NTNU
Det skapande universitetet

Utrekning av eit dokument sin sentimentverdi basert på setningsanalyse

Vidar Lillebø

Master i datateknikk
Innlevert: juni 2013
Hovedveiledar: Pinar Öztürk, IDI
Medveiledar: Arvid Holme, IDI

Noregs teknisk-naturvitskaplege universitet
Institutt for datateknikk og informasjonsvitenskap

Samandrag

Føremålet i denne oppgåva er å undersøkje metodar for sentimentanalyse av norske nyhende.

Det har blitt laga ein arkitektur og modell for gjennomføring av sentimentanalysen, der sentimentverdien for heile dokumentet har grunnlag i sentimentanalyse av setningane. For å teste systemet opp mot metodar som ikkje tek omsyn til setningane har det blitt gjennomført to eksperiment.

Resultata viser at det er tenleg å leggje analyse av setningane til grunn når ein skal klassifisere heile dokumentet. Veikskapar til modellen er identifisert, og det er kome med forslag til moglege utbetringar, og korleis analysen kan nyttast til å skape eit bilete av finansmarknaden.

Abstract

The purpose of this thesis is to investigate methods for sentiment analysis of norwegian news.

An architecture and model for performing sentiment analysis has been devised and implemented. The system finds a sentiment value for a document based upon sentiment analysis of the document's sentences.

The results show that it is feasible to base a document's sentiment value upon values for each sentence. The results also show for which circumstances the *bag-of-words* model fails for norwegian language. Based on the work presented, directions for possible improvements and future work is given.

Forord

Denne oppgåve er skrive ved Institutt for datateknikk og informasjonvitenskap, Noregs teknisk-naturvitskaplege universitet, innanfor studieretninga intelligente system.

Takk til vegleiar Pinar Özturk og medvegleiar Arvid Holme som har gitt rettleiing og råd igjennom utarbeidinga av denne oppgåva.

Vidar Lillebø
Trondheim, 11. juni 2013

Innhald

Figurar

Tabellar

Algoritmar

Kapittel 1

Introduksjon

I nyhende finnast der masse ustrukturert informasjon, som potensielt kan vere nyttig. I denne oppgåva skal vi sjå på korleis vi kan hente ut informasjon automatisk frå nyhende. Spesifikt vil det blir sett på om teksten uttrykkjer eit positivt eller negativt bodskap. Dette er kjent som polaritetsanalyse og er eit felt innan *sentimentanalyse*.

Sentimentanalyse er eit felt, der ein i korte trekk ønskjer hente ut informasjon knytt til korleis ting er omtalt. For å gjere dette nyttar ein gjerne metodar innan kunstig intelligens (AI). Dette feltet har i det siste fått mykje merksemd, grunna informasjonsmengda som har blitt tilgjengeleg via internett. Denne informasjonsmengda stig så fort at det er umogleg å gå igjennom alt for hand.

Metodar for sentimentanalyse er freista brukt i ulike domene, og med mange ulike mål. Til dømes kan metodane vere brukt i samband med overvaking av merkevarer og populariteten til visse produkt. Innan politikk kan metodane nyttast til å finne ut kva meiningar som er populære hjå folket, og kan til dømes brukast i samband med å føreseie utfall av val [1]. Eit anna bruksområde kan vere kvalitetssikring av kritikkar som er innsendt av brukarane av ein nettstad, der ein kan sjekke om det er samsvar karakter og tekst.

I forsøk rundt sentimentanalyse er det stor spreiing i kva som er brukt som inndata, og kva analysen til slutt gir informasjon om. Inndata kan til dømes vere Twitter, bloggar, nyhendeartiklar, osb. Utdata kan til dømes vere i kva grad teksten er subjektiv (det ein person tykkjer om eit emne) eller objektiv (faktaopplysingar) eller om teksten er

negativ eller positiv (polaritetsanalyse).

Nyhende kan analyserast på fleire måtar. Kvantitative og kvalitative mål på teksten kan hentast, døme på slike inkluderer sentiment, relevans og om nyhendet omtalar noko nytt [2]. Om slike mål representerer noko nyttig kan nyhende analyserast statistisk [3], og er av stor interesse innan finans.

Konkret for denne oppgåva, vil det bli sett på sentimentanalyse av nyhende på setningsnivå, og korleis analysen av setningane kan brukast for å klassifisere heile dokument. Ein arkitektur og modell for eit sentimentanalysesystem vil bli utvikla og implementert, og det vil bli sett på i kva grad dette kan nyttast i høvet norske nyhende. I denne oppgåva er domenet vidare avgrensa til finansrelatert nytt, dette då det er venta at tilpassing til eit gitt domene vil kunne gi ein meir treffsikker klassifikator [4]. Informasjonen som blir freista henta ut kan til dømes nyttast til overvaking av portefølje, til trading, eller generell analyse av marknaden. Det er klart at nyhende spelar ei stort rolle for investorane sine handlingar. Ved å finne ein modell for sentiment av nyhende, kan den nyttast både til å gjere egne slutningar basert på dataa, og å analysere andre sine handlingar [5, 6]. Målet her er kunne føreseie framtidig prisutvikling i finansmarknaden.

1.1 Motivasjon

Bakgrunnen for denne oppgåva er ønskje om å predikere aksjekursar på bakgrunn i nyhendestraumar frå internett. Det er utført ein god del forsking innan dette området [7, 3]. Felles for forskinga som ser på samanhengen mellom *breaking news* og intradagkursar er at alle nyttar heile dokument [7].

Ein vanleg framgangsmåte er å bruke *bag-of-words* teknikkar. Her vert dokumentet sett på som ein «sekk av ord» der ordfølgje ikkje spelar noko rolle. Samtlege modellar klassifiserer artiklar som positive, negative eller nøytral. Klassifiseringsresultatet har så vorte samanlikna med aktuelle aksjekursar. Dette i von om at positive nyhendeartiklar korrelerer med stigande aksjekursar, og negative med fallande. Ein innlysande veikskap som er omtalt i [7], er kvaliteten på klassifisering av dokumenta som positiv, negativ eller nøytral.

Større delar av litteraturen på dette område konsentrerer seg om relasjonen mellom nyhende og utviklingar i aksjekursen, men dette vil ikkje bli undersøkt i denne oppgåva.

Oppgåva konsentrerer seg om sentimentanalyse, og i kva grad metodane og modellen kan nyttast til korrekt sentimentanalyse av norske finansnyhende.

1.2 Oppgåvestruktur

Denne oppgåva er strukturert slik at kapittel 1 og 2 presenterer bakgrunn, motivasjon og mål for oppgåva. Kapittel 3 presenter relevant forsking på området, og aktuell teori som ligg bak arkitekturen og modellen som skal undersøkjast. Ein gjennomgang av arkitektur og modell er gitt i kapittel 4. Presentasjon av prototypen er i kapittel 5, og gjennomgang av eksperiment og resultat er i kapittel 6. Konklusjon basert på problemformulering er gitt i kapittel 7, og vert følgt av moglege retningar for vidare arbeid i kapittel 8.

Kapittel 2

Problemformulering og metode

Gitt tidlegare forsking rundt nyhende og finans, og motivasjon for oppgåva er det følgjande spørsmål eg ønskjer å undersøkje:

> *Kan ein fastsetje ein sentimentverdi for ein nyhendeartikkel ved å nytte sentimentanalyse på setningane til dokumentet?*

For å kunne svare på dette problemet vil følgjande bli utført:

P1 Utvikle ein modell og metode for å finne sentimentverdi av setningane.

P2 Bruke sentimentverdien av setningane for å finne ein sentimentverdi for dokumentet.

Det skal utviklast arkitektur og modell for klassifisering av dokument basert på sentimentverdi av setningane dokumentet består. Oppgåva vil ta føre seg norske nyhende, og modellen vil såleis vere tilpassa norsk språk.

Ein prototype for eit sentimentklassifiseringssystem vil bli implementert etter føringane gitt av arkitektur og modell. Denne prototypen vil saman med ulike test- og treningssett danne grunnlaget for eksperiment som vil vurdere sterke og svake sider ved modellen. Prototypen vil bli trent og testa på sett av dokument og setningar som har blitt markert som anten positive eller negative. Treningssettet består av artiklar som kan observerast i norske media.

Å finne sentimentverdi for dokumentet ved hjelp av setningane vil bli samanlikne med å bruke ein klassifikator som ser på heile dokumentet som *bag-of-words*. Testane skal danne grunnlaget for å finne ut i kva grad nyhendeartiklar kan klassifiserast med metodane og modellen som er skisserte.

Kapittel 3

Teori

Dette kapittelet gjev ein introduksjon til nyttig teori som er brukt i oppgåva. Først vil sentimentanalyse generelt bli presentert. Så kjem ein gjennomgang av sentimentanalyse innan finans, sidan det er dette bruksområdet som er motivasjonen bak oppgåva. Til slutt vil metodar og teori knytt til arkitektur og modell bli forklart. Dette inkluderer preprosessering, Naive Bayes-klassifikatoren, *features*, kryssvalidering, statistisk aggregering og vurderingsmål for yting.

3.1 Sentimentanalyse

Tekstgruvedrift (*text data mining*), er eit populært felt i moderne datateknologi. Dette då informasjonsmengda som er tilgjengeleg i dag veks så raskt at det er umogleg å analysere alt med bruk av konvensjonelle metodar. Samtidig finnast der sannsynleg store mengder aktuell kunnskap i denne informasjonen som kan bli henta ut og brukt.

Sentimentanalyse er ei oppgåve innan tekstgruvedrift som i det siste har fått mykje merksemd. Bakgrunnen for dette er at det gjennom internett har det vore ei oppblomstring av sider der innhaldet i stor grad er skapt av brukarane sjølve. Såleis har kritikkar, meiningar og liknande blitt tilgjengeleg for eit stort publikum. Denne teksten er også tilgjengeleg digitalt, noko som tyder at vi kan bruke ulike algoritmar til å analysere teksten. Å vite kva som finnast der ute er derfor både viktig og interessant. Slik kan ein få innsyn i korleis produkt, ytringar, eller handlingar blir oppfatta av folk. Ein annan

side ved dette er at folk ofte legg andre sine meiningar til grunn når dei skal gjere seg opp ei meining [8]. Av dette følgjer det ei rekkje moglege bruksområde.

Sentimentanalyse har fram til no først og fremst fokusert på tolking av meldingar. Denne typen tekster består hovudsakleg av subjektivt språk, handlar om eit avgrensa domene (t.d. film), og inneheld også ei numerisk vurdering, noko som gjer det enklare å sjekke kor godt analysen eigentleg fungerer. Denne typen analyse blir gjort med ganske gode resultat. Moglegheita for å bruke sentimentanalyse i finansdomenet har blitt utforska i fleire artiklar. Ulike metodar har blitt brukt, men det vanlege er å bruke bag-of-words modellen saman med andre vanlege metodar frå kunstig intelligens. Metodane som har blitt brukt kan grovt delast inn i to grupper; dei lærer sentimentverdiar frå eit klassifisert treningssett, og dei som brukar eit leksikon av ord med tilknytta sentimentverdiar.

Ved bruk av leksikon, konstruerer ein ein modell for sentiment basert på ord med kjent sentiment. Dette innebér å konstruere eller tilpasse ei ordbok (leksikon) med sentimentverdiar tilknytt kvart enkelt ord. Dokument kan då klassifiserast med å telje positive og negative ord, eller bruke meir sofistikerte metodar [9].

Når ein nyttar maskinlæringsmetodar kontruerer ein ein modell for sentiment, som vil bli lært opp på eit klassifiserte treningsdata. Eksempel på dette algoritmar som kan nyttast i dette høve er *Naive Bayes*, *Support Vector Machine*, *decision tree* og *vector-distance*-klassifikatorar. I tillegg til dette blir data ofte preprosessert for å forbetre analyseresultata. Metodar for dette kan for eksempel vere stemming, stoppordfilter, *Part Of Speech*-tagging og omskriving (t.d. ikkje bra -> ikkjebra). Metodane for preprosessering som er brukt i denne oppgåva er nærmare presentert i 3.3

Ein fordel med å bruke maskinlæringsmetodar er at ein enkelt kan endre og utvide språkmodellen, utan å måtte lage nye kunnskapsbasar som reflekterer den nye strukturen. Ved bruk av metodar som tek utgangspunkt i teljing av positive og negative termar kan det vere mykje å arbeid å lage ei kunnskapsbase.

Sentimentanalyse av nyhende fell noko til sidan for det primære fokusområdet til forsking rundt sentimentanalyse [10]. Hovudskilnaden i sjølve analysen ligg i at nyhende i større grad skildrar objektive meiningar, medan bloggar, *Twitter*, meldingar og liknande skildrar subjektive meiningar. Likevel er bruk av sentimentanalyse innan nyhende svært interessant. Området som er bakgrunnen for denne oppgåva er analyse av nyhende for bruk innan finans. Med slik analyse kan det skapast eit meir oppdatert

og komplett bilete av marknaden, noko som gir betre grunnlag for avgjersler med tanke på risiko, gevinst og fondsforvalting [9]. I mykje av litteraturen om bruk av sentimentanalyse innan finans, har føremålet vore å knytte analyse av nyhende opp mot utvikling av aksjekurs. Målet i mange av forsøka er å kunne nytte informasjonen om sentiment til artiklane i ein kortsiktig aksjehandelsstrateg (*day trading*) [11], men der også freista brukt i t.d. langsiktig porteføljeforvalting [12].

3.2 Sentimentanalyse i finans

Motivasjon til mykje av arbeidet rund finans og sentimentanalyse har vore å funne ein samanheng mellom sentiment som kjem til uttrykk i nyhende og utvikling i aksjekursar. Bakgrunnen her har vore at nyhende kan påverke investorar og *traderar* til å selje og kjøpe aksjer. Denne motivasjonen er det som også var utspringet for denne oppgåva.

Analyse av nyhende har også andre bruksområde innan finans. Dette inkluderer mellom anna marknadsundersøkingar, scenarioplanlegging, analyse av konkurrentar og generelt i finansrelaterte avgjersleprosessar.

Når det kjem til korleis teksten vert klassifisert i dette domenet er der i hovudsak brukt to metodar. Den eine metoden brukar maskinlæring til å lære seg korleis *features*, karakerstikkar ved teksten, påverkar sentimentverdien til heile dokumentet. Den andre metoden tek utgangspunkt i ordlister (leksikon) der kvart ord er klassifisert i kategoriar som positivt og negativt. Ordlistene her kan vere generelle, eller domenetilpassa. *The General Inquirer Dictionary* [13] er eit døme på ei generell, engelskspråkleg ordliste som er mykje brukt innan sentimentanalyse. Dette er ei ordliste som består av 1915 positive og 2291 negative ord. Denne ordlista er mellom anna brukt i Tetlock et al. [11], der frekvensen av *negative ord* er sett opp mot aksjekursen. I domenet finans har det vist seg at dei generelle ordlistene ikkje er heilt egna. I Loughran and McDonald [4] er det presentert ordlister tilpassa finansdomenet.

Dei vanlege måtane å analysere nyhende på er presentert i Das [14]. Denne presenterer mellom anna preprosesseringsmetodar, ulike klassifikatorar og vekting. Hovudlinene her er *bag-of-words*-modell på dokumentnivå.

3.3 Preprosessering av tekst

For preprosessering av teksten finnast der ei rekkje med metodar som gjer den meir egna for bruk i eit klassifiseringssystem.

3.3.1 Setningsdeling

Setningsdeling er ein prosess der ein vil finne starten og slutten på kvar setning [15]. Denne prosessen blir komplisert av at punktum ofte blir brukt til andre ting enn berre å markere slutten på ei setning, mellom anna til å markere forkortingar, i tidsuttrykk og i ordenstal.

Døme 3.1.

- Bruk av punktum i forkortingar gjer setningskilje ved punktum tvetydig.
- Dette gjer m.a. maskinell oppdeling av setningar vanskelegare.

Døme 3.1 viser korrekt oppdeling av setningar der punktum er brukt til andre ting enn å markere setningskilje.

3.3.2 Tokenisering

Tokenisering er ein relatert prosess, og går i hovudsak ut på å dele setninga opp i mindre element som vi skal handsame (tokens). På norsk markerer mellomrom skilje mellom ord, og det er ord vi gjerne ønskjer å nytte som *tokens*. Teiknsetjing og liknande kan gjere denne prosessen vanskelegare. Resultatet av denne prosessen er ei liste med *tokens*.

Døme 3.2.

> Tokens er den minste eininga ein handsamar. For å prosessere tekst er tokens vanlegvis ord og teiknsetjing.
>
> ⟶ Tokens; er; den; minste; eininga; ein; handsamar; .; For; å; prosessere; tekst; er; tokens; vanlegvis; ord; og; teiknsetjing; .;

3.3.3 Fjerning av stoppord

Stoppord er ord som filtrert vekk i under preprossering av tekst. I høve sentimentanalyse er dette ord som ikkje er tillagt spesiell tyding. I andre høve er stoppord vanlege, korte ord som *i, på, å, til,* osb. Denne prosessen er relatert til *feature selection* (sjå seksjon 3.5.2).

Det vanlege i forbinding med fjerning av stoppord er å bruke (lage) ei stoppordliste. Når ein går igjennom dokumentet vert ord finnast i denne stoppordlista tekne vekk.

3.3.4 Stemming

Stemming er å erstatte avleia ord med stammen av ordet [16]. Dette vil redusere talet på *features,* og tileigne sentiment til dei avleiingar som ikkje måtte vere observert i treningssettet. Stammen ordet vert erstatta med treng ikkje å vere den reelle stammen til ordet, sålenge alle avleiingar vert erstatta med same ord.

Døme 3.3. Døme frå dokumentasjonen til Snowball stemmer [17].

- opparbeide ⟶ opparbeid
- opparbeidede ⟶ opparbeid
- opparbeidelse ⟶ opparbeid
- opparbeider ⟶ opparbeid
- opparbeides ⟶ opparbeid
- opparbeidet ⟶ opparbeid

3.4 Naive Bayes-klassifikatoren

Vanlege klassifikatorar for klassifisering av tekst er mellom anna *Support Vector Machine* (SVM) og Naive Bayes. Naive Bayes er ofte førstevalet då den er enkel og rask, og presterer ganske godt.

Naive Bayes er ein probabilistisk klassifikator som tek utgangspunkt i Bayes' teorem. Denne er eit døme på ein *supervised* læringsmetode, dvs. læring under oppsyn der

klassifikatoren blir lært opp med eit klassifisert treningssett. *Naive Bayes* er ein vanleg klassifikator i samband med klassifisering av tekst.

For å klassifisere eit dokument, ønskjer vi å finne sannsynet for at eit dokument er av klasse C, gitt *features* (klassifiseringstrekk) $F_1, \ldots, F_n$.

$$p(C|F_1, \ldots, F_n) \tag{3.1}$$

Ved å bruke Bayes' teorem kan 3.1 bli skrive som

$$p(C|F_1, \ldots, F_n) = \frac{p(C)\ p(F_1, \ldots, F_n|C)}{p(F_1, \ldots, F_n)} \tag{3.2}$$

Sidan det vi ønskjer å finne er klassen med størst sannsyn, og divisoren ikkje er avhengig av C, samt at verdiane av F_i er gitt, er det berre dividenden som her er av interesse. Ved å bruke kjederegelen, og ein føresetnad om at F_i er betinga uavhengig av alle andre F_j der $i \neq j$, kan dividenden bli skrive som $\prod_{i=1}^{n} p(F_i|C)$. 3.2 vert såleis:

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C) \tag{3.3}$$

Det Z er ein skaleringsfaktor tilsvarande divisoren i 3.2. Vi ønskjer å velje klassen (hypotesen) med det høgste sannsynet. Dvs:

$$\text{classify}(f_1, \ldots, f_n) = \underset{c}{\text{argmax}}\ p(C = c) \prod_{i=1}^{n} p(F_i = f_i|C = c). \tag{3.4}$$

Ein veikskap med denne klassifikatoren er at den ikkje klarer å lære seg eventuelle samhandlingar mellom dei *features* som er gitt.

3.5 Features

Grunnlaget til klassifikatoren er eit sett med *features* (klassifiseringstrekk). Ein *feature* er ein karakteristikk for teksten, t.d. om eit gitt ord er med eller ikkje i teksten. *Features* er som oftast bolske eller numeriske.

For at klassifikatoren skal fungere godt, må settet med *features* (klassifiseringstrekk) den tek omsyn til vere godt. Dette involverer to prosessar, kalla *feature extraction* og *feature selection*. Om settet med *features* ikkje er relevant i forhold til problemet vi

ønskjer å løyse vil klassifikatoren basere seg på særeigne trekk ved treningssettet, og vil ikkje kunne generaliserast til nye dokument, eit fenomen som er kjent som *overfitting*.

3.5.1 Feature extraction

Feature extraction er i maskinlæring ein prosess der data vert omgjort til ei rekkje karakteristikkar som representerer dei delar av data ein ønskjer å analysere. Dette settet med karakteristikkar er ofte kalla *feature vector*. Algoritmane arbeider med desse vektorane for å finne samanhengar som kan nyttast til å til dømes klassifisere inndata.

Innan tekstprosessering ønskjer ein å finne ein representasjon som tek omsyn til tydinga til teksten, og er såleis sterkt knytt opp mot ein språkmodell [18]. Uthentinga av *features* er ofte gjort på data som er preprosessert på ulike måtar (sjå 3.3).

Ein vanleg språkmodell for bruk til klassifisering av tekst er *bag-of-words*-modellen. [9]. *Features* som vert henta ut frå teksten i denne modellen er *unigram*, eller orda i teksten. Desse vert representert i ein vektor, der verdiane i vektoren fortel om ordet. Desse verdiane kan vere bolske (om ordet er til stades i setninga eller ikkje), talet på gongane ordet er brukt i teksten (frekvens), eller ei anna vekting som t.d. *tf-idf*. Ordfølgje og grammatikk vert såleis ignorert i denne modellen.

Andre features som er vanleg å nytte i eit tekst klassifiseringssystem er bigram (n-gram), eller ord med tilknytt *Part Of Speech*-tag. Ved å bruke kunnskapsbasar som til dømes Wordnet [19], kan ord i teksten erstattast med meir generelle konsept (t.d. katt og hund erstattast med husdyr). Synonym kan slåast sammen til eit sett, og nokon som reduserer antalet features, og maskinlæringa treng berre å lære seg tydinga til gruppa med ord, og ikkje kvart enkelt.

3.5.2 Feature selection

Feature selection er å velje det ut settet med dei mest relevante *features*. Dette er i gjort i tanke om at enkelte *features* anten er irrelevante eller redundante. Å fjerne desse vil kunne fjerne støy frå inndata, og hindre *overfitting*.

Det finnast ulike metodar for å finne nyttige *features*, men i dette prosjektet er det informasjon om *information gain* som er nytta, ein metode som har grunnlag i statistisk korrelasjon.

Pearsons chi-squared-test (likning 3.5) er eit mål på kor uavhengige *features* er frå klassa.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.5}$$

3.6 Kryssvalidering

Kryssvalidering er ein effektiv metode for å estimere ytinga til ein klassifikator [20]. Metoden som er brukt er kjent som *10-fold cross-validation*, der treningssettet blir delt opp i 10, 9 av desse delane blir brukt til å trene klassifikatoren, og blir testa mot den siste delen. Dette blir gjort 10 gangar, og ved å ta snittet av desse gangane har ein eit godt mål på prestasjonen til klassifikatoren.

3.7 Aggregering

Aggregering er innan statistikk ulike metodar for å kombinere data. I høve denne oppgåva tyder det å kombinere data om setningane til ein verdi som er representativ for heile dokumentet. Døme på slike metodar er aritmetisk snitt, median, typetall og liknande.

3.8 Vurderingsmål for yting

Innan vurdering av klassikatorar er der tre mål som er vanleg å nytte: *precision*, *recall* og *accuracy*. Både *precision* og *recall* er mål på relevans, medan *accuracy* er eit mål antalet som er rett klassifisert.

	Faktisk klasse	
Predikert klasse	TP (sann positiv)	FP (falsk positiv)
	FN (falsk negativ)	TN (sann negativ)

Tabell 3.1: Typer feil i klassifisering

Positiv og negativ referer til predikatet til klassifikatoren, sann og falsk referer til om predikatet er høvesvis rett eller gale. Matematiske definisjonar er gitt i likninga-

ne 3.6a-c [21, s. 138]. Merk at desse definisjonane er noko annleis ein dei brukt innan informasjonsattfinning (*Information Retrieval*).

$$Precision = \frac{TP}{TP + FP} \tag{3.6a}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.6b}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN} \tag{3.6c}$$

Kapittel 4

Arkitektur/Modell

Dette kapittelet fortel om språkmodellen som er nytta, og dei ulike prosessane og metodane som er vert brukt til å klassifisere teksten.

4.1 Språkmodell

Ei setning kan ofte tolkast positivt eller negativt. Dette gjerast utifrå ord og setningsbygnad. Setningsbygdnad, ord og ordbøying er komplisert, og å lage ein komplett modell for dette er difor vanskeleg. For å kunne klassifisere setningar som positive eller negative er det brukt ein forenkla modell frå språket som skal kunne nyttast til å gi polariteten til setninga.

Setningane vert omgjort til ein binærvektor-representasjon der dimensjonane representerer orda, og verdien viser om det aktuelle ordet er med i setninga. Informasjon som ligg i rekkjefølgja av orda vert dermed ikkje tatt omsyn til. Denne modellen er kalla *bag-of-words*, og er svært vanleg innan tekstklassifisering.

Ved bruk av *feature selection* blir det valt ut ei rekkje med *features* det skal takast omsyn til. Sidan orda i denne samanhengen tilsvarar *features*, kan denne prosessen skildrast som å finne orda som best representerer meininga til setninga.

Finans er eit domene prega av mykje tal, og desse tala er gjerne referert til i artiklane vi skal handsame. I dette systemet blir desse tala ignorert. Men, slike tal vert gjerne skildra med positivt og negativt språk i denne omliggjande teksten.

Døme 4.1. Ein artikkel om kvartalsresultata til Statoil kunne ha tittelen «Uhyre sterke tall fra Statoil i Q4», eventuelt «Rød bunnlinje for Statoil» i eit dårleg kvartal.

Såleis vil noko av informasjon som ligg i desse tala bli teke omsyn til. Det er føreset at andre system kan få tak i slike tal på strukturert form, og analysere på ein betre måte enn det eit tekstanalyseringsverkty vil gjere.

Ved å trene opp klassifikator på treningssettet, vil dermed visse ord utmerke seg som spesielt positive – eller negative. På grunn av måten klassifikatoren fungerer på, vil det at eit ord ikkje er ein del av dokumentet også påverke resultatet. Døme på korleis dei mest signifikante orda er representert i den endelege klassifikatoren er gitt i tabell 4.1. I praksis er dette dei mest observerte positive og negative orda. For informasjon om *Naive Bayes*-klassifikatoren sjå 3.4.

I tabellen under er ordet «økonomi» tillagt positiv tyding. Ved større treningssett vil dette ordet mest sannsynleg vere likt fordelt mellom klassene, og vere tillagt ei meir eller mindre nøytral tyding, og såleis gi ein meir korrekt klassifikator. Klassifiseringstrekket «inneheld(økonomi)» bør fjernast då den i grunn skulle vere irrelevant når det kjem polariteten setninga uttrykkjer.

Feature (F)	$P(positiv\|F)$	$P(negativ\|F)$	$P(positiv\|\neg F)$	$P(negativ\|\neg F)$
inneheld(steg)	0.09741	0.00596	0.90059	0.99204
inneheld(falt)	0.00994	0.09343	0.98807	0.90457
inneheld(positive)	0.04572	0.00596	0.95228	0.99204
inneheld(rødt)	0.00596	0.03777	0.99204	0.96023
inneheld(ned)	0.01391	0.08151	0.98409	0.91650
inneheld(opp)	0.08151	0.01391	0.91650	0.98409
inneheld(nedgang)	0.00596	0.03379	0.99204	0.96421
inneheld(svak)	0.00596	0.02982	0.99204	0.96819
inneheld(økonomi)	0.00994	0.04572	0.98807	0.95228
inneheld(lavere)	0.00596	0.02584	0.99204	0.97216
inneheld(imidlertid)	0.02584	0.00596	0.97216	0.99204
inneheld(oppgang)	0.03777	0.00994	0.96023	0.98807
inneheld(positivt)	0.02186	0.00596	0.97614	0.99204
inneheld(godt)	0.02186	0.00596	0.97614	0.99204

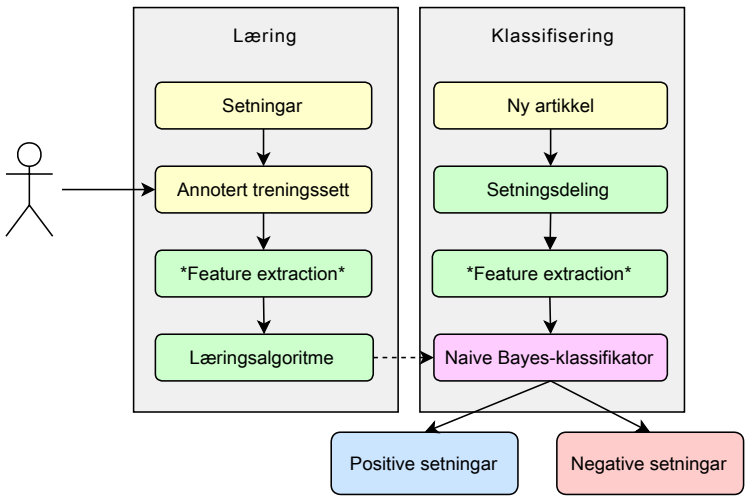
Tabell 4.1: Døme på positive og negative *features*, med tilknytte sannsyn i klassifikatoren

4.2 Overordna struktur

Eit oversyn over prosessane relatert til opplæring og klassifisering er gitt i figur 4.1.

For opplæring nyttast der eit sett med setningar annotert med klasse positiv/negativ. Sidan det ikkje finnast eit ferdig treningssett som kan nyttast vert eit slikt treningssett laga manuelt. Dette treningsettet vert så transformert til ein *feature vector*-representasjon som vert nytta til å lære opp klassifikatoren.

For klassifisering vert ein artikkel delt i opp i setningar som vi bruker same *feature extraction* metode på som i læringsdelen. Basert på denne vert setninga klassifisert som positiv eller negativ.



Figur 4.1: Dei ulike prosessane i systemet

4.3 Preprosessering

I opplæringsprosessen tek systemet utgangspunkt i setningar med tilhøyrande klasse. Desse setningane vert *tokenisert* (sjå seksjon 3.3.2) før dei vert prosessert vidare av systemet. Om stemming har vore brukt kjem dette også inn her.

Når heile dokument skal klassifiserast vert setningane delt opp og *tokenisert* før det vert henta ut *features* som vil bli nytta til klassifisering av enkeltsetningar.

4.4 Trening av klassifikator

For bruk i dette systemet er det valt å bruke ein *Naive Bayes*-klassifikator. Ein introduksjon til klassifikatoren sin verkemåte er gitt i avsnitt 3.4. Kjernen i opplæringsprosessen er å telje antalet *features* som kan observerast i treningssettet for kvar klasse.

4.4.1 Ekstrahering av klassifiseringstrekk

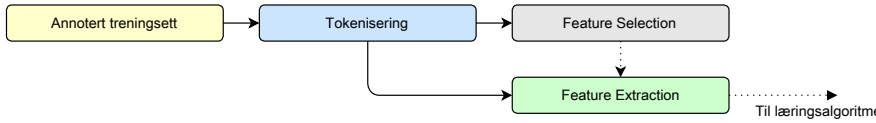
For å kunne nytte ein klassifikatoren må *featurs* (klassifiseringstrekka) hentast ut frå setningane (*feature extraction*). Dette er ei prosess som må gjerast for setningane både når dei klassifiserast, eller nyttast til opplæring.

Feature extraction er i hovudsak ein ein-til-ein funksjon som ser slik ut:

$$t_n = (s_n, c_n) \Rightarrow T_n = ((f_{n,1}, \ldots f_{n,i}), c_n) \tag{4.1}$$

der t_n er ei klassifisert setning, s_n er setningar og c_n er klassen. Den nye representasjonen, T_n er eit sett med features $(f_{n,i}, \ldots f_{n,i})$ og tilhøyrande klasse.

I dette systemet vert *feature selection* utført på treningsdataa. Valte features vert så henta ut i *feature extraction* prosessen (sjå figur 4.2).



Figur 4.2: Samanhengen mellom *feature extraction* og *feature selection*

Feature selection

Reduksjon, eller val av *features* er ein prosess der ein fjernar redundante og irrelevant *features* (*feature selection*). Dette for å forhindre overfitting, og forbetre grunnlaget til

klassifikatoren. Å basere denne prosessen på «information gain» er vanleg. Konseptet er presentert i 3.5.2.

Dette er implementert som i algoritme 1. I hovudsak går dette ut på å telje ord (termar) som opptrer i setningar av dei ulike klassene.

Data: Annotert treningssett
Resultat: Mest informative *features*

begin
  tf_t — Antal forekomstar av term t
  cf_c, t — Antal forekomstar av term t, gitt klasse c
  for $(s, c) \in$ *treningssett* **do**
    for $t \in$ *tokenize(s)* — Tokenisér setninga
    do
      $tf_t \longleftarrow tf_t + 1$
      $cf_{c,t} \longleftarrow cf_{c,t} + 1$
    end
  end
  pt $\longleftarrow$ antal termar i $cf_{positive}$ — tal på «positive» ord
  nt $\longleftarrow$ antal termar i $cf_{negative}$ — tal på «negative» ord
  tt $\longleftarrow$ pt + nt
  for $(t, f) \in tf$ — t: term, f: antal
  do
    $O_{positive} \longleftarrow cf_{positive,t} \cdot (tt + cf_{positive,t} - f - pt)$
    $E_{positive} \longleftarrow (f - cf_{positive,t})(pt - cf_{positive,t})$
    $O_{negative} \longleftarrow cf_{negative,t} \cdot (tt + cf_{negative,t} - f - nt)$
    $E_{negative} \longleftarrow (f - cf_{negative,t})(nt - cf_{negative,t})$
    $score_t \longleftarrow \frac{(O_{positive} - E_{positive})^2}{E_{positive}} + \frac{(O_{negative} - E_{negative})^2}{E_{negative}}$
      dette tilsvarar χ^2 er som definert i likning 3.5
  end
  velg N beste termar i frå $score$ (høgast verdi)
end

Algoritme 1: Val av klassifiseringstrekk

Feature extraction

Korleis setningane blir omgjort til vektorane som vert brukt til klassifisering er skildra i algoritme 2, viser korleis dette kan bli gjort i kode, gitt settet med valte features som i algoritme 1.

Data: setning som liste av tokens
Resultat: Vektor f for inndata

```
1 begin
2     for ord ∈ valte features do
3         if ord ∈ setning then
4             f_ord ⟵ true
5         else
6             f_ord ⟵ false
7         end
8     end
9 end
```

Algoritme 2: Frå setning til feature-vektor

4.5 Klassifisering av setningar

Klassifisering av setningar foregår som vist i klassifiseringsdelen av figur 4.1. I utgangspunktet gir *Naive Bayes*-klassifikatoren sannsyn for at setninga er av dei ulike klassene. Om $P(positiv) > P(negativ)$ vert setninga klassifisert som positiv, og tilsvarande for negativ.

Figur 4.3 viser hovudlinene i korleis klassifikatoren vil fungere på setningar observert i datasettet. Første punkt viser setninga, delane av setningane som utifrå intuisjon påverkar sentimentverdien til setninga i positiv retning er markert i blått. Neste punkt viser dei *features* som vert henta ut. Verdt å merke seg er at ordet «kanonoverskudd», ikkje er med då det ikkje finnast i treningssettet. Setninga vert klassifisert til positiv med grunnlag i at orda «doble», og «omsetningen» er tilstades i setninga.

1. Kjell Inge Røkkes Aker Solutions tror de kan doble omsetningen og skaffe seg kanonoverskudd.
2. inneheld(doble) ∧ inneheld(omsetningen)
3. positiv: **0.953**, negativ: **0.047**

Figur 4.3: Gjennomgang av positivt døme [Delar av artikkel frå DN.no]

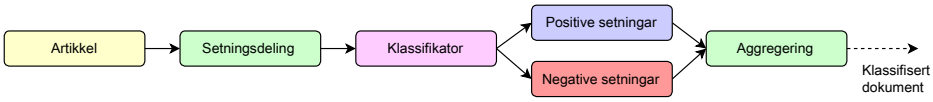
Figur 4.4 syner tilsvarande for eit negativt døme. Legg merke til at «oljeanalytiker» og «meget» er i dette tilfellet feilaktig oppfatta som negative ord av klassifikatoren.

1. Det var en meget negativ rapport, sier oljeanalytiker Jim Ritterbusch i Illinois til nyhetstjenesten.
2. inneheld(meget) ∧ inneheld(negativ) ∧ inneheld(oljeanalytiker)
3. positiv: **0.002**, negativ: **0.998**

Figur 4.4: Gjennomgang av negativt døme [Teksten er henta frå DN.no]

4.6 Klassifisering av dokument

Gitt eit dokument $D = \langle s_1, \ldots, s_n \rangle$, der D er dokumentet og s_n er ei setning n, og polariteten til kvar setning er kjent. Vi ønskjer her å finne polariteten til heile dokumentet. Polariteten til dokumentet er gitt ved polariteten til setningane og i kva grad kvar setning er viktig for heile dokumentet.



Figur 4.5: Klassifisering av dokument er gitt av setningane

Generelt kan sentimentverdien til eit dokument uttrykkast slik:

$$S(d) = \sum_{i=1}^{n} S(s_i)\, w_i \tag{4.2}$$

der S er sentimentverdi, s_i er setning i, og w er vekting. Om setningane setningane blir vekta likt, er polariteten til heile dokumentet definert til å vere snittet av polariteten til setningane som dokumentet består av.

Det vert skild mellom to ulike dokument – dei med ingress og dei utan ingress. Det er kjent at sentiment i artikkelteksten, ingress og tittel til ei viss grad attspeglar kvarandre.

Den første metoden som blir brukt for å klassifisere dokumentet er at dokumentet er snittet av tittel, ingress og teksten. For artiklar utan ingress gjeld snittet av tittel og tekst. For dei ulike delane vert setningane vekta likt. Setningane i tittel vert såleis vikti-

gare enn setningane i ingressen, og setningane i ingress vert viktigare enn setningane i teksten.

Aggregeringsmetodane som vert brukt:

1. Gjennomsnitt av gjennomsnitt av sentimentverdiane av setningane i tittel, ingress og hovudtekst (**metode 1**)

2. Gjennomsnitt av medianverdiane til tittel, ingress og hovudtekst (**metode 2**)

Desse metodane vil verte sett opp mot **metode 3**, som der heile dokumentet vert sett på, og klassifisert ved hjelp av bag-of-words-modellen. Den metoden ser såleis ikkje på setningane slik som metode 1 og 2 gjer, og vil fungere som samanliknings-grunnlag opp mot systemet føreslått.

I tillegg til metodane forklart over vil det bli sett på kor godt ein kan klassifisere artiklane ved å berre sjå på tittel og ingress, referert til som **metode 4**.

For alle desse metodane gjeld det at om artikkelen ikkje har ingress er det berre tittel og hovudtekst som vert sett på. For metode 1 og 2 tyder dette at det er snittet av høvesvis snittverdi og medianverdi for tittel og ingress som vert brukt.

Kapittel 5

Prototype

For å kunne teste den struktur, modell og metodane skildra i kapittel 4, har det vorte lage ein prototype på systemet. Dette svarar til **P1** i kapittelet 2.

Følgjande verkty og programvarebibliotek har blitt nytta:

- Python 2.7.4
- Qt 4.8.4
- PyQt4 4.10
- NLTK 2.0

Python er eit programmeringsspråk. Qt er eit programvarerammeverk mykje brukt til å lage program med grafiske brukargrensesnitt (GUI). PyQt4 er ein programvare som gjer at ein kan nytte Qt i Python. NLTK er er eit verktysett for prosessering av naturleg språk.

Prototypen som er laga er eit program med eit grafisk brukargrensesnitt der ein kan studere detaljane rundt korleis teksten blir klassifisert. Resultat av dei ulike prosessane vert vist og kan kontrollerast. Med grunnlag i den opptrente klassifikatoren vert positive og negative ord markert, og ein kan såleis sjå på grunnlaget klassifikatoren nyttar når den skal klassifisere ei gitt setning.

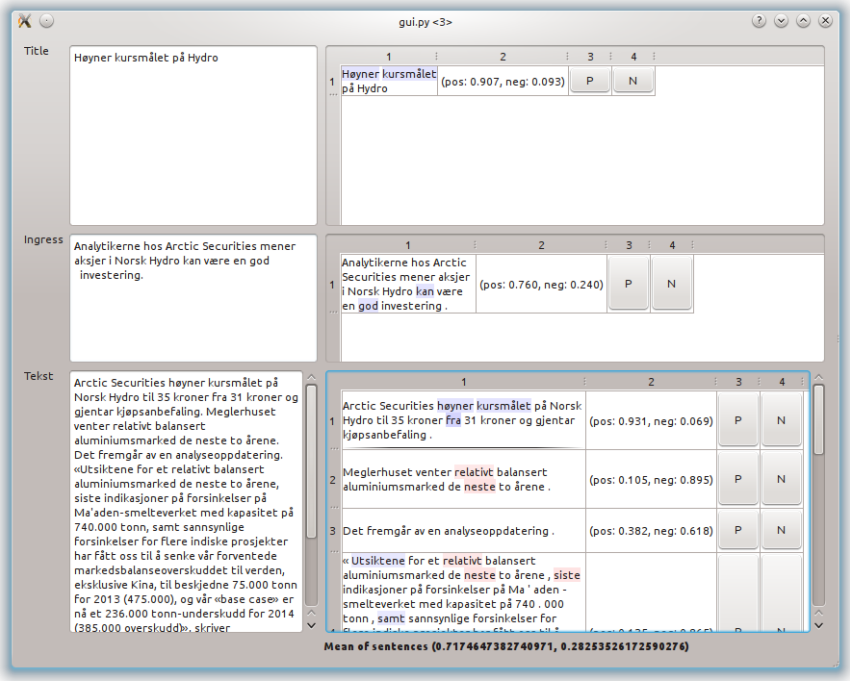
Programmet er samansett av to delar – klassar som omhandlar grensesnittet og klassar som omhandlar sjølve klassifiseringa av tekst. Delane som tek føre seg klassifiseringa er implementert tett opp til forklaringane av arkitekture og modellen i kapittel 4.

Systemet tek utgangspunkt i preprosessert tekst som vert delt opp i setningar. For å dele setningane har NLTK [22], med norsk *punkt tokenizer* vorte brukt. Der stemming har blitt brukt har *Snowball stemmer* blitt brukt, denne også tilgjengeleg gjennom NLTK. NLTK [22] sin implementasjon av *Naive Bayes*-klassifikatoren har blitt brukt.

I tillegg har det blitt laga ein *crawler* som har traversert vevsidene til DN.no, lasta ned og lagra nyhendeartiklar på eit format som prototypen kan nytte. Her følgjer ein linkane på vevsidene, finn artiklar, og deler dei opp i tittel, ingress og artikkeltekst, som ein lagrar til fil slik at dei kan nyttast av systemet. I tillegg til verktya nemnt over har *BeautifulSoup4* blitt tatt i bruk for å lette denne prosessen.

Figur 5.1 syner korleis prototypen deler teksten opp i setningar og klassifiserer einskilde setningar, som vert slått saman til ein verdi. Teksten som vert klassifisert kan endrast på for å sjå korleis dette endrar måten teksten vert klassifisert på. Setningar som er positive eller negative kan leggast til i treningssettet. Dette fører til at det vert trent ein ny klassifikator, og dokumenta ein analyserer vil bli oppdatert for å reflektere den nye informasjonen.

Såleis vil ein raskt få eit oversyn over kva grunnlaget for klassifiseringa er. Positive og negative vert markert med høvesvis blå og raud bakgrunn for å korleis systemet tyder dei ulike orda.



Figur 5.1: Analyse av eit dokument i det grafiske brukargrensesnittet

Kapittel 6

Eksperiment og resultat

Vi har to hovudbolkar med eksperiment. Undersøking rundt sentimentanalyse av einskilde setningar er teke føre seg i eksperiment 1, medan undersøkingar rundt aggregering og klassifisering av heile dokument er teke føre seg i eksperiment 2.

Klassiseringssystemet som er implementert i kapittel 5 er brukt. Dette er implementert etter linene gitt i kapittel 4. Med bakgrunn i problemformuleringa og **P2** (kapittel 2) vert dette systemet så brukt til å:

1 Teste ytelsen av det føreslåtte systemet for sentimentanalyse av setningar.

2 Undersøke om sentiment av setningar er nyttig for klassifisering av heile artiklar.

Det vil bli sett på konkrete eksempel og kva variasjonar av modell og metode som best. I tillegg det bli undersøkt kva veikskapar som finnast ved systemet som er implementert.

6.1 Datasett

Artiklar frå DN.no har blitt lasta ned ved å bruke eit sjølvskrive program som går igjennom vevsidene og lagrar artiklane til fil. Dette datasettet består av 14467 artiklar. Artiklane består av ein tittel og ei artikkeltekst. Nokre av artiklane har også ein ingress. Døme på korleis artiklane kan sjå ut er gitt i vedlegg 1. Frå dette datasettet er artiklar

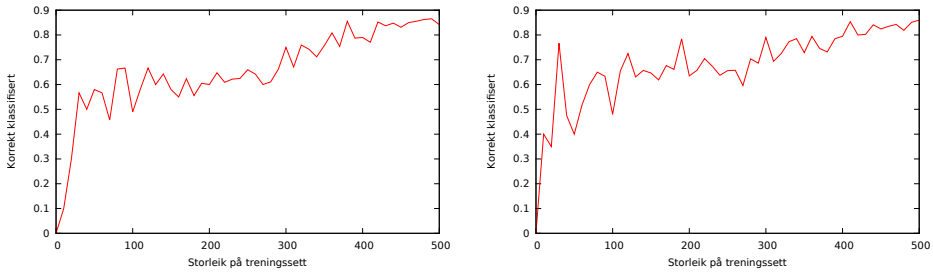
og setningar plukka ut for å lære opp, og teste klassifiseringssystemet. Forsøka som er gjort er med andre ord køyrd på artiklar og setningar som kan observerast i norske media.

Treningssettet er 500 setningar vilkårleg utvalt i frå datasettet. Setningane har ei klar positiv eller negativ tyding, som dei har blitt annotert med. Treningssettet består av like mange negative som positive døme.

6.2 Eksperiment 1

Eksperiment 1 tek sikte på å vurdere om arkitektur og modell er egna til å klassifisere einskilde setningar, og såleis sikre grunnlaget for eksperiment 2. Setningane blir her sett på som *bag-of-words*, og der *features* vert henta ut, og setningane klassifisert slik som skildra i kapittel 4.

Kryssvalidering med treningssett av ulike storleikar vil bli nytta til vurdere klassifikatoren og treningssettet. Kryssvalidering er forklart i 3.6. Å bruke denne metoden gir ein ein god indikasjon korleis klassikatoren presterer. Ved å sjekke dette målet opp mot ulike storleikar på treningssettet, kjem evna systemet har til å lære til syne.



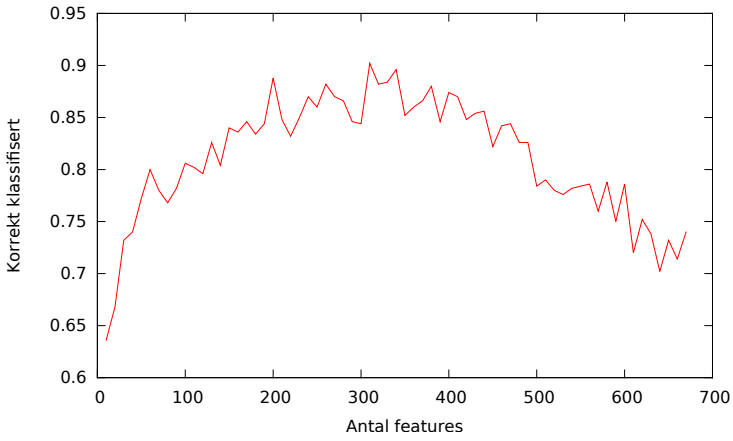
(a) Utan stemming (b) Med stemming

Figur 6.1: Yting ved treningssett av ulike storleikar, med og utan stemming

Figur 6.1(a) syner korleis klassifikatoren si evne til å klassifisere setningar er, gitt ulike storleikar på treningssettet. Der settet til som vert brukt er mindre enn heile treningssettet på 500, er det eit vilkårleg utval som blir nytta. Ein kan vidare sjå at det krevst eit treningssett av ein viss storleik for at klassifikatoren skal gi fornuftige resultat, og at systemet lærer seg å klassifisere etterkvart som storleiken på treningssettet aukar.

Modellen implementert har med andre ord til å lære av treningsettet. Figur 6.1(b) syner tilsvarande der *stemming* er freista brukt. Ved bruk av heile treningssett gav ikkje *stemming* ei klar betring i ytinga, men ved mindre treningssett er der indikasjon på at *stemming* betrar resultatet. Til trass for dette er *stemming* ikkje nytta elles i eksperimenta. I begge desse køyringane er det nytta 400 *features*.

Val av *features* har mykje å seie for evna klassifikatoren har til å klassifisere setninga. I denne oppgåva er valt å velje ut *features* basert på *information gain*, slik som forklart i seksjon 4.4.1. I treningssettet er det i utgangspunktet 679 ord som er observert meir enn ein gong, og er såleis potensielle *features*. Her ønskjer ein å vere viss om at å redusere antal *features* gir ein betre klassifikator, og finne det ideelle antalet som kan nyttast. Figur 6.2 viser korleis ytinga til klassifikatoren varierer med antal *features*. Ikkje uventa gir for lite eller for mange *features* dårlegare resultat. Grafen viser ei betring frå 75% korrekt klassifiserte setningar til 89% korrekt klassifisert ved å redusere antal brukte *features* til omlag 300-350. Dette viser at *feature selection* har stor innverknad på klassifikatoren. Med grunnlag i dette er det difor brukt 350 *features* vidare.



Figur 6.2: Features og yting

6.2.1 Feilklassifisering av setningar

Systemet klassifiserer ei rekkje setningar feil. Under følgjer døme på slike setningar, og resonnering rundt kva grunnen til dette kan vere.

Døme 6.1. *DNO hoppet på negativ børs.*

Dette dømet viser at ei enkelt setning kan omtale fleire emner med ulikt sentiment. DNO er omtalt positivt, medan børsen generelt negativt. Setninga er klassifisert negativ med grunnlag i ordet *negativ*. Ordet *hoppet* er ikkje attkjent som eit positivt ord.

Døme 6.2. *Gir opp vindkraft etter tap på 200 millioner kroner.*

Opp er i utgangspunktet attkjent som eit positivt ord, saman med *gir* skal dette eigentleg vere tolka som negativt.

Døme 6.3. *Knalltall fra Norwegian.*

På norsk nyttar ein samskriving. Når journalistane skal vere kreative så kan det føre til ord som ikkje er observert i treningssettet.

Døme 6.4. *Norges Bank har undervurdert de positive tegnene.*

Denne setninga ytrar ei misnøye rundt handlingane til Noregs Bank, men ordet *positiv* får denne setninga til å bli klassifisert som positiv.

Døme 6.5. *Statoil håper å kunne skryte av nye gigafunn.*

Skryte og *nye gigafunn* indikerer noko positivt, men orda er ikkje å observere i treningssettet.

Døme 6.6. *Dette er selve indrefileten i et veldig spennende område.*

Setningar som er positive i overført tyding er vanskeleg. *Indrefilet* kan neppe vere noko negativt, men dette har ikkje systemet lært seg av treningssettet.

Døme 6.7. *Oljeprisene snur fra en ukes bunnivå.*

Kva vei ordet snur påverkar setninga er avhengig av ordet «bunnivå» som kjem seinare i setninga. Snur er attkjent som eit negativt ord av systemet.

Dette viser at språkmodellen på langt nær klarer å plukke opp sentiment i litt meir kompliserte setningar.

6.2.2 Oppsummering eksperiment 1

Eksperiment 1 har sett på korleis setningane vert klassifisert i dette systemet føreslått. Det er vist at systemet har evne til å lære seg å klassifisere setningar. Antal features brukt er optimalisert mot treningssettet, og denne parameteren er brukt vidare.

Ved å sjå på setningar klassifisert feil av systemet er det identifisert problem ved språkmodellen. Problema er i stor grad knytt til ord som ikkje vert nytta som features då dei ikkje er å observere i treningssettet av ulike grunnar, men der er også døme på setningar der ein må ta omsyn til følgja, eller fleire ord i setninga for å kunne klassifisere rett.

6.3 Eksperiment 2

For eksperiment 2 har eit sett på 50 artiklar blitt valt ut for å samanlikne metodane som nyttar setningsbasert sentimentanalyse med metodane som ser på heile dokumentet under eit. Det vil i dette eksperimentet bli testa forhold direkte knytt til problemformuleringa i kapittel 2.

Metode 1 og **metode 2** nyttar aggregering av sentimentverdiane av setningane for å finne klassen til dokumentet. Skilnaden mellom metode 1 og metode 2 ligg i at **metode 1** nyttar gjennomsnittet av sentimentverdiane til setningane, medan **metode 2** nyttar medianverdien. **Metode 3**, nyttar trent klassifikator på heile dokumentet, og er den vi ønskjer å samanlikne mot. **Metode 4** er som metode 3, men tittel og ingress er sett på som heile dokumentet. Dei ulike metodane er nærmare forklart i 4.6.

Resultata for kvar artikkel er presentert i tabell 6.1. Det er samanlikna fire metodar for klassifisering av artiklane. Hovudgrunnen til at metode 3 og 4 jamt over gjev høgare sannsynsverdiar enn metode 1 og 2 er at desse i større grad ignorerer setningar der polariteten er uviss. I metode 1 og 2 får uvisse setningar sannsynsverdi på omlag 0.5 som vert teken med i snittet.

Oppsummering av resultata er gitt i tabell 6.2. *Precision*, *recall* og *accuracy* er som definert i seksjon 3.8. Av tabellen ser vi at det er metode 2 som gir best resultat, med *precision* på 0.952 og *recall* på 0.8. Framfor metode 1 er dette ein vesentleg auke i *precision* utan reduksjon i *recall*. Metode 3 har også same *recall*, men har vesentleg dårlegare *precision*. Metode 4, som ignorerer artikkelteksten og ser på tittel og ingress

		Metode 1		Metode 2		Metode 3		Metode 4		
Artnr.	Ingress	$P(pos)$	$P(neg)$	$P(pos)$	$P(neg)$	$P(pos)$	$P(neg)$	$P(pos)$	$P(neg)$	Klasse
1	Nei	0.61055	0.38945	0.65297	0.34703	0.99962	0.00038	0.48180	0.51820	P
2	Ja	0.43775	0.56225	0.43479	0.56521	0.00084	0.99916	0.15574	0.84426	P
3	Ja	0.80224	0.19776	0.79153	0.20847	0.11925	0.88075	0.99976	0.00024	P
4	Ja	0.80359	0.19641	0.78779	0.21221	0.99999	0.00001	0.99854	0.00146	P
5	Ja	0.53938	0.46062	0.57785	0.42215	1.00000	0.00000	0.48180	0.51820	P
6	Ja	0.63384	0.36616	0.65239	0.34761	0.89305	0.10695	0.99356	0.00644	P
7	Ja	0.62645	0.37355	0.62452	0.37548	0.78823	0.21177	0.53781	0.46219	P
8	Ja	0.64756	0.35244	0.65138	0.34862	1.00000	0.00000	0.86820	0.13180	P
9	Nei	0.57163	0.42837	0.61142	0.38858	0.99999	0.00001	0.48180	0.51820	P
10	Ja	0.40928	0.59072	0.44086	0.55914	0.99997	0.00003	0.48180	0.51820	P
11	Ja	0.69945	0.30055	0.73989	0.26011	0.99997	0.00003	0.93451	0.06549	P
12	Ja	0.37372	0.62628	0.36799	0.63201	0.02232	0.97768	0.29123	0.70877	P
13	Ja	0.16704	0.83296	0.17830	0.82170	0.00265	0.99735	0.00319	0.99681	P
14	Ja	0.81908	0.18092	0.82883	0.17117	1.00000	0.00000	0.99929	0.00071	P
15	Ja	0.57845	0.42155	0.57575	0.42425	0.95018	0.04982	0.87935	0.12065	P
16	Nei	0.69235	0.30765	0.67500	0.32500	0.77905	0.22095	0.86820	0.13180	P
17	Ja	0.71389	0.28611	0.73465	0.26535	0.04443	0.95557	0.86820	0.13180	P
18	Ja	0.67937	0.32063	0.82465	0.17535	0.99904	0.00096	0.80026	0.19974	P
19	Ja	0.41739	0.58261	0.43208	0.56792	0.99839	0.00161	0.10848	0.89152	P
20	Ja	0.51940	0.48060	0.51448	0.48552	1.00000	0.00000	0.39905	0.60095	P
21	Nei	0.69534	0.30466	0.66788	0.33212	0.99988	0.00012	0.85396	0.14604	P
22	Ja	0.56035	0.43965	0.64142	0.35858	0.99996	0.00004	0.96065	0.03935	P
23	Ja	0.64326	0.35674	0.64277	0.35723	0.95378	0.04622	0.82413	0.17587	P
24	Ja	0.71353	0.28647	0.76077	0.23923	0.99999	0.00001	0.99614	0.00386	P
25	Nei	0.53214	0.46786	0.54232	0.45768	0.97076	0.02924	0.48180	0.51820	P
26	Ja	0.30292	0.69708	0.34888	0.65112	0.99997	0.00003	0.28833	0.71167	N
27	Ja	0.50642	0.49358	0.48180	0.51820	0.99070	0.00930	0.48180	0.51820	N
28	Ja	0.29033	0.70967	0.27588	0.72412	0.37301	0.62699	0.28718	0.71282	N
29	Ja	0.27070	0.72930	0.34936	0.65064	0.00000	1.00000	0.12879	0.87121	N
30	Ja	0.35571	0.64429	0.35852	0.64148	0.99988	0.00012	0.08570	0.91430	N
31	Ja	0.28173	0.71827	0.28457	0.71543	0.00117	0.99883	0.00579	0.99421	N
32	Ja	0.28956	0.71044	0.33683	0.66317	0.50169	0.49831	0.00367	0.99633	N
33	Nei	0.16008	0.83992	0.14118	0.85882	0.00019	0.99981	0.02538	0.97462	N
34	Ja	0.21412	0.78588	0.24173	0.75827	0.00051	0.99949	0.08765	0.91235	N
35	Ja	0.19891	0.80109	0.19524	0.80476	0.00958	0.99042	0.03260	0.96740	N
36	Ja	0.40873	0.59127	0.40237	0.59763	0.70082	0.29918	0.25697	0.74303	N
37	Ja	0.45083	0.54917	0.45422	0.54578	0.00000	1.00000	0.48180	0.51820	N
38	Ja	0.45924	0.54076	0.44771	0.55229	1.00000	0.00000	0.16304	0.83696	N
39	Nei	0.48161	0.51839	0.48180	0.51820	0.31485	0.68515	0.48180	0.51820	N
40	Ja	0.33895	0.66105	0.30793	0.69207	0.05263	0.94737	0.15574	0.84426	N
41	Ja	0.35455	0.64545	0.35217	0.64783	0.00003	0.99997	0.26791	0.73209	N
42	Ja	0.29311	0.70689	0.41056	0.58944	0.04572	0.95428	0.34220	0.65780	N
43	Nei	0.42403	0.57597	0.48180	0.51820	0.00856	0.99144	0.48180	0.51820	N
44	Ja	0.22014	0.77986	0.22987	0.77013	0.93877	0.06123	0.01408	0.98592	N
45	Nei	0.26812	0.73188	0.15253	0.84747	0.99917	0.00083	0.12442	0.87558	N
46	Ja	0.29220	0.70780	0.35571	0.64429	0.99998	0.00002	0.02538	0.97462	N
47	Ja	0.36508	0.63492	0.35845	0.64155	0.14152	0.85848	0.11641	0.88359	N
48	Ja	0.50546	0.49454	0.49247	0.50753	0.99974	0.00026	0.88170	0.11830	N
49	Ja	0.53515	0.46485	0.61730	0.38270	0.00271	0.99729	0.68505	0.31495	N
50	Ja	0.46040	0.53960	0.46824	0.53176	0.71141	0.28859	0.15360	0.84640	N

Tabell 6.1: Klassifisering av artiklar

som eit dokument gjer det noko betre enn metode 3. Ver merksam på at ikkje alle artiklar har ingress, og at det er eit avgrensa utval med titlar i treningssettet.

	TP	TN	FP	FN	*Precision*	*Recall*	*Accuracy*
Metode 1	20	22	3	5	0.869	0.8	0.84
Metode 2	20	24	1	5	0.952	0.8	0.88
Metode 3	20	14	11	5	0.645	0.8	0.68
Metode 4	15	23	2	10	0.882	0.6	0.76

Tabell 6.2: Samanlikning av klassifikatormetodane

Der TP=*True Positive*, TN=*True Negative*, FP=*False Positive*, FN=*False Negative*, metode 1=snitt, metode 2=median, metode 3=dokument, metode 4=tittel og ingress som dokument

Om ein ser på artiklane der metode 2 klassifiserer feil er der **1** falsk positiv og **5** falske negativ.

Artikkelen som er falsk positiv er nr. 49 i tabell 6.1. Denne er feil klassifisert av metode 1 og 2, men korrekt klassifisert av metode 3. Ved nærmare undersøkelse av denne ser ein at ord som *krise*, *negativ* ikkje er attkjent som spesielt negative av klassifikatoren, medan ordet *ser* er attkjent som eit positivt ord. Dette fører til ei stor overvekt av setningar som er klassifisert som positive, og fører til at metodane som baserer seg på i setningane gjer feil. Ordet *fall* er nok til at klassifikatoren som er på heile dokument klassifiserer artikkel nr. 49 som negativ. Eit større treningssett vil i dette tilfelle mest sannsynleg føre til at klassifikatoren kjenner att fleire av dei negative orda og gi rett resultat.

Artiklane som er klassifisert falsk negativ er 2, 10, 12, 13 og 19. Grunnane til at desse artiklane er feil klassifisert er samansett. For artikkel 2 er det generelt positiv omtale, der delar av teksten skildrar moglege negative ting som kan skje. Dei negative delane av teksten vert til dels avfeid. Samstundes vert dei positive delane av teksten ikkje attkjent, på grunn av ord som ikkje finnast i treningsettet. Mykje av det same gjeld for resten av artiklane som er feil klassifisert – systemet kjenner ikkje tydinga til ei rekkje ord, og artiklane omtalar ulike sider av saka positivt og negativt. Verdt å merke seg er at artiklane som er feil klassifisert av metode 2, òg er feil klassifisert av dei andre metodane, bortsett frå artikkel 10 og 19 som er rett klassifisert av metode 3.

6.3.1 Oppsummering eksperiment 2

I dette eksperimentet har det blitt vist at dokument kan klassifiserast ved å nytte sentimentverdiar av setningane dokumentet består av. Det er nokre problem med å klassifisere nokre typar dokument, og der er klart at veikskapane funne i eksperiment 1 er også tatt med vidare.

Kapittel 7

Diskusjon og konklusjon

Gjennom denne oppgåva har det blitt presentert ein arkitektur og modell for klassifisering av nyhendeartiklar. Det har blitt laga ein prototype, som har blitt brukt til å vurdere sterke og svake sider ved modellen.

Å fastsetje sentimentverdi av einskilde setningar kan gjerast ganske godt ved å bruke metodane i skissert i kapittelet om arkitektur og modell. Gjennom eksperiment 1 har det blitt avdekka veikskapar ved språkmodellen, der nokre av dei er særskild fordi den er nytta på norsk språk. Det kan generelt visast tilbake til desse veikskapane på setningar som systemet klassifiserer feil. Med grunnlag i desse funna er det i kapittel 8 kome med forslag til utbetringar som kan føre til at systemet klassifiserer også desse setningane rett.

Systemet kan i stor grad også finne polariteten til heile dokument. Ulike metodar har blitt prøvd ut, for å finne kva som best høver seg for å klassifisere heile dokument. Metodar som nyttar sentimentverdiar av setningane har blitt sett opp mot metodar som nytter klassikator på heile dokument. Metodar som nyttar sentimentverdien av setningane gjer det i stor grad betre enn metodar som ser på dokumentet som *bag-of-words*, noko som var det vi ville undersøkje i høve problemformuleringa.

Det skal her nemnast at ved bruk av *bag-of-words*-modellen, for heile dokument kan vere ein fordel å nytte anna ein bolske *features* i *feature vector*-representasjonen. Å til dømes bruke vekting av termane kunne betra resultatet til klassifikatoren som nytta *bag-of-words*-modellen på heile dokumentet [14].

Gjennom eksperiment 2 var det klart at veikskapane funne i eksperiment 1 var med vidare. Dette førte til ein del av feilklassifiseringane. På dokumentnivå kan det vere vanskeleg å finne samla polaritet for ein artikkel som drøftar positive og negative sider om ei sak, eller omhandlar fleire saker. Å berre ha to klassar på dokumentnivå er noko grovt.

Å finne sentimentverdi til ei setning er vesentleg lettare enn å finne sentimentverdi til eit heilt dokument. I høve problemformuleringa, er det vist at å basere dokumentet sin sentimentverdi på setningane sine verdiar har visse fordelar. Systemet som er føreslått kan kjenne att ei rekkje positive og negative dokument og setningar.

Kapittel 8

Vidare arbeid

Gjennom eksperiment 1 og 2 vart det avdekka nokre veikskapar som svekte systemet si evne til å klassifisere visse setningar og dokument. Der finnast ei rekkje tilfelle der setningane, og dokumentet ikkje kan klassifiserast fordi systemet ikkje kjenner orda i teksten. Her bør det i første omgang nyttast eit større treningssett for å lære opp systemet til å kjenne att også desse orda.

På norsk er samskriving i utstrekt bruk. Som vist i eksperiment 1, er dette noko som også fører til at systemet ikkje kjenner tydinga av ord, sjølv om orda det samskrivne ordet er samansett av kan ha kjent polaritet. Det hadde såleis vore interessant å kunne dele opp slike ord og finne ein sentimentverdi for dei.

Døme 8.1. *Knallbra* er eit ord som kan vere brukt i forbinding med skildring av ei hending, resultat eller liknande. Både ordet *knall* og *bra* har i seg sjølv positive tydingar, men for klassifikatoren er det samansette ordet ukjent. Ved å bruke ei ordliste til å dele opp det ukjende ordet kan systemet finne ei tyding for det samskrivne ordet.

Klassifikatoren bommar på ting som ikkje vert teke omsyn til i språkmodellen, dvs. på setningar som inneheld ord som modifiserer sentimentet til andre ord i setninga. Bruk av bigram evt. trigram (av ord) kan fange opp mange av desse tilfella.

Døme 8.2. I klassifikatoren er ordet «ser» lagt inn som eit positivt i ord, sannsynlegvis i samanhengar som «det ser lyst ut», men ordet kan òg finnast i negative samanhengar. Ved bruk av bigram vil *features* vere <'ser', 'lyst'> og <'ser', 'mørkt'> som ikkje har fleire

tydingar.

Det er stor skilnad mellom språk brukt i tittelen til ein artikkel og resten av artikkel. Med grunnlag i dette kan det vere tenleg å bruke trene opp ein eigen klassifikator til bruk på titlar.

Sentimentverdiane i systemet er i prinsippet sannsynet for at setninga er positiv, og sannsynet for at setninga er negativ. Desse seier ikkje noko om i kva grad den er positiv eller negativ, og gir berre uttrykk for uvissa i klassifiseringa. Det kunne derfor vore interessant å knytte polariteten av setninga opp mot ord som «håper», «tror», «kanskje» og liknande, og såleis finne ulike grader av positivitet og negativitet.

Det kan også undersøkjast om norsk ordvev [23] kan nyttast i systemet. Bruk av Wordnet [19] for tekstklassifisering er synt å betre resultata i visse høve [24]. Såleis vil ord som tyder det same bli slått samen i samband med klassifiseringa, slik som forklart i seksjonen om *feature extraction* (3.5.1).

Resultata syner at klassifisering av polariteten til dokument i stor grad kan gjerast maskinelt. Ved å gjere tilpassingar og utbetringar av modellen og systemet, slik som skissert over, er det venta at eit slikt system større grad kan klare å klassifisere artiklane rett. Neste steg med tanke på motivasjonen til oppgåva er å sjå nærmare på samanhengen mellom finansmarknaden og sentimentanalyse av nyhende. Saman med eit system som identifiserer kva firma teksten handlar om kan ein sjå på utviklinga av ein viss aksjekurs sett opp mot omtalen til firmaet.

Vona er å finne ein samanheng tilsvarande likning 8.1. Der $k_0 S_x(t)$ er av vesentleg storleik.

$$U_x(t) = k_0 S_x(t) + k_1 M(t) + \ldots \tag{8.1}$$

der U_x er aksjekursen for firma $_x$, S_x er sentimentverdi, M er marknadsutviklinga, i tillegg til andre faktorar kursen kan vere avhengig av, til dømes sentimentverdi til andre firma i same bransje.

Litteratur

[1] Andranik Tumasjan, Timm O Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the fourth international AAAI conference on weblogs and social media*, pages 178–185, 2010.

[2] Peter Ager Hafez. How news events impact market sentiment. *Professor David J. Hand, Professor of Statistics, Imperial College, London; Chief Scientific Advisor, Winton Capital Management; and President, Royal Statistical Society*, page 129, 2011.

[3] Leela Mitra and Gautam Mitra. Applications of news analytics in finance: A review. *The Handbook of News Analytics in Finance*, pages 1–36, 2011.

[4] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[5] Kiyoshi Izumi, Hiroki Matsui, and Yutaka Matsuo. Integration of artificial market simulation and text mining for market analysis. In *Advances in Hybrid Information Technology*, pages 404–413. Springer, 2007.

[6] Leela Mitra, Gautam Mitra, and Dan Dibartolomeo¶. Equity portfolio risk estimation using market information and sentiment. *Quantitative Finance*, 9(8): 887–895, 2009.

[7] Marc-André Mittermayer and Gerhard Knolmayer. *Text mining systems for market response to news: A survey*. Institut für Wirtschaftsinformatik der Universität Bern, 2006.

[8] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[9] Gautam Mitra and Leela Mitra. *The handbook of news analytics in finance*, volume 596. Wiley, 2011.

[10] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. In *Proceedings of LREC*, volume 10, 2010.

[11] Paul C Tetlock, MAYTAL SAAR-TSECHANSKY, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

[12] Peter Hafez and Junqiang Xie. Factoring sentiment risk into quant models. *Available at SSRN*, 2012.

[13] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[14] Sanjiv Das. News analytics: Framework, techniques and metrics. *SCU Leavey School of Business Research Paper*, (11-08), 2010.

[15] David D Palmer. *Tokenisation and sentence segmentation*. Marcel Dekker, Inc., New York, USA, 2000.

[16] Julie B Lovins. *Development of a stemming algorithm*. MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[17] Snowball. Norwegian stemming algorithm. `http://snowball.tartarus.org/algorithms/norwegian/stemmer.html`.

[18] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

[19] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[20] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

[21] David Louis Olson and Dursun Delen. *Advanced data mining techniques*. Springer, 2008.

[22] Steven Bird. *Natural language processing with Python*. O'Reilly, Beijing Cambridge Mass, 2009. ISBN 978-0-596-51649-9.

[23] Nasjonalbiblioteket. Leksikalske ressurar. `http://www.nb.no/Tilbud/Forske/Spraakbanken/Tilgjengelege-ressursar/Leksikalske-ressursar`.

[24] Sam Scott and Stan Matwin. Text classification using wordnet hypernyms. In *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 38–44, 1998.

Vedlegg

1 Døme på artikkel

ID 1

Tittel Statoil finner olje nær Gullfaks Sør [DN.no]

Ingress

Tekst Statoil har gjort et oljefunn på anslagsvis 0,3-1,1 millioner standard kubikkmeter utvinnbare oljeekvivalenter i undersøkelsesbrønn 33/12-9 S sørvest for Gullfaks Sør i Nordsjøen i utvinningstillatelse 152. Hensikten med brønnen er å påvise petroleum i midtre jura reservoarbergarter. Brønnen påtraff lett olje i en 80 meter brutto kolonne i øvre del av Brentgruppen, i bergarter med god reservoarkvalitet. Funnet er planlagt knyttet opp til eksisterende infrastruktur i Gullfaksområdet. Brønnen ble ikke formasjonstestet, men det er utført datainnsamling og prøvetaking for å fastslå petroleumssystem og olje/vannkontakter. Brønn 33/12-9 S er den fjerde undersøkelsesbrønnen i utvinningstillatelse 152. Statoil er operatør i lisensen (PL 152) med 70 prosent eierandel. Petoro eier de øvrige 30 prosentene.