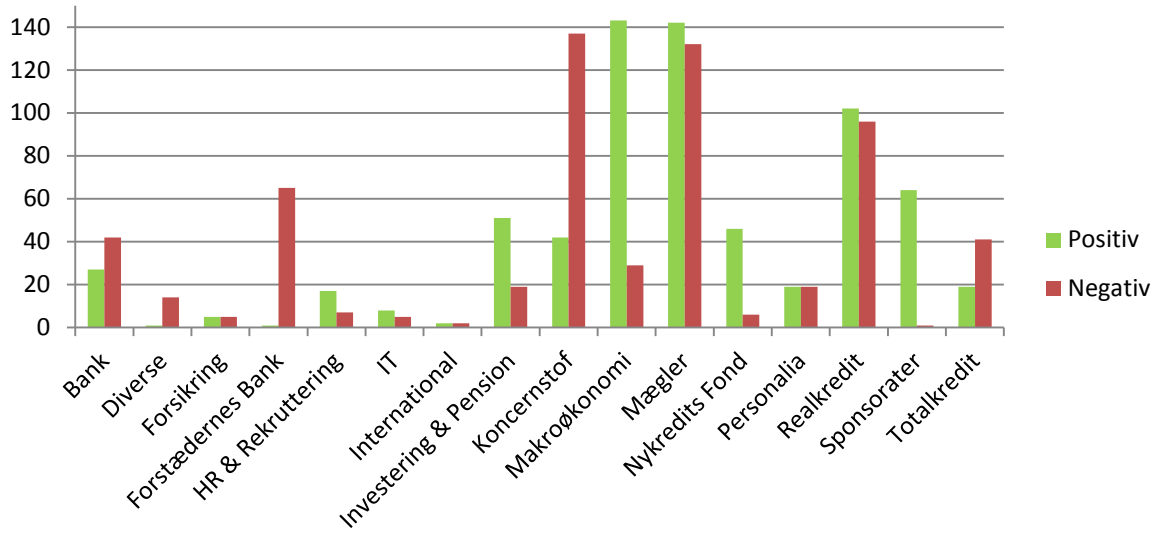
De to artikler er som skrevet kun *næsten* identiske, så hvad der i realiteten er to semantisk ækvivalente artikler bliver ikke fanget af det script, der i preprocessing-fasen fjerner dubletter i korpusset. Jeg har derfor gjort brug af Python-klassen difflib, som indeholder en metode, der kan beregne, hvor similære to tekster er. Dette sker på baggrund af, hvor mange tegn, de to tekster har tilfælles, og hvordan de er placeret i forhold til hinanden. To tekster, der ingen tegn har tilfælles har en similaritet på 0.0, mens to identiske tekster har en similaritet på 1.0. Jeg har sat en tærskel på 0.75, og hvis to teksters similaritet bliver beregnet til mere end dette, regnes de for semidubletter, hvorefter den ene tekst bliver fjernet fra korpusset.

212 artikler optræder i semidubletform, og de figurerer hver mellem 2 og 16 gange i korpusset. Jeg fjerner 453 semidubletter, og dette bringer endeligt antallet af artikler ned på 1309, som er fordelt på 689 positive og 620 negative:

Tabel 6.2

	Samlet	**Positiv**	**Negativ**
Bank	69	27	42
Diverse	15	1	14
Forsikring	10	5	5
Forstædernes Bank	66	1	65
HR & Rekruttering	24	17	7
IT	13	8	5
International	4	2	2
Investering & Pension	70	51	19
Koncernstof	179	42	137
Makroøkonomi	172	143	29
Mægler	274	142	132
Nykredits Fond	52	46	6
Personalia	38	19	19
Realkredit	198	102	96
Sponsorater	65	64	1
Totalkredit	60	19	41
I alt	**1309**	**689**	**620**

[12] Artikel fra bilag A med id-koden e25bd9b3



Figur 6.2

6.3 Kategorisering af artikler

Som udgangspunkt var det meningen, at det program, der skal fremstilles, automatisk skulle kunne tilskrive nye, ukendte artikler én af ovenstående kategorier. Distributionen af træningssættets forskellige typer artikler er dog for skævvredet til, at der umiddelbart er mulighed for dette. En så ujævn klassedistribution giver nemlig hurtigt problemer. Hvis en classifier eksempelvis bliver trænet på et binært fordelt datasæt, hvor 10% af instanserne udgøres af klasse A, mens 90% af instanserne udgøres af klasse B, vil classifieren gennemsnitligt kunne opnå en accuracy på 90% ved blot at tilskrive ethvert datapunkt klassen B (Forman, s. 261).

6.4 Kvaliteten af data

Kvaliteten af data afgøres også af mange ting, men som minimum må man forlange, at de enkelte instanser er klassificeret korrekt – i hvert fald i den udstrækning dette nu kan lade sig gøre.

Det er åbenlyst, at det ikke er en triviel opgave at konstruere en maskine, som kan afgøre, om en tekst forholder sig positivt eller negativt til et givent emne. Men selv for mennesker kan det være svært, om ikke umuligt at blive enige om, hvorvidt en tekst er positiv eller negativ.

6.4.1 Intercoder agreement

Ønsker man som i denne undersøgelse at gøre brug af manuelt opmærket data, er man nødt til at vise, at disse data er pålidelige. I løse vendinger kan data siges at være pålidelig, hvis det kan vises, at annotørerne (*the coders*) som står for opmærkningen, i en grad, der afhænger af undersøgelsens rammer og formål, er enige om, hvilke kategorier – i denne undersøgelses tilfælde positiv og negativ – datasættets instanser skal tilskrives. Er dette tilfældet, kan man udlede, at annotørerne har tilegnet sig en similær forståelse for de retningslinjer og det annoteringssystem (*coding scheme*), der er udgangspunktet for opmærkningen af datasættet. Dette kaldes *intercoder agreement* og kan bestemmes gennem forskellige typer beregninger (Artstein & Poesio, s. 557).

At annotørerne er enige om, hvordan datasættet skal opmærkes, er også udtryk for, at annoteringssytemet så at sige kongruerer med indholdet af instanserne i datasættet – at det giver mening at klassificere dem i forhold til systemet – og dette er af yderste og helt fundamentale vigtighed for hele undersøgelsen. For hvis annotørerne ikke (i tilstrækkelig høj grad) er enige om, hvordan datasættet skal opmærkes, må det skyldes, at annotørerne enten tager fejl i deres vurderinger eller at annoteringssystemet, som de opmærker på baggrund af, simpelthen ikke er velegnet i forhold til data. Datas pålidelighed er altså en nødvendig forudsætning for undersøgelsens validitet, for kan data ikke regnes for pålidelige, må alle resultater og konklusioner i undersøgelsen i bedste fald betragtes som værende tvivlsomme og i værste fald meningsløse (Lombard, s.3).

6.4.1.1 Beregning af intercoder agreement

Det simpleste mål for enigheden i et korps af annotører er den procentuelle enighed, også kaldet den observerede enighed. Fx kan to annotører tilskrive et datasæts 150 instanser én af de to kategorier A og B, hvilket de gør på følgende måde:

Tabel 6.3

		Annotør 1		
		A	B	I alt
Annotør 2	A	**35**	20	55
	B	15	**80**	95
	I alt	50	100	**150**

Tabellen – en såkaldt *contingency table* - viser, at annotør 1 har klassificeret 50 instanser som A, hvoraf annotør 2 er enige i 35 af dem. Ligeledes har annotør 2 klassificeret 95 instanser som B, hvoraf annotør 1 er enige i 80 af dem. Således kan man i tabellens diagonal aflæse, at de to annotører er enige i klassifikationen af 115 af de 150 instanser, hvilket giver en observeret enighed på 76.67%. Det er klart, at den observerede enighed ikke umiddelbart giver mening, hvis der er mere end to annotører (hvilket som regel er et absolut minimum og et antal, som kun benyttes til ganske små studier (Artstein & Poesio, s. 562)). Jeg beskriver disse tilfælde i 6.4.1.3.

Der er imidlertid visse problemer med at anvende den observerede enighed som metrik. For det første indeholder den en væsentlig bias, som i højere grad sandsynliggør enighed inden for kategorier af lavere dimensioner. I klassifikationsopgaven illustreret med Tabel 6.3 er det et binært valg mellem de to klasser A og B. Ved blot at gætte er der således ½*½=¼ chance for, at begge annotører tilskriver en instans den samme klasse:

Tabel 6.4

		Annotør 1		
		A	B	I alt
Annotør 2	A	**¼**	¼	½
	B	¼	**¼**	½
	I alt	½	½	**1**

Ved igen at aflæse tabellens diagonal ses det, at det er sandsynligt, at de to annotører ved helt tilfældigt at vælge en klasse er enige om klassifikationen af halvdelen af datasættets instanser. Hvis antallet af mulige klasser derimod var 3 i stedet for to, så situationen således ud:

Tabel 6.5

			Annotør 1		
		A	B	C	I alt
	A	**1/9**	1/9	1/9	1/3
Annotør 2	B	1/9	**1/9**	1/9	1/3
	C	1/9	1/9	**1/9**	1/3
	I alt	1/3	1/3	1/3	**1**

I dette tilfælde ville annotørerne ved blot at gætte sandsynligvis opnå enighed i kun 1/9 + 1/9 + 1/9 = 1/3 af klassifikationen af datasættets instanser.

Den anden grund til, at den observerede enighed er et problematisk mål, er at den ikke korrigerer for fordelingen af instanser på de kategorier, der skal opmærkes i henhold til. Hvis én kategori optræder hyppigere end en anden kategori, vil en annotør nemlig have en tendens til oftere at tilskrive en instans den hyppigst optrædende kategori (Artstein & Poesio, s. 559).

6.4.1.2 Korrigering for tilfældig enighed

Den observerede enighed udgør altså et problem, fordi den ikke tager højde for de sandsynligheder, der ligger til grund for, at to eller flere annotører rent tilfældigt kan være enige i klassifikationen af et datasæts instanser. For at komme dette problem til livs kan man gøre brug af andre teknikker, som korrigerer for denne tilfældige enighed.

For det første skal det bestemmes hvor stor enighed, man rent tilfældigt kan forvente – denne værdi kaldes A_e, mens A_O er den observerede enighed. Således bestemmer $1 - A_e$ hvor stor enighed, der maksimalt kan opnås i forhold til den tilfældige enighed. $A_o - A_e$ bestemmer derfor hvor stor enighed, der – ud over hvad der kan forventes af den tilfældige enighed – er opnået. Endelig bestemmer værdien $(A_o - A_e)/(1 - A_e)$ størrelsen af den enighed, der er opnået, i forhold til hvad der er muligt ud over den tilfældige enighed. Denne værdi kaldes S, π eller κ:[13]

$$S, \pi, \kappa = \frac{A_O - A_e}{1 - A_e}$$

De tre koefficienter S, π og κ ligger alle mellem $A_e/(1 - A_e)$ og 1, hvor 1 er fuldstændig enighed, 0 er svarende til tilfældig enighed, og $A_e/(1 - A_e)$ er fuldstændig uenighed (dvs. $A_o = 0$). Ligeledes gælder det for alle tre koefficienter, at de alle antager, at annotørerne, der står for opmærkningen, fungerer uafhængigt af hinanden. Det betyder, at sandsynligheden for, at annotørerne c_1 og c_1 er enige om, at en instans tilhører en given kategori k, er lig produktet af sandsynligheden for, at de hver især tilskriver instansen kategori k: $\mathrm{P}(k|c_1) \cdot \mathrm{P}(k|c_2)$. Den forventede enighed er så sandsynligheden for, at c_1 og c_2 er enige om hver kategori, dvs. summen af produktet over alle kategorier:

[13] Der findes flere varianter af κ – jeg benytter Cohens variant (Artstein & Poesio, s. 559)

$$A_e^S = A_e^\pi = A_e^\kappa = \sum_{k \in K} \mathrm{P}(k|c_1) \cdot \mathrm{P}(k|c_2)$$

Forskellen mellem de tre koefficienter ligger i de antagelser, der ligger til grund for, hvordan man beregner $\mathrm{P}(k|c_i)$ - sandsynligheden for, at annotør c_i vil tilskrive en tilfældig instans klassen k:

- S er baseret på antagelsen om, at hvis et hold annotører opmærker data rent tilfældigt, vil det give en uniform distribution. Dvs. for ethvert par af annotører c_m, c_n og for to vilkårlige klasser k_j, k_l er $\mathrm{P}(k_j|c_m) = \mathrm{P}(k_l|c_n)$.
- π er baseret på antagelsen om, at hvis et hold annotører opmærker data rent tilfældigt, vil det for hver annotør give en identisk distribution. Dvs. for ethvert par af annotører c_m, c_n og for en vilkårlig klasse k er $\mathrm{P}(k|c_m) = \mathrm{P}(k|c_n)$.
- κ er baseret på antagelsen om, at hvis et hold annotører opmærker data rent tilfældigt, vil det give en separat distribution for hver annotør. Antagelsen tager udgangspunkt i, at hvis instanserne i datasættet tilfældigt bliver tilskrevet en klasse, styres denne tilskrivning af en given fordeling, som er unik for hver annotør. Denne givne fordeling bliver for hver annotør estimeret ved at se på, hvorledes annotøren rent faktisk har bedømt instanserne i datasættet. $\mathrm{P}(k|c_i)$, som er sandsynligheden for at annotør c_i vil tilskrive en tilfældig instans klassen *k*, bliver således estimeret ved at benytte $\widehat{\mathrm{P}}(k|c_i)$; andelen af instanser, som annotør c_i klassificerede som *k*.

Således ser koefficienterne for tallene i Tabel 6.4 sådan ud:

Tabel 6.6

Koefficient	Forventet enighed A_e	Enighed korrigeret for tilfældig enighed
S	$2 \cdot \left(\frac{1}{2}\right)^2 = \frac{1}{2}$	$\frac{\left(\frac{115}{150}\right) - \frac{1}{2}}{1 - \frac{1}{2}} = \mathbf{0.533}$
π	$0.35^2 + 0.65^2 = 0.545$	$\frac{\left(\frac{115}{150}\right) - 0.545}{1 - 0.545} = \mathbf{0.487}$
κ	$\left(\frac{50}{150}\right) \cdot \left(\frac{55}{150}\right) + \left(\frac{100}{150}\right) \cdot \left(\frac{95}{150}\right) = 0.544$	$\frac{\left(\frac{115}{150}\right) - 0.544}{1 - 0.544} = \mathbf{0.488}$

6.4.1.3 Flere end to annotører

Når der er mere end to annotører, giver det ikke mening at tale om observeret enighed, da der (praktisk talt) altid vil være en mængde instanser, som annotørkorpset ikke vil være entydigt enige om. Det er også sværere at visualisere data, da hver annotør forårsager en ekstra dimension i contingency-tabellen, så det er heller ikke muligt at gøre brug af tabeller som Tabel 6.3. De tre koefficienter S, π og κ kan således kun benyttes til bestemmelse af enighed, når der er to annotører.

I stedet for observeret enighed, kan man beregne enigheden for hvert par af annotører og herefter lave et gennemsnit af dette. Rent praktisk er det dog bedre i stedet at gøre brug af såkaldt generaliserede versioner af de nævnte koefficienter.

6.4.1.3.1 Fleiss' Multi-π

Fleiss' Multi-π er en metode til beregning af observeret enighed, når der er mere end to annotører. I stedet for contingency-tabeller laves der en såkaldt *agreement table*, som illustrerer hvor mange af de tre eller flere annotører, der tilskriver de enkelte instanser de forskellige klasser. Se eksempel:

Tabel 6.7 Agreement table

	Positiv	Negativ
Artikel 1	2	1
Artikel 2	0	3
⋮		
Artikel N	1	2
I alt	**90 (0.3)**	**210 (0.7)**

Række 1 angiver, at Artikel 1 er blevet bedømt positiv af 2 annotører, mens en enkelt har bedømt den negativ. I forhold til contingency-tabellen går der noget information tabt i agreement-tabellen, for det bliver ikke oplyst, hvordan den enkelte annotør bedømmer de enkelte instanser. Denne information benyttes jo til at beregne κ-koefficienten, men ikke π-koefficienten.

6.4.1.3.1.1 Observeret enighed A_o^π

Lad $n_{i,k}$ være antallet af annotører, der tilskriver instans *i* klasse *k*. Eksempelvis er $n_{\text{Art1,Pos}} = 2$ i Tabel 6.7. For hver kategori *k* er der enighed mellem $\binom{n_{i,k}}{2}$ par af annotører i deres bedømmelse af instans *i*.

I bedømmelsen af instans *i* er der for kategori *k* enighed mellem $\binom{n_{i,k}}{2}$ par af annotørerne. Har for eksempel fire annotører tilskrevet instans *Ins_1* kategorien *Kat_1*, er der altså enighed mellem $\binom{n_{Ins_1,Kat\_1}}{2} = \binom{4}{2} = 6$ annotørerpar. Enigheden blandt annotørerne for bedømmelsen af instans *i* (benævnt agr_i) er summen af $\binom{n_{i,k}}{2}$ over alle kategorierne *k* (i eksemplet i Tabel 6.7 altså Positiv og Negativ) divideret med $\binom{c}{2}$, hvor *c* angiver antallet af annotører. Da $\binom{c}{2}$ jo angiver antallet af måder at udvælge to – altså et par – annotører, er $\binom{c}{2}$ altså det maksimale antal af par, der kan være enighed mellem.

$$\text{agr}_i = \frac{1}{\binom{c}{2}} \sum_{k \in K} \binom{n_{i,k}}{2} = \frac{1}{c(c-1)} \sum_{k \in K} n_{i,k}(n_{i,k} - 1)$$

For resultaterne i Tabel 6.7 beregnes enigheden mellem annotørerne for bedømmelsen af Artikel 1 således:

$$\text{agr}_{\text{Art\_1}} = \frac{1}{\binom{3}{2}} \left(\binom{n_{\text{Art\_1,Pos}}}{2} + \binom{n_{\text{Art\_1,Neg}}}{2} \right) = \frac{1}{3} \left(\binom{2}{2} + \binom{1}{2} \right) = \frac{1}{3}(1 + 0) = \frac{1}{3} \approx 0.33$$

Den samlede observerede enighed A_o^π beregnes nu som et gennemsnit af agr_i for alle instanser *i* i datasættet ($|i|$ angiver antallet af instanser):

$$A_o^{\pi} = \frac{1}{|i|}\sum_{i \in I} \text{agr}_i = \frac{1}{|i| \cdot c(c-1)}\sum_{i \in I}\sum_{k \in K} n_{i,k}(n_{i,k}-1)$$

6.4.1.3.1.2 *Forventet enighed A_e^{π}*

Eftersom den observerede enighed bliver beregnet på baggrund af enighed på parvist niveau blandt annotørerne, giver det mening også at anskue den forventede enighed som sandsynligheden for, at to vilkårlige annotører tilfældigvis er enige i deres bedømmelse af en given instans. Som i beregningen af π (for to annotører) fortolkes den tilfældige enighed som den enighed, der kan forventes på baggrund af fordelingen, der afspejler hvorledes det samlede annotørkorps har bedømt hele datasættet. Således benytter man i beregningen af den forventede enighed tallet $\widehat{\mathrm{P}}(k)$, som angiver hvor stor en del af annotørernes bedømmelser, der er lig *k*. $\widehat{\mathrm{P}}(k)$ angiver sandsynligheden for, at en annotør helt tilfældigt vil tilskrive en vilkårlig instans klassen *k*, og det er lig antallet af gange, annotørerne har tilskrevet en instans klassen *k* (benævnes n_k), divideret med det samlede antal bedømmelser (dvs. antallet af annotører ganget med antallet af instanser):

$$\widehat{\mathrm{P}}(k) = \frac{1}{ic} n_k$$

Eftersom sandsynligheden for, at en enkelt annotør tilfældigt tilskriver en instans klassen *k* er lig $\widehat{\mathrm{P}}(k)$, er sandsynligheden for, at to annotører gør det – og derved er enige - $\left(\hat{P}(k)\right)^2$. Således er den forventede enighed lig summen af $\left(\hat{P}(k)\right)^2$ for samtlige kategorier $k \in K$:

$$\mathrm{A}_{\mathrm{e}}^{\pi} = \sum_{k \in K}\left(\hat{P}(k)\right)^2 = \sum_{k \in K}\left(\frac{1}{ic} n_k\right)^2 = \frac{1}{ic^2}\sum_{k \in K} n_k^2$$

6.4.2 Annotering af Nykredit-korpusset

Det er som sagt Infomedia, der har leveret data til dette projekt, og det er også hos dem, den manuelle opmærkning af korpusset har fundet sted. Hos Infomedia modtager de folk, som forestår klassifikationen, et mindre klassifikationskursus, inden de bliver kastet ud i opgaven. Jeg har ikke nærmere kendskab til indholdet af dette kursus. Hver tekst bliver kun klassificeret af en enkelt medarbejder. Herefter bliver der løbende foretaget stikprøver som en form for kvalitetskontrol. Det formodes, at dette giver fint brugbare resultater i forhold til den kommercielle anvendelse af dette arbejde, men det er klart, at opmærkningen i forhold til validitet og videnskabelig anvendelighed er mere problematisk. Mere om dette senere.

Det er også klart, at der blandt korpussets 1309 artikler vil være eksempler på tekster, der kan tolkes begge veje. Et klassisk eksempel på en svært klassificerbar tekst, er en tekst, hvor forfatteren har gjort brug af ironi. Artiklerne fra Nykredit-korpusset er alle fra nyhedsmedier – en type medier, der begår sig i en genre, som sjældent gør brug af ironi, men der er stadig en del artikler, som ikke er entydigt positive eller negative.

6.4.3 Clear Cases

Der er dog også tekster, hvor der ikke burde være nogen tvivl om, hvilken klasse de tilhører – de såkaldte *clear cases*.

6.4.3.1 Good case

Følgende artikel (som muligvis mere blot er en sætning end en decideret artikel) er klassificeret positivt, hvilket vist ikke kan gøres anderledes:

> *Nykredit har markant øget forventningerne til resultatet før skat for 2010, og venter nu at kunne tjene op til 3,2 milliarder kr.*

Nykredit øger forventningerne og tjener masse af penge – umulig at tolke negativt.

6.4.3.2 Bad Cases

Der findes også en mængde *bad cases* – klassificeringer, som åbenlyst er fejlagtige.

6.4.3.2.1 Fejlagtigt negativt klassificeret artikel

Som det fremgår af Tabel 6.2, er en enkelt artikel i kategorien ”Sponsorater” klassificeret negativ. Dette kan umiddelbart undre, da et sponsorat normalt er associeret med noget positivt. Og læser man artiklen igennem, er det klart, at den er fejlklassificeret. Teksten omhandler i korte træk glade børn, som pynter et juletræ og spiser guf:

> *Og de syntes bestemt ikke, det var kedeligt. Dagen blev nemlig indledt med æbleskiver og saftevand - bagefter stod den på julepynt - som de selv havde lavet - det skulle på det store juletræ i Nykredits lokaler. De mange børn fra Stjernehusets Solgruppe samt Spire-og Radisestuerne i Tirsdalens Børnehave, gik til opgaven med stor omhu - de kunne dog ikke nå de øverste grene og måtte indimellem have hjælp af de voksne. Det gjorde heller ikke så meget - de havde nemlig selv lavet pynten, og den blev fordelt så godt de kunne.Til sidst blev træet tændt, mens de små fik en velfortjent belønning - en slikpose hver.*[14]

En negativ læsning er svær at forestille sig, men det er ikke desto mindre tilfældet. Ser man på de enkelte ord, er der både plus- og minusord – fx ”kedeligt”, ”velfortjent” og ”belønning” – og var klassificeringen sket på en statistisk baggrund, kunne der måske plæderes for både en negativ og positiv klassificering, men dette er ikke tilfældet. Infomedias medarbejdere har jo manuelt læst hver enkelt tekst igennem og truffet et valg herudfra.

6.4.3.2.2 Fejlagtigt positivt klassificerede artikler

Det er svært at finde eksempler på artikler, som er negative i forhold til Nykredit. En tekst som denne, synes umiddelbart negativ:

> *Det bliver sværere at være husejer i Danmark. Nye internationale krav til finanssektorens likviditet betyder, spår Nykredit, at boliglån ikke bare bliver dyrere men også sværere at få bevilget.*[15]

[14] Artikel fra bilag A med id-koden e255efd5
[15] Artikel fra bilag A med id-koden e25942c4

Den er klassificeret positivt, men for Nykredit er ovenstående som sådan ikke negativt, og der er heller intet i teksten, som markerer, at det enten er positivt eller negativt for Nykredit. Det samme gør sig gældende for den følgende tekst, som det er umuligt at finde noget positivt ved – heller ikke for Nykredit og Nybolig. Ikke desto mindre er den klassificeret positiv:

> *Pushere, narkomaner og prostituerede gør lejligheder på Vesterbro nær Hovedbanegården usælgelige. Støj fra slagsmål og den åbenlyse kvindehandel i gaderne gør boligkøberne utrygge. »Det er jo ikke der, du sender dine børn ud at lege på fortorvet,« siger Robert Børnum fra Estate Mæglerne.*[16]

6.5 Validering af datasættet

Da det er mit indtryk, at kvaliteten af opmærkningen af korpusset halter visse steder, har jeg besluttet at foretage en opmærkning på egen hånd, og holde denne opmærkning op mod Infomedias. Det vil forhåbentlig give et indtryk af hvilke type tekster, der – måske - giver anledning til fejlklassificeringer, og det vil ligeledes give et indtryk af hvor stor en fejlprocent, der er i opmærkningen.

Det virker uoverskueligt at engagere eksterne folk til at opmærke samtlige 1309 tekster i korpusset, så jeg i stedet har valgt at tage 10% ≈ 130 af teksterne, og sætte to annotører til at opmærke disse for herefter at holde disse resultater op mod Infomedias opmærkning. Der er 65 tekster af hver klasse, og de er alle mellem 600 og 900 tegn lange. Forud for opmærkningen er annotørerne blevet givet følgende instruktion:

> *Korpusset består af 130 tekster, som alle er mellem 600 og 900 tegn lange. Teksterne omhandler Nykredit i den forstand, at ordet 'Nykredit' indgår i hver tekst. Når du har læst en tekst, skal du give den bedømmelsen 0, 1 eller -1.*
>
> ***Bedømmelsen -1** gives, hvis indholdet i teksten forholder sig negativt eller kritisk til Nykredit eller elementer relateret hertil. Dette kan komme til udtryk på forskellig vis.For eksempel kan forfatteren af artiklen udtrykke sig kritisk i forhold til Nykredit, men det kan også være personer beskrevet i artiklen, som kritiserer Nykredit.*
> ***Bedømmelsen 1** gives, hvis indholdet i teksten forholder sig positivt til Nykredit eller elementer relateret hertil. Denne positivitet kan komme til udtryk på samme måde som i det negative tilfælde.*
> ***Bedømmelsen 0** gives, hvis det ikke synes muligt at bedømme en tekst efter ovenstående kriterier.*
>
> *Det understreges at bedømmelsen sker på et sprogligt og holdningsmæssigt grundlag - ikke et topikalt grundlag. Det er således tekstens forhold til Nykredit, der skal bedømmes efter og IKKE emnet for teksten. En tekst, der omhandler et emne, der umiddelbart kan have en negativ konnotation - fx finanskrisen – skal IKKE bedømmes negativt, hvis Nykredit eller elementer relateret hertil ikke også bliver omtalt i kritiske vendinger.*

Som det fremgår af ovenstående, har annotørerne mulighed for at tilskrive teksterne klasserne 0, dvs. neutrale tekster, til trods for, at jeg selv i preprocessing-fasen frasorterede de relativt få tekster, som tilhørte denne klasse. Årsagen til dette er, at de eller den annotør hos Infomedia, der oprindeligt

[16] Artikel fra bilag A med id-koden e23c7189

opmærkede teksterne, også havde muligheden at tilskrive teksterne en neutral klasse. Hvad der ligger til grund for, at de i så lav en grad har gjort brug af denne mulighed, ved jeg ikke.

Det var i øvrigt ikke muligt for mig at fremskaffe de instruktioner, som Infomedias annotører gør brug, hvorfor jeg var nødt til at fremstille mine egne.

6.5.1 Analyse af forsøget

Med de to eksterne annotørers (A) og (B) og Infomedias (IM) egen opmærkning har vi altså et annotørkorps på tre personer. Det skal siges, at det sandsynligvis nok ikke er den samme annotør hos Infomedia, der har opmærket hverken det fulde korpus eller de 130 artikler, som er blevet udvalgt til dette forsøg på at teste korpussets validitet. Men man må formode, at Infomedia-annotørerne har de samme forudsætninger for opmærkningen, da de alle er blevet givet den samme instruktion om dette.

Mellem de tre annotører er der en observeret enighed (A_O^{π}) på blot 31.5%. Dette skyldes især den eksterne annotør (B), som i 117 ud af 130 tilfælde har tilskrevet en artikel den neutrale klasse, og den observerede enighed (A_o) mellem (B) og (IM) er 6.92%. Mellem (A) og (IM) er den observerede enighed 46.92%, og mellem (A) og (B) er der en observeret enighed på 40.77%.

Den instruktion og de retningslinjer, jeg i 6.5 stak ud til mine annotører, har jeg selv forfattet uafhængigt af den instruktion/det kursus, Infomedias medarbejdere har gennemgået. Jeg har naturligvis skrevet den, så jeg synes, at den giver mening i forhold til det, datasættet skal bruges til, men der er jo ingen garanti for, at denne er i overensstemmelse med Infomedias retningslinjer. Dette må antageligvis være årsagen til den lave observerede enighed mellem de eksterne annotører og Infomedias.

6.5.1.1 Årsag til den lave observerede enighed

Men da (A) og (B) er blevet givet den samme instruktion forud for opmærkningen af korpusset, ville det være rimeligt at forvente en højere observeret enighed disse eksterne annotører imellem. Årsagen til dette skyldes tilsyneladende, at de har læst – eller i hvert fald forstået – instruktionen forskelligt. Se fx:

> *Så skete det, som alle godt vidste i forvejen. Hans Henrik Palm blev onsdag morgen erklæret konkurs i Sø- og Handelsretten.Konkursen sker efter anmodning fra skattevæsenet, som Hans Henrik Palm skylder flere millioner kroner.Hans Henrik Palm har ud over gæld til skattevæsenet også rigtigt mange penge i klemme i Amagerbanken, der indtil nu har holdt hånden under ham.Men bankens nye ledelse nåede altså ikke at få set på Palms gæld til banken, før skattefar mistede tålmodigheden.Palm har blandt andet investeret kraftigt i boliger på Østerbro, investeringer som Amagerbanken indtil nu har valgt ikke at ville nedskrive prisen på.Ifølge Palms seneste regnskab, har han en gæld på 450 mio. kr., hvoraf de 377 mio. kr. er gæld til henholdsvis Amagerbanken og Nykredit Bank.Så måtte den gamle bokser til sidst kaste håndklædet i ringen. Kl. 09.40 blev han erklæret konkurs..*[17]

I overensstemmelse med den instruktion, jeg skrev, har (B) forholdt sig til sproget og hvorledes Nykredit bliver omtalt i artiklen, og (B) tilskriver således artiklen den neutrale klasse. (A) har ligesom (IM) tilskrevet teksten den negative klasse, hvilket kan give anledning til at tro, at Infomedias annotører er blevet

[17] Artikel fra bilag A med id-koden e24c0e8a

instrueret i at klassificere i forhold til emnet i teksten (i dette tilfælde altså en konkurs, som alt andet lige må betegnes som et negativt fænomen) og ikke i forhold til sproget og hvorledes Nykredit bliver omtalt.

Blandt de 130 artikler, der er blevet udvalgt til dette forsøg, er der faktisk ingen eksempler på, at Nykredit bliver omtalt i negative vendinger. Således kan det ikke være på baggrund af sproglige forhold, at Infomedias annotører har klassificeret 65 af artiklerne som værende negative. Langt størstedelen af artiklerne forholder sig for så vidt helt neutralt til Nykredit, og det må således være på baggrund af emnet for artiklen, at Infomedias opmærker artiklerne. Der er til gengæld enkelte eksempler på, at artiklen i positive vendinger omtaler Nykredit:

> *Nykredit har fået tildelt prisen "Best Danish Fund Selection Team of the Year".Det skriver Nykredit i en meddelelse.Det er Tell Media Group, som uddeler prisen på baggrund af en afstemning blandt de største internationale kapitalforvaltere, der opererer i det nordiske marked.- Vi er naturligvis glade og stolte af at modtage en sådan anerkendelse, og det er et udtryk for at vore kunder sætter pris på vores arbejde og indsats, skriver Claus Bilde, afdelingsdirektør i Manager Selection i Asset Management.Ud over Nykredit var de nominerede til prisen i år PFA Pension og Kirstein Finansrådgivning.RB-Børsen*[18]

Denne type artikler bliver klassificeret positivt af alle annotører (eksterne og Infomedias).

Det er dog mit klare indtryk, at rigtig mange artikler bliver klassificeret forkert af Infomedias annotører. Eksempelvis bliver følgende artikel klassificeret positivt:

> *Der var 596 konkurser i september efter sæsonkorrektion, og det er på niveau med den forudgående måned. Det er det højeste niveau i adskillige år. Den seneste opgørelse fra Danmarks Statistik viser samtidig, at de fleste konkurser sker i hovedstadsområdet. Nykredit vurderer, at konkurserne ikke kun skal tilskrives finanskrisen, men også det faktum at mange virksomheder blev stiftet i samfundsøkonomisk medvind. Chefanalytiker i Nykredit Market Jakob Legård Jakobsen vurderer, at mange af forretningsstrategierne derfor ikke har været helt gennemtænkte, når det kommer til overlevelse på længere sigt. Link til udsendelseIndslaget forekommer ca. 2 min. inde i udsendelsen.*[19]

Hverken sprog eller omtale af Nykredit kan betegnes positiv (sidstnævnte dog heller ikke negativ). Emnet er konkurser – tilmed en lille rekord – og det burde være klart, at dette emne må listes under klassen 'Negativ'. Omvendt er følgende artikel klassificeret negativt:

> *HØRSHOLM: I dag er der på ny fernisering for endnu en kunstudstilling i Nykredits lokaler i Hovedgaden 55 i Hørsholm.Banken har i gennem snart fem år åbnet dørene for kunsten, lokale kunstnere, og ikke mindst lokale kunstinteresserede borgere. Denne gang er det kunstneren Marianne Hagemann fra Fredensborg der udstiller.Marianne Hagemann, der har malet i en lang årrække, betegner sig selv som autodidakt, men har gennem tiden suppleret og forstærket sine medfødte talenter med en bred vifte af undervisning fra billedkunstnere og tegnere.Marianne*

[18] Artikel fra bilag A med id-koden e23cab19

[19] Artikel fra bilag A med id-koden e238b1d6

> *Hagemann har gennem tiden udstillet flere gange i Hørsholm og Fredensborg. I 2007 blev Marianne kåret som »Årets Billedkunstner« i Fredensborg Kommune.Ferniseringen, hvor Nykredit byder på et glas og lidt mundgodt, er åben for alle interesserede. Det er fra klokken 17 til 19.*[20]

Denne gang kan hverken sprog eller emne betegnes negativt. Emnet er kunst og en forhåbentlig festlig fernisering hos den lokale Nykredit, og denne skal naturligvis klassificeres positivt – Infomedias annotør har tilskrevet den klassen 'Negativ'.

Der er rigtig mange eksempler på denne type klassificeringer fra Infomedias annotørkorps, og det er medvirkende til, at det i mange tilfælde virker ganske tilfældigt, hvordan der klassificeres. Derudover har der tilmed sneget sig en tysksproget tekst ind blandt de 130 tekster, der blev valgt ud til granskning.

6.5.2 Datas velegnethed

Som ovenstående formentlig er med til at vise, er det tvivlsomt, om denne type data overhovedet er egnet til opinion mining. Jeg mener i hvert fald ikke, at jeg har haft held med at vise, at datasættet er pålideligt.

6.6 Opinion mining på journalistiske artikler

I forhold til sentiment analysis er nyhedsartikler meget anderledes at have med at gøre end eksempelvis filmanmeldelser, som er en typisk genre, når der laves forsøg med opinion mining. Sidstnævnte indeholder som regel mange subjektive sætninger og udtryk, mens nyhedsgenren gerne har en mere objektiv og beskrivende karakter. Se følgende eksempel:

> *Kerteminde: Det bliver Nybolig Erhverv i Odense, der skal sælge de 374 grunde i sommerbyen for Kerteminde Kommune.- Vi henvendte os til fire erhvervsmæglere - det ene firma - ét fra København svarede ikke - og vi har valgt det billigste af de tre andre, siger borgmester Sonja Rasmussen, der tilføjer, at alle mæglere blev bedt om at redegøre for deres netværk.*

Teksten – som i øvrigt er klassificeret negativt – indeholder blot et enkelt adjektiv, nemlig *billigste* – et adjektiv, der ikke siger meget om tekstens polaritet.[21] Manglen på adjektiver, der kan fungere som subjektive markører, gør sig ligeledes gældende i denne artikel, som er klassificeret negativt:

> *INDBRUD: Ejendomsmæglerne hos Nybolig i Vojens har en anelse mere albuerum i dagen.I løbet af natten er ukendte gerningsmænd nemlig brudt ind på kontoret i Rådhuscentret og er stukket af med flere designerstole.Indbruddet er sket omkring klokken 04.20, og tyven slap af sted med 12 sorte Hans Wegner Y-stole med hynder, et Piet Hein superellipse bord i jubilæumsudgave, samt tre trebenede Arne Jacobsen stole.Tyven kom ind gennem en opbrudt dør.*[22]

Ukendte, sorte, trebenede og *opbrudt* er de eneste adjektiver – ord, der ikke umiddelbart udtrykker nogen form for subjektivitet.

[20] Artikel fra bilag A med id-koden e243db71

[21] Artikel fra bilag A med id-koden e23de71c

[22] Artikel fra bilag A med id-koden e257c420

Dette udgør et problem, hvis man vil træne en classifier på tekster, som er udpræget subjektive (fx brugergenerede anmeldelser), for derefter at evaluere classifieren på nyhedsartikler. Der er en primær årsag til dette.

Det kan nemlig diskuteres, om en nyhedsartikel overhovedet udtrykker subjektivitet – måske er den blot beskrivende, og det der betragtes som en negativ artikel, handler i virkeligheden blot om et *emne*, som gerne *associeres* med noget negativt – finanskrisen eksempelvis. Dette så vi tidligere eksempler på. Og hvor man i en subjektiv anmeldelse – og især i den type anonymt brugergenerede anmeldelser, man finder i stor stil på nettet - kan møde vendinger á la ”filmen stinker” eller måske endda ligefrem ”sutter røv”, så gør man det bare ikke i en journalistisk skrevet nyhedsartikel.

Således er problemet, at de markører, som normalt indikerer subjektive udtryk, ikke kommer i spil inden for nyhedsgenren. Men hvad skal vi så klassificere på baggrund af, hvis vi ikke kan gøre brug af det arsenal, der normalt køres i stilling i forbindelse med opinion mining? Måske man i virkeligheden kommer bedre afsted med at udføre opinion mining på baggrund af de markører, man normalt ville have brugt til at klassificere tekster efter deres emner.

Efter denne beskrivelse af mine data vil jeg nu gøre rede for det lille program, jeg har skrevet.

7 Implementering af program

Som skrevet har jeg arbejdet med en implementering af et program, som kan tilskrive en tekst en af klasserne *positiv* eller *negativ*. Arbejdet er lavet i Python med diverse frameworks og forskellige moduler.

7.1 Python

Python er et programmeringssprog, der er meget udbredt inden for maskinel behandling af natursprog. Det er et sprog, der i forhold til mange andre programmeringssprog har en syntaks, semantik og læsbarhed, der gør, at det er yderst let tilgængeligt. Dette skinner ikke mindst igennem via de 19 aforismer, som beskriver tankegangen bag sproget, og som med kommandoen **import this** kan kaldes frem – tre af disse er netop *Simple is better than complex, Complex is better than complicated* og *Readability counts*.[23]

Til Python findes et væld af moduler og eksterne frameworks, som er med til at lette arbejdet. Et af disse frameworks er NLTK – Natural Language Tool Kit.

7.2 Natural Language Tool Kit

NLTK er et open source-værktøj, og det er et af de mest udbredte Python-frameworks inden for arbejdet med natursprog. Det indeholder en mængde værktøjer, korpusser og klasser, der kan bruges til at repræsentere data benyttet i natursprogsbehandling. Gennem NLTK er det desuden muligt at benytte en begrænset mængde maskinlæringsalgoritmer.

7.3 Scikit-Learn

Der er en del forskellige muligheder, når det kommer til programmer og frameworks, som gør det muligt at afprøve forskellige maskinelæringsalgoritmer. Jeg overvejede bl.a. Mallet[24], RapidMiner[25] og Weka[26], men jeg besluttede mig i sidste ende for at gøre brug af Scikit-Learn, da jeg således i mit arbejde kunne holde mig inden for Python. Ligesom preprocessing af data, der jo som skrevet finder sted i Python, er Scikit Learn nemlig funderet på Python.

Scikit-Learn er et maskinlærings-framework til Python, og det indeholder et væld af forskellige maskinlæringsalgoritmer, bl.a. nearest neighbours, Naive Bayes og Support Vector Machine, som jeg tidligere beskrev. Endvidere er der mulighed for nemt at træne og teste på forskellige former for *splits*.

7.3.1 Træning og test på et mindre datasæt

Datasættet består af 1309 instanser, hvilket ikke er meget, da der både skal trænes og testes på dette sæt. Der findes forskellige metoder til at håndtere dette problem (Wiebe & Frank, s. 149).

7.3.1.1 Holdout

Holdout-metoden gemmer en del af datasættet til test og træner på resten. Tommerfingerreglen er, at man afsætter 70% af datasættet til træning og 30% til test (Elkan, s. 1). Holdout-metoden er sårbar i den forstand, at man let kan være uheldig, når instanser til træningssættet skal vælges. Det er nemlig vigtigt, at hver klasse i trænings- og testsættet er repræsenteret i samme forhold, som den er repræsenteret i det fulde datasæt. For hvis tingene sættes på spidsen og alle instanser af en given klasse er fraværende i

[23] http://www.python.org/dev/peps/pep-0020/
[24] http://mallet.cs.umass.edu/
[25] http://rapid-i.com/content/view/181/190/
[26] http://www.cs.waikato.ac.nz/ml/weka/

træningssættet, vil classifieren efter træning håndtere instanser af denne klasse rigtig dårligt. Situationen bliver derudover yderligere forværret af, at klassen, der er fraværende i træningssættet, vil blive tilsvarende overrepræsenteret i testsættet.

7.3.1.2 Stratification

For at undgå ovenstående scenario kan man gøre brug af *stratified holdout*. Her holdes igen en del af datasættet tilbage til test, mens der trænes på resten og sørges for, at proportionerne klasserne imellem er de samme i trænings- og testsættet som i det fulde datasæt.

7.3.1.3 Repeated stratified holdout

En enkelt kørsel med stratified holdout giver dog stadig et usikkert billede af den givne classifiers præcision. Derfor kan man tilfældigt vælge fx 2/3 af datasættet til træning og 1/3 til test – begge er stratificerede – og så gentage denne proces et antal gange. Classifierens præcision vil så være gennemsnittet af resultaterne fra hver kørsel.

7.3.1.4 Stratified k-fold cross-validation

Med repeated stratified holdout vælger man ved hver kørsel et tilfældig trænings- og testsæt. Ved *cross-validation* splitter man én gang for alle datasættet op i *k* lige store såkaldte *folds*. En enkelt fold bliver taget fra til test, mens der trænes på resten. Dette gentages fold for fold, så der kun bliver testet på hver fold en enkelt gang. Classifierens præcision regnes endelig ud som et gennemsnit af resultatet fra hver af de *k* kørsler. Det er vigtigt at bemærke, at dette gennemsnit ikke er udtryk for en endelig og foreløbig optimal classifiers performance, som herefter vil kunne benyttes til klassifikation. I stedet er det normal procedure at træne en endelig classifier på det fulde datasæt, og så lade det opnåede gennemsnit fra cross-valideringen udgøre et uformelt estimat på classifierens performance (Elkan, s. 5).

I tilfælde som vores, hvor datamængden er begrænset, plejer stratified 10-fold cross-validation at være normen, da der er teoretisk evidens for, at dette antal giver det bedste estimat af den pågældende maskinlæringsalgoritmes præcision (Wiebe & Frank, s. 150 - Elkan, s. 4). Det er således den metode, jeg har valgt. Endvidere er det standardprocedure at køre den 10 gange, så classifieren i alt bliver kørt 100 gange på datasættet.

7.4 Beskrivelse af program

Håndteringen af det Excel-ark, der udgør rådata, spreder sig over to Python-filer - pre_processing.py og opinion_mining.py.

7.4.1 pre_processing.py

xlrd er et Python-bibliotek, som gør det muligt at trække data ud af et Excel-regneark. Hver række i Excel-arket indeholder en artikel, som udgøres af en XML-fil og nogle tilknyttede metadata. XML-filen indeholder informationer som titel, brødtekst, kilde, kategori, klasse (positiv, negativ, neutral) m.m.

7.4.1.1 Samling af artikler

Der bliver oprettet en klasse kaldet *artikel*, som bliver givet egenskaberne *tekst, overskrift, kategori, kilde, id_kode* og *klasse.* Informationerne fra hver XML-fil bliver trukket ud med regulære udtryk og gemt i et artikel-objekt. Som beskrevet i sidste kapitel, er der 329 dubletter blandt artiklerne – disse bliver naturligvis fjernet. Ligeledes bliver 55 neutralt opmærkede tekster fjernet.

7.4.1.2 Leksikon

For at lave en statistik over hvor mange gange, de forskellige termer i datasættet bliver benyttet, læses artiklerne ind ét for ét. Med et regulært udtryk bliver hver artikel bliver til at begynde med renset for følgende tegn (i kantede paranteser): ["«,.:;\'!?]. Herefter bliver artiklen splittet op i enkeltord, der bliver optalt i en ordbog, som angiver hvor mange gange de enkelte termer optræder i datasættet.

For at spare tid ved kommende kørsler, gemmes samlingen af artikler samt leksikon i separate filer, som hurtigt kan læses ind, så disse processer ikke skal udføres igen og igen.

7.4.2 opinion_mining.py

Først indlæses de filer, der indeholder artiklerne og leksikonnet. En funktion (find_feature_termer()) hvilke termer, der skal være en del af featurevektorerne, der repræsenterer artiklerne. Dette sker på baggrund af hvor hyppigt, termerne optræder i datasættet. Funktionen er sat til at medtage termer, der optræder mellem 50 og 400 gange, men dette er altså parametre, der kan ændres.

Funktionen doc_feature_vector() producerer nu for hver artikel en repræsenterende featurevektor. For hver feature – dvs. term – i vektoren er det angivet med 0 eller 1, om termen optræder i den pågældende artikel. Hver featurevektor bliver nu parret med en værdi, der angiver den korrekte (i forhold til Infomedias opmærkning) klasse.

Listen af featurevektorer bliver herefter gjort klar, så vi kan træne og teste en Naive Bayes og SVM-classifier med Scikit Learn.

7.4.3 Træning og test med Scikit Learn

Jeg benytter mig af de Naive Bayes og Support Vector Machine-classifiers, der er en integreret del af Scikit Learn. Disse classifiers er jo klart definerede og fungerer naturligvis, som de skal - upåklageligt – så når jeg træner og tester, er det jo i virkeligheden data og de features, der er udvalgt til at repræsentere artiklerne, jeg prøver af.

Som beskrevet i 7.3.1.4 kører jeg en 10-dobbelt stratified 10-fold cross-validation for både NB og SVM. Til at starte med træner jeg på det fulde featurerum, dvs. på træningssættet, hvor featurevektorerne indeholder 991 features. Dette giver følgende resultater:

Tabel 7.1

Antal features	NB	SVM
991	75.2%	71.9%

Umiddelbart bemærkes det, at de to classifiers' accuracy er ganske lav. Dette er formentlig et resultat af den lave datakvalitet, jeg postulerede og forsøgte at påvise i 6.5. Desuden er der nok også en del støj blandt de 991 features, der er blevet udvalgt til at repræsentere artiklerne.

Det bemærkes ligeledes, at SVM-classifieren scorer 2.3% lavere end NB-classifieren. Dette kan overraske, da SVM ofte betragtes som en algoritme, der håndterer data med et stort antal features godt. Men her ser det altså ud til, at NB-classifieren gør et bedre stykke arbejde.

En del af forklaringen kan igen være lav datakvalitet. Men hvad der måske er vigtigere er, at det er en betingelse, at det træningssæt, hvorpå SVM-algoritmen trænes, skal være tilstrækkeligt stort, hvis SVM-algoritmen skal kunne identificere det optimalt separerende hyperplan (Colas, s. 96). Og da træningssættet består af kun 1178 instanser, kan dette være en del af forklaringen på, hvorfor NB-classifieren overgår SVM-classifieren.

7.5 Alternativ til min implementering - Majority Voting på sætningsniveau

Adjektiver er som regel den ordklasse, der indeholder den stærkeste semantiske orientering (Gamon & Aue, s. 58). Men som det blev nævnt i forrige kapitel bliver denne type ord ikke brugt i lige så høj grad i journalistiske tekster, som de gør i fx anmeldelser, der normalt er genstanden, når der bliver lavet eksperimenter med opinion mining. Vi må således søge andre steder, hvis vi skal finde de markører, som giver anledning til, at en journalistisk tekst fremstår positiv eller negativ.

Jeg tager udgangspunkt i en idé af Gamon & Aue (s. 57), der igen tager udgangspunkt i Turney & Littman (2002). Sidstnævnte antog, at termer af samme semantiske orientering har en tendens til at optræde i samme dokument på tværs af sætningerne. Gamon & Aue gjorde den ekstra antagelse, at termer af modsat semantiske orientering har en tendens til ikke at optræde i samme sætning. Denne sidste antagelse svarer til antagelsen af, at sætninger af denne typen *'Jeg hader X'* og *'Jeg elsker Y'* (som kun indeholder sentimenter af én orientering) forekommer oftere end sætninger af typen *'Jeg hader X, men jeg elsker Y'*.

Sidstnævnte antagelse betyder, at sætninger som hovedregel er entydige i forhold til deres semantiske orientering. Jeg vil derfor konstruere en classifier, der fungerer på følgende måde:

7.5.1 Bag of bag of words

Også denne classifier er funderet på en bag of words-model – faktisk en bag of bag of words-model. For i stedet for at se hver tekst som en enkelt pose, hvori alle tekstens ord ligger og roder rundt, betragter jeg i stedet teksten som en pose, hvori der ligger adskillige poser fyldt med ord – én for hver sætning. Idéen er nu, at tekstens – altså hele artiklens –polaritet bliver afgjort via *majority voting*. Hver sætning i artiklen har en enkelt stemme, som er enten *positiv* eller *negativ* (eller *neutral*, skulle det ske, at sætningens polaritet ikke kan bestemmes), når det skal afgøres, om artiklen er positiv eller negativ.

Ligesom en artikels polaritet bliver bestemt ud fra de enkelte sætningers polaritet, bliver sidstnævnte afgjort af de enkelte ords semantiske orientering – altså ligeledes en majority voting. Ordenes semantiske orientering skal afgøres på samme måde, som i Turney og Littmans forsøg beskrevet i 5.3. Men hvor Turney og Littman vælger deres lister af positive og negative ord ud fra intuitionen, vil jeg vælge at konstruere disse lister på samme måde, som det i 5.2 er beskrevet, at Pang et. al gør det – valg af ord på baggrund af frekvensdistributionen og inspektion af denne.

7.5.2 Sætningsbaseret frem for dokumentbaseret

Mit valg om at lade den *endelige* afgørelse finde sted på baggrund af, hvordan de enkelte sætninger er orienteret (dvs. først afgøres sætningens orientering på baggrund af ordene i sætningen, og SÅ afgøres dokumentets klasse) er funderet på Gamon og Aues antagelse om, at termer af modsat semantiske orientering har en tendens til ikke at optræde i samme sætning. Dette har jeg kombineret med min egen antagelse af, at en artikel, hvor størstedelen af sætningerne er enten positive eller negative, er tilsvarende

positiv eller negativ. Det virker intuitivt som en plausibel antagelse, omend der naturligvis findes undtagelser, for eksempel ironi.

8 Afslutning

Udgangspunktet for denne opgave var at undersøge mulighederne for at lave et program, der kunne udføre opinion mining på de data omhandlende Nykredit, jeg var blevet givet. Det viste sig, at data var for ujævnt opmærket til, at det reelt kunne bruges som basis for træning af maskinlæringsalgoritmerne.

Endvidere blev det diskuteret, om det overhovedet giver mening at udføre opinion mining på data af denne type i forhold til det Infomedia skal bruge det til. Opinion mining omhandler jo normalt at afdække den subjektive holdning, en forfatter af en tekst kan have, men tilsyneladende er det ikke således, at Infomedias annotører har opmærket data. I stedet ser det det ud til, at det ville være mere givende at opmærke tekster i forhold til emne, evt. med en dybere analyse af, hvem emnet handler om – hvem er det eksempelvis i tekster omhandlende konkurser, der er ramt af konkursen.

Det er også klart, at der er brug for validerede, subjektivitetsbaseret opmærkede data på dansk, hvis det skal være muligt at udføre eksperimenter i stil med det præsenterede i afsnit 5.4. Skønt Wiebe og Riloffs eksperimenter i udstrakt grad var automatiserede, var en vigtig del af processen funderet på eksisterende opmærkede data. Dette er ikke muligt med dansksproget data.

9 Litteraturliste

Artstein, R., & Poesio, M. (2008). *Survey Article Inter-Coder Agreement* for *Computational Linguistics*, Volume 34 Issue 4, December 2008, p. 555-596, MIT Press Cambridge, MA, USA

Alpaydin, Ethem*: Introduction to Machine Learning*, 2nd edition, The MIT Press 2010

Burges, C. J. C. (1998): *A Tutorial on Support Vector Machines for Pattern Recognition*, in Data Mining and Knowledge Discovery, 2, s. 121-167

Christianini, N & Shawe-Taylor, J (2000): *An introduction to Support Vector Machines and other Kernel-based learning methods*, Cambridge University Press

Elkan, C. (2012), *Evaluating Classifiers*, forelæsningsnoter

Gamon, Michael, & Aue, Anthony (2005): *Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms*, in Feature Engineering for Machine Learning in Natural Language Processing, ACL-05

Guyon, Isabelle (2003): *An Introduction to Variable and Feature Selection*, Journal of Machine Learning Research 3, s.1157-1182

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002): *A gene selection method for cancer classification*, in Machine Learning *46*, s. 389–422

Hatzivassiloglou, V.; McKeown, K.(1997): *Predicting the Semantic Orientation of Adjectives*, in Proc. 35th Ann. Meeting Assoc. for Computational Linguistics, s. 174-181

Jurafsky, Daniel & Martin, James H. (2000): *Speech and Language Processing*, University of Colorado, Boulder

Kanayama, H. & Nasukawa, T. (2006): *Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis*, in Proc. Conf. Empirical Methods in Natural Language Processing, s.355-363

Klein, D. (n.d.). *Lagrange Multipliers without Permanent Scarring*

Kononenko, Ivan; Kukar, Matjaz (2007): *Machine Learning og Data Mining*, Woodhead Publishing

Lombard, M. (2005): *Practical Resources for Assessing and Reporting Intercoder Reliability* in Content Analysis Research Projects, (2002), 1–18.

Pang, B., Lee, L., Rd, H., & Jose, S. (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*

Pang, B., & Lee, L. (2008): *Opinion Mining and Sentiment Analysis* - (C. C. Aggarwal & C. Zhai, Eds.)*Foundations and Trends® in Information Retrieval*, *2*(2), 1–135.

Thue Poulsen, Ebbe (2001): *Funktioner af en og flere variable* - G.E.C. Gads Forlag, København.

Sandford Pedersen, Bolette; Wedekind, Jürgen; Bøhm-Andersen, Steen; Juel Henrichsen, Peter; Hoffensetz-Andresen; Kirhmeier-Andersen, Sabine; Kjærum, Jens Otto; Bie Larsen, Louise; Maegaard, Bente; Nimb, Sanni; Rasmussen, Jens-Erik; Revsbech, Peter; Erdman Thomsen, Hanne: *Det danske sprog i den digitale tidsalder*, Springer-Verlag, 2012

Solovej, Jan Philip (2002): *Supplement 2002*, Matematisk Afdeling, Københavns Universitet

Sørensen, M. (2003): *En Introduktion til Sandsynlighedsregning*, 3. udgave, Afdeling for Teoretisk Statistik, Københavns Universitet.

Turney, Peter (2002): *Thumbs up or thumbs down? Semantic orientation applied to unsupervised clas- sification of reviews*, , in Proc. 40th Ann. Meeting Assoc. for Computational Linguistics, s. 417-424

Turney, P. D., & Littman, M. L. (2002): *Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus*

Witten, Ian h. & Frank, Eibe (2005): *Data Mining - Practical Machine Learning Tools and Techniques*, Elsevier/Morgan Kaufmann, 2nd edition

Zhang, H. (2004): *The Optimality of Naive Bayes*, edited by Barr, Valerie; Markov, Zdravko in FLAIRS Conference 2004