NTNU
Det skapende universitet

Automatisk organisering av domenebasert læringsmateriell

Anny Marit Svendsen

Master i informatikk
Oppgaven levert: Juni 2006
Hovedveileder: Arvid Holme, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Sammendrag

Målet med denne oppgaven er å se om en automatisk kan organisere kunnskapsobjekter tilrettelagt for et tredimensjonalt navigerbart konseptkart. Forutsetningen for å gjøre dette mulig er at kunnskapen er fremstilt i enheter, kalt læringsobjekter. Et læringsobjekt består av en eller flere kunnskapsobjekter knyttet sammen med ett eller flere læringsmål, og ulike læringsaktiviteter. Et kunnskapsobjekt er bygd opp av en eller flere ressurser. Disse ressursene kan være tekst, lyd, bilde, video osv.

En ser for seg at kunnskapsobjektene befinner seg i samlinger, repositories. Kunnskapsobjektene skal kunne utvikles av både faglig veileder og lærende, men hovedsakelig tenker en at de skal produseres av profesjonelle aktører. Her er det snakk om en produksjonslinje. Den profesjonelle aktøren står for produksjon og presentasjon av objektene, mens faglig veileder sørger for innhold og pedagogisk tilrettelegging.

Til hvert kunnskapsobjekt er det knyttet en tekstlig beskrivelse som danner grunnlag for vektorisering. Det er denne vektoriseringen som gjør det mulig å indeksere kunnskapsobjektene, samt sammenligne dem. Sammenligningen av de ulike objektene danner grunnlag for klyngeanalyse. Klyngeanalysen vil organisere like objekter i grupper slik at hver gruppe deler felles egenskaper, dvs. har en stor grad av likhet. Med denne grupperingen kan man representere objektene visuelt, for eksempel i en tredimensjonal graf.

Som en del av denne oppgaven er det utviklet to prototyper for klyngeanalyse. I den ene benyttes det K-means-algoritme, mens i den andre benyttes en agglomerativ hierarkisk algoritme. Dataen fra disse analysene er tenkt brukt i en tredimensjonal representasjon. Prototypen for K-means har en todimensjonal framstilling slik at man får et visst innblikk i hva en visuell framstilling av objektene innebærer.

En vil blant annet se av testene at K-means med brukervalgte initielle klyngerepresentanter gir det beste resultatet for klyngeanalysen, og at de tekstlige beskrivelsene av kunnskapsobjektene bør være av størrelse som en vanlig A4-side.

Innhold

Elektroniske vedlegg (vedlegg_mastergrad.zip):

Dokumentsamling	– tekstfilene brukt til testing
Javadoc	– dokumentasjon til kildekode
kildekode	– kildekode til implementasjon
hierarkisk_6avhver.txt	– Fullstendig resultat av kjøring med hierarkisk klyngeanalyse
Snarvei til Hierarkisk.bat	– startfil for hierarkisk klyngeanalyse med ekstra minne til java
Snarvei til kmeans.bat	– startfil for k-means klyngeanalyse med ekstra minne til java

Figurliste

1 Introduksjon

1.1 Problembeskrivelse

I dagens læringssystemer satses det mye på administrasjon av nettbasert læring, men lite på produksjon og organisering og presentasjon av læremateriellet.

Jeg vil i denne oppgaven fokusere på nettbasert læring sett fra den lærendes ståsted, og ut fra en konstruktivistisk modell skal en la de lærende selv bygge opp sin egen kunnskapsbase. Til dette foreligger det domenebaserte læringsmateriell i form av kunnskapsobjekter. Ved bruk av diverse klyngeanalysemetoder skal jeg forsøke å organisere kunnskapsobjektene tilrettelagt for et navigerbart konseptkart. For at dette skal være mulig må kunnskapen splittes opp i form av lærings- og kunnskapsobjekter.

Jeg ser for meg repositories, dvs. samlinger av kunnskapsobjekt. Disse samlingene skal både faglig veileder kunne søke i for å utvikle eget læringsmateriell, og studenten for å løse oppgaver. Et av hovedproblemene som dukker opp med en slik samling er da: Hvordan lagre og gjenfinne kunnskapsobjekter?

1.2 Mål

Målet med denne oppgaven er først og fremst å gi en innføring i den teorien som ligger bak vår objektorienterte læringsmodell. I den sammenheng er det naturlig også å gi en innføring i de datatekniske metoder som benyttes, da hovedsaklig informasjonsgjenfinning og klyngeanalyse.

For å illustrere bruken av modellen skal det utvikles en metode for klyngeanalyse basert på K-means. Ved hjelp av denne håper jeg å kunne vise at det er mulig å samle kunnskapsobjekter i naturlige klynger. Prototypen bør også kunne brukes til søking etter konkrete kunnskapsobjekter.

1.3 Oppgavens oppbygging

Denne oppgaven tar først for seg en grunnleggende introduksjon i læring med tanke på rollen motivasjon har, samt to modeller for læring, behaviorisme og konstruktivisme, og en oversikt over ulike læringsstrategier. Så gir oppgaven en innføring i nettbasert undervisning/læring. Her blir det sett på de ulike aktørene i nettbasert undervisningssammenheng. Etter dette er gjort, følger et kapittel om læringsobjekt og kunnskapsobjekt. Dette kapitlet redegjør for vår gruppes definisjon av disse objektene, samt beskrivelse av dem i form av en metafor kalt togmodellen. Med vår gruppe menes alle mastergradsstudentene som har amanuensis Arvid Holme som veileder.

Videre blir det sett på informasjonsgjenfinning. Informasjonsgjenfinningsdelen tar for seg preprosessering av dokumenter, informasjonsmodeller som kan benyttes, vekting av termer og bruk av metadata til dokumenter. Så gis det en innføring i emnet klyngeanalyse. Det finnes her to hovedinndelinger: Ikke-hierarkisk klynging og hierarkisk klynging. Innenfor de to hoveddelene er det mange teknikker som kan benyttes, men det er i denne oppgaven lagt hovedvekt på K-means innenfor ikke-hierarkisk klyning, og bottom-up (agglomerativ metode) innenfor hierarkisk klynging.

Da dette er beskrevet blir det gitt en oversikt over vår modell og hva vi i vår gruppe jobber mot. Her tar oppgaven for seg hvilken målsetting vi har, hvordan vi har tenkt at lærings- og kunnskapsobjektene skal lages og beskrives, hvordan vi har tenkt at et rammeverk for jobbing og presentasjon av disse objektene skal være, og hvordan man kan knytte bruk av fagtermer til et mer naturlig språk ved hjelp av latent semantisk indeksering.

Etter disse teoretiske kapitlene følger den praktiske delen. Først beskrives de to prototypene utviklet for denne oppgaven, en for K-means og en for agglomerativ hierarkisk analyse. Etter dette beskrives testing av ulike klyngeanalyseringen på dokumentsamlinger. Her er hovedfokuset bruk av K-means i ulike sammenhenger. Et utsnitt av en hierarkisk metode er med for å vise forskjellen på disse to.

Til slutt i oppgaven oppsummeres det hele og gir rom for konklusjon, samt veien videre med hensyn på hvordan man kan knytte dette sammen til et endelig nettbasert læringssystem.

2 Læring

Tradisjonelt har læring (særlig i skolesammenheng) foregått ved at en lærer ”lærer bort”, dvs. forteller om et emne, mens den lærende passivt hører på. Denne modellen bygger på en behavioristisk teori hvor man benytter ytre motivasjon for å oppnå det man ønsker. Selv om denne modellen ofte har blitt kritisert, særlig de siste tiårene, preger den fortsatt store deler av undervisningen som foregår i skolesammenheng. Man kan ikke si at det har foregått drastiske endringer fra tidlig av 1900-tallet da skoler ble vanlige og fram til i dag. Man har typisk hatt en ”forteller og tilhører”-situasjon, noe man fortsatt i stor grad har.

Noe av grunnen til at man i de siste tiårene har kritisert den behavioristiske modellen og i stedet ønsker å benytte konstruktivistiske metoder, er kanskje nettopp på grunn av den teknologiske utviklingen som har blitt gjort, og stadig åpner for nye muligheter for en mer aktiv læringssituasjon.

Det viser seg at motivasjon har mye å si for læring med tanke på hvordan man lærer ting og dermed også hvor godt man lærer ting. De ulike modellene behaviorisme og konstruktivisme bygger på de ulike formene for motivasjon; indre og ytre motivasjon. Dagens læringssituasjoner kan deles opp i ulike læringsstrategier. Et kurs som får studentene til å bruke en dypinnrettet læringsstrategi vil følge en konstruktivistisk modell, og det er denne modellen som er utgangspunktet for i vår gruppe og vårt arbeid med nettbasert læring.

Det blir i dette kapitlet først sett på forskjellen på indre og ytre motivasjon og hva motivasjon har å si for læring. Deretter blir det sett på to modeller for læring; behaviorisme og konstruktivisme, og til slutt en oversikt over ulike læringsstrategier.

2.1 Motivasjon

Det er ofte slik at barn oppfatter ting de ”må” gjøre som mye kjedeligere enn ting de ”får” gjøre, eller maten jeg ”må” spise er mye fælere enn maten jeg ”får” spise. Det er

eksperimenter fra psykologisk forskning som illustrerer dette. Man skiller mellom ytre motivasjon og indre motivasjon.

Ytre motivasjon er påvirkning utenfra, for eksempel leser man til en eksamen for å få god karakter, eller man gjør oppgaver for å få en form for belønning. Indre motivasjon derimot kommer fra det å kjenne den gode følelsen av autonomi og mestring. Det er altså det å mestre oppgaven i seg selv som er motivasjonen, ikke det å få en slags belønning utenfra. Ved indre motivasjon er man gjerne helt oppslukt av oppgaven man utfører, mens ved ytre motivasjon gjør man det ”for å få noe annet”. Hvis man er totalt hengiven til en aktivitet kan man snakke om flytopplevelser. ”Flytopplevelser kan sammenlignes med en slags rus som alle mennesker har behov for å være i med jevne mellomrom” (Øiestad 1993: 46). En aktivitet som er ytre motivert vil ofte innebære kortvarige flytopplevelser, mens en aktivitet som er indre motivert vil ofte ha lange flytopplevelser. Derfor kan man si at indre motivasjon gir rom for mer læring enn ytre motivasjon. Så hvis man greier å få barn til å lære ved at de har en indre motivasjon som driver dem, har man oppnådd mye. Det er dette som pedagogisk programvare og spill kan bidra til. Det gir nye muligheter for å lære ved å utforske ting selv, og barna får forhåpentligvis flytopplevelser som bidrar til bedre og mer læring. (Øiestad 1993).

Ikke bare hos barn er denne flytopplevelsen viktig for læring. Også hos ungdom og voksne vil indre motivasjon være en bedre læringsfaktor enn ytre motivasjon. På samme måte som at pedagogisk programvare og spill kan være en økt læringsfaktor hos barn, kan teknologien også bidra til nye metoder for læring hos ungdom og voksne. Man får muligheten til selv å være mer aktiv i læringsprosessen, i stedet for passiv læring (for eksempel det man kjenner som ”vanlige” forelesninger hvor en fagperson forteller om et emne og studenten er en passiv tilhører) som er en mye brukt læringsmetode, men som ofte er kritisert nettopp på grunn av at den er passiviserende. Man snakker her gjerne om læring sett fra et konstruktivistisk perspektiv. Disse teoriene, behaviorisme og konstruktivisme er nærmere beskrevet i seksjon 2.2 og 2.3.

2.2 Behaviorisme

Behaviorisme er studiet av ytre observerbar atferd, og den oppfatning at psykologien som helhet skal dreie seg kun om det. Denne teorien ble grunnlagt og lansert av John B. Watson (1878-1958) gjennom en forelesningsserie ved Columbia University i New York i vårsemesteret 1913. Behaviorisme var en protest mot mentalistisk psykologi som opererte mellom sjel og bevissthet. Watson ville ha en vitenskap om mer konkrete og synlige fakta. Han ville at psykologien skulle bli en objektiv vitenskap om atferd (både dyrs og menneskers) og en gren av naturvitenskapene. Ved at vitenskapen skulle bli mer objektiv, ønsket man å unngå begreper som sjel, bevissthet, mentale tilstander, sansing og persepsjon, følelser og emosjoner, vilje, forestillinger osv. I stedet skulle en bruke begreper som stimulus, respons, refleks, vaner og læring. Særlig ble stimulus og respons begrep som senere ble populære og behaviorismen omtales av og til som ”S-R-psykologi”. Stimuli er påvirkninger fra omgivelsene, og responser er organismens reaksjoner på påvirkningene. Man skulle finne lovmessige sammenhenger mellom stimuli og responser, slik at når en bestemt stimuli var gitt, kunne en predikere responsen. Likedan, når en respons ble observert, kunne en slutte seg til den aktuelle stimulus. Responser kunne være ulærte (reflekser) eller lærte (ved betinging). (Ilstad 2002).

Ved å benytte en behavioristisk teori fokuserer man altså som nevnt mer på det som er målbart og mulig å observere, i stedet for det som har med de mentale prosessene å gjøre. I en læringssituasjon vil da læring være det å få fram ønsket respons i forhold til hva som blir gitt som stimulus. Med andre ord; læreren ønsker å se et bestemt stimulus fra den lærende, i forhold til det som presenteres. Læring er da å forme den lærende slik at det alltid blir gitt riktig respons ut fra stimulus som blir presentert av læreren. For å oppnå dette kan man gjerne bruke ideen om belønning og straff. Ved belønning mener man at det gis positiv respons (som regel verbalt) når den lærende gir riktig svar, og ved straff mener man negativ respons (som regel verbalt) fra læreren. Dette er da med på å lose den lærende fram riktig vei. Denne metoden benytter da ytre motivasjon (beskrevet nærmere i seksjon 2.1) som læringsfaktor, i stedet for indre motivasjon som konstruktivisme i større grad gjør.

2.3 Konstruktivisme

Konstruktivisme sies ofte å stamme fra Giambattista Vico (1668-1744) og Immanuel Kant (1724-1804). Denne teorien har som grunnlag at det ikke finnes noen bestemt måte å erkjenne virkeligheten, men at mennesker aktivt skaper deres viten om verden og hverandre. Erkjennelsen er formidlet via konstruksjoner eller redskaper og deres fortolkning. Konstruktivismen innenfor psykologien er også knyttet til Jean Piaget (1896-1980) og Lev Semenovich Vygotskys (1896-1934) navn og til studiet av utviklingsprosessen av den menneskelige oppfatning og erkjennelse.

Utviklingsprosessen handler om at hvert enkelt individ som konstruerer sin egen realitet (verden). Dette er subjektivt og baserer seg på erfaringer. Hver enkelt har sine egne forståelsesformer, begreper og redskaper som det trenger til dette. En voksen person er ikke ”gitt” fra naturens eller Guds hånd, men bygger seg selv opp. Særlig Vygotsky og senere Pierre Bourdieu (1930-2002) understreker de sosiale og samfunnsmessige rammer for konstruksjonen og konstruktøren. Hvert individ skaper seg ikke en konstruksjon som blir ferdig, men dette er en evigvarende prosess, med andre ord; konstruksjonen er noe dynamisk som utvikler seg over tid.

Konstruktivismens utgangspunkt er altså at vi mennesker ikke erkjenner, forstår og erfarer omverden og virkelighet som den er ”i seg selv”, men vi erkjenner og forstår den på forskjellige måter. Hvordan vi erkjenner og forstår den avhenger av hva vi deltar i, hvordan og når vi deltar, hvordan vi iakttar den og i hvilken situasjon og hvilken kulturell sammenheng. Konstruktivisme handler om at det vi sier er ”sant”, ”virkelig” og ”riktig” er formet og fortolket av mennesker, med andre ord konstruert. Konstruktivisme interesserer seg derfor ikke for hvordan ting er ”i seg selv”, men hvordan og hvorfor vi mennesker erfarer, forstår og beskriver dem på forskjellige måter. (Leksikon.org: Konstruktivisme).

Ved å benytte en konstruktivistisk teori innenfor læring er man altså opptatt av at den lærende selv er deltakende og med på å bygge opp ”sannheten”. Dette er i kontrast med behavioristisk teori. Som beskrevet i forrige seksjon er den lærende passiv og skal formes etter et bestemt mål i den behavioristiske modellen for læring. Etter den konstruktivistiske modellen derimot, er den lærende mer aktiv og bidrar selv til å bygge opp sin kunnskap.

2.4 Læringsstrategier

”Vi ser på læring som en endring i studentens oppfatning av omverdenen innen det område som er studert og kompetanse i å håndtere og løse problemer på feltet, analytisk evne, evne til å skille mellom sentrale og perifere spørsmål, evne til å bruke fagets redskaper på en hensiktsmessig måte. Det vil si læring er en kvalitativ endring i studentens forståelse, faglige, sosiale og/eller tekniske kompetanse på et felt. Dette står i motsetning til en oppfatning av læring som en kvantitativ endring, det vil si at kunnskap er en større mengde viten og evne til å huske eller gjengi detaljer. Vår utfordring blir å stimulere til meningsfylt læring.” (Rekkedal 1999).

Med utgangspunkt i pedagogisk forskningen har en forskergruppe ved Göteborgs Universitet lagt grunnlaget for utvikling av teorier knyttet til hvordan læring foregår. Disse teoriene står på mange måter i kontrast til tidligere psykologisk forskning og læringsstudier. Forskningen viser at studenter tilnærmer seg læring på ulike måter. Disse kan deles opp i dypinnrettet tilnærming, overfladisk tilnærming og strategisk tilnærming til læring. (Rekkedal 1999).

2.4.1 Dypinnrettet tilnærming til læring

Intensjonen med denne tilnærmingen er å forstå ideene selv. Man skaper sin egen kunnskap gjennom å relatere ideer til tidligere kunnskap og erfaring. Relateringen foregår ved at man søker etter mønstre og underliggende prinsipper. Her blir forklaringer og bakgrunnsmateriell kontrollert og egne konklusjoner blir trukket ved at forklaringer og argumenter blir vurdert grundig og kritisk. Den lærende er i denne tilnærmingen aktivt interessert i kursinnholdet.

2.4.2 Overflatisk tilnærming til læring

Intensjonen er her å tilfredsstille kurs- og opplæringskrav. Den lærende vil i denne tilnærmingsmetoden reprodusere kunnskap gjennom å studere uten å reflektere over hensikt eller strategi. Kursinnholdet blir behandlet som urelaterte kunnskapsbiter hvor man memorerer fakta og framgangsmåtene blir en rutine. Dette fører til at den lærende vil finne det vanskelig å finne mening i nye ideer som presenteres, og får lett følelsen av stort arbeidspress og studieangst.

2.4.3 Strategisk tilnærming til læring

I en strategisk tilnærming til læring er intensjonen å oppnå best mulig karakter. Den lærende vil her organisere læringen gjennom å legge stor innsats i studiearbeidet og finne fram til effektive studiebetingelser og materiell, noe som gjør at den lærende er særlig oppmerksom på vurderingskrav og kriterier og vil styre tiden og innsatsen effektivt. Her blir arbeidet tilpasset lærerens prioriteringer og preferanser.

Rekkedal sier i sitt foredrag at den store forskjellen mellom studenter som lykkes og studenter som mislykkes ligger i måten de tilnærmer seg og organiserer læringsinnholdet. Studenter som ser etter meningen og en egen organisering av framstillingen har en dypinnrettet tilnærming, mens studenter som ser på innholdet som en samling fakta som skal læres og huskes har en overflatisk tilnærming til læringsmateriellet. Videre sier han at undervisnings- og evalueringsopplegget fra kursets side vil påvirke studentenes læringsstrategi.

Et kurs som får studentene til å bruke en dypinnrettet læringsstrategi vil følge en konstruktivistisk modell, og det er dette som nevnt tidligere, vi i vår gruppe jobber mot.

3 Nettbasert undervisning/læring

I undervisningssammenheng har man etter hvert blitt opptatt av hvordan man kan utnytte teknologien for å gi et bedre læringstilbud. En av grunnene til dette er at tilgangen til Internett har blitt meget utbredt. I dag har de aller fleste utdanningsinstitusjoner, skoler, bedrifter og etter hvert de fleste hjem tilgang til Internett. Dette har også ført til at kompetansen på bruk av Internett har økt, og det har for mange blitt en del av hverdagen. En annen årsak er at bruk av datamaskiner gir gode muligheter for multimedia. Man kan presentere lærestoff på helt nye måter ved bruk av lyd, bilder, video og animasjoner. Kombinerer man disse mulighetene med bruk av Internett kan man ha en interaktiv læringssituasjon hvor kunnskapen blir formidlet på nye måter.

Aktørene i nettbasert undervisning/læring er opptatt av hvordan man kan øke distribusjonen og tilgangen til læringstilbud. Det satses store summer på selve læringsprosessen gjennom bruk av nettbaserte læringsplattformer.

Selv om disse tilbudene har økt i omfang de siste årene, er det mange kritikere som hevder at det eneste man har oppnådd er å flytte auditoriet til Internett. Mange mener altså at man ikke har oppnådd noe nytt, men bare overfører de gamle læringsmetodene til en elektronisk arena. De hevder at læringsplattformene er sterke på administrative oppgaver og svake i pedagogisk sammenheng. I dagens systemer er det stort sett faglærer som må lage og organisere lærestoffet, produsere en elektronisk presentasjon av stoffet (tilpassert et nettbasert system) samt legge det tilgjengelig på nett. Ettersom dette blir gjort individuelt og sannsynligvis forskjellig i hvert kurs eller på hver skole, har læringsmateriellet liten grad av standardisering og er derfor lite egnet for gjenbruk. Det finnes heller ikke noen effektive metoder for å søke i eksisterende læringsmateriell. Organiseringen blir også stort sett utført manuelt og det finnes ikke mange modeller for automatisk organisering av læringsmateriell. (Holme 2006).

En årsak til at dagens systemer har blitt kritisert og sett på som en administrativ løsning og ikke en pedagogisk løsning, kan være at utvikling av interaktivt lærestoff krever at forfatteren har god kompetanse, god tid og gode verktøy til å produsere de interaktive bitene, noe forfatteren ikke alltid har. Om man skal se på hvordan man kan forbedre den nettbaserte undervisningen slik at det ikke blir som kritikerne hevder; gamle metoder overført til en

elektronisk arena, må man se på hvem som er aktørene i nettbaserte kurs i forhold til læringsbegrepet.

3.1 Aktører i nettbasert undervisningssammenheng

Man kan dele inn aktørene i nettbaserte kurs inn i tre hovedgrupper. Disse er den lærende, faglig veileder (lærer) og produsenten. Disse aktørene kan også betraktes som roller, da det kan variere hvilken rolle en person oppfyller til ulik tid. Samme person kan også ha flere roller. (Larsen 2006).

3.1.1 Den lærende

Den lærende har som mål å forbedre sin kompetanse og kunnskap innen et bestemt område og dermed skaffe seg det nødvendige lærestoffet. Den lærendes behov er å ha tilgang til oppdatert lærestoff, ha tilgang på lærestoff som det er behov for akkurat der og da (slippe å lese gjennom en hel lærebok for eksempel), lære uavhengig av tid og sted, selv kunne finne fram til lærestoffet som er nødvendig for å nå læringsmålet, lage sin egen læringssti ut fra egen kompetanse, forvalte lærestoffet og utveksle lærestoff med andre.

3.1.2 Faglig veileder

Faglig veileder har ansvaret for hva den lærende skal lære. Veilederen har ansvaret for hva lærestoffet skal omfatte og at det er i samsvar med en eventuell fagbeskrivelse eller at det gir en reell kompetanse. Videre har veilederen ansvar for å kvalitetssikre lærestoffet som blir videreført fra produsenten til den lærende og gi informasjon og sette krav til produsenten om hva som skal lages. Veilederens behov er å kunne oppdatere og revidere lærestoffet, finne igjen lærestoff slik at det kan brukes i nye læringssammenhenger, gjenbruke lærestoff og ressurser, kombinere ressurser til et lærestoff, være konsistent i utvikling av lærestoff, kunne tilpasse lærestoffet individuelt til den lærende og tilby lærestoffet i ulike former (tekst, bilde, video, animasjon osv.).

3.1.3 Produsenten

Produsenten har som oppgave å utvikle de interaktive ressursene. Produsenten vil ved hjelp av ulike dataverktøy stå for presentasjonen av lærestoffet. Dette vil kunne gjøres interaktivt om ønskelig. Produsentens behov er å motta eksakt informasjon om hva som skal presenteres og hvordan det er ønskelig at det skal presenteres (om faglig veileder har spesielle ønsker, ideer og tanker rundt presentasjonsbiten), produsere ressurser i formater som er tilgjengelig på alle systemer (særlig da med tanke på ulike operativsystemer, teksteditorer og nettlesere), produsere ressurser i formater og størrelser som gjør det relativt raskt å laste ned, gjenfinne ressurser, gjenbruke ressurser, omarbeide og revidere ressurser samt å ha en kostnads- og tidseffektiv produksjon.

3.1.4 Lærestoff i en produksjonslinje

Man ser altså her for seg en profesjonell utvikling av lærestoff. Man har en produksjonslinje. Det er ikke forventet at faglig veileder skal ha god kompetanse innenfor IKT (informasjons- og kommunikasjonsteknologi) slik at det kan lages presentasjoner av lærestoffet i form av bilder, animasjoner, videoer osv., men man ønsker at profesjonelle produsenter tar for seg presentasjonsdelen av lærestoffet. Faglig veileder skal stå for innholdet og den pedagogiske delen, mens produsenten står for presentasjonen. Slik blir det best utnyttelse av tid og ressurser. Det vil være misbruk av både tid å ressurser om man lar en person med meget lav kompetanse innenfor et emne ha ansvaret for akkurat det emnet.

En og samme person kan som nevnt tidligere også ha flere aktørroller. Faglig veileder kan for eksempel også ha rollen som produsent dersom det er ønskelig og tilstrekkelig kompetanse. Dersom lærestoffet skal organiseres som ren tekst, er det nok oftest læreren som står for produksjonen, men er det en mer avansert presentasjon som er ønskelig, som video, animasjoner og lignende kan produksjonen settes bort til produsenter som har den nødvendige kompetansen. Det kan også være veldig viktig med et tett samarbeid mellom produsenten og læreren slik at ikke innholdet og pedagogikken ”forsvinner” i presentasjonen.

Også den lærende kan produsere stoff som senere kan benyttes som lærestoff for andre lærende. Da vil den lærende også kunne ha rollen som produsent i tillegg til den lærende rollen.

Om man ser på behovene til de ulike aktørene i produksjonslinjen er det en del fellestrekk. Dette er behovet for å kunne gjenfinne, gjenbruke, revidere samt behovet for å kunne finne igjen og lage ressurser og lærestoff. Vi har derfor i vår gruppe og vårt arbeide med nettbasert læring, lagt vekt på disse momentene. Særlig var tanken om gjenbruk og gjenfinning viktig da vi kom fram til en måte å dele opp kunnskapen på.

4 Læringsobjekt og kunnskapsobjekt

I forbindelse med nettbasert læring, er det en viss enighet om at læringsobjekt er små enheter som er byggesteinene for et kurs. Det har vært mange ulike måter å definere et læringsobjekt på, og dette er et tema som er veldig omdiskutert for tiden.

I følge Wiley (Wiley 2000) kommer ideen om læringsobjekt opprinnelig fra objektorientert programmering. Dahl og Nygård utviklet på 60-tallet det objektorienterte programmeringsspråket Simula. Denne tankegangen tar utgangspunkt i at verden består av objekter. Disse objektene kan settes sammen på forskjellige måter og kan brukes om igjen i helt andre problemstillinger. Man ønsker altså objekter som er frittstående og som da gir mulighet for gjenbruk. Også i undervisning/læringssammenheng ønsket man å dele opp kunnskapen på en slik måte at man kunne sette den sammen i ulike sammenhenger på ulike måter.

Senere har det kommet andre definisjoner og metaforer for hva et læringsobjekt er. I artikkelen ”Lær av Lego” (REN 2002) sammenligner man læringsobjekter med Legoklosser. Klossene finnes i ulike størrelser og kan settes sammen på ulike måter. De kan så tas fra hverandre, en kloss kan erstatte en annen og de kan omgrupperes og settes sammen igjen på en helt ny måte. I tillegg er de så enkle at alle, til og med barn kan sette dem sammen.

Denne metaforen har blitt kritisert av Wiley. Han mener at det ikke er mulig å sette sammen læringsobjekt vilkårlig slik legoklosser kan og samtidig oppnå et bra læringsopplegg. Derfor sier han at det er nødvendig med en annen metafor som er mer sammenlignbar og kommer med atomteori som ny metafor. Ikke alle atomer kan kombineres med alle andre atomer, de kan bare settes sammen i bestemte strukturer og noe kunnskap trengs for å kunne sette de sammen.

The IEEE Learning Technology Standards Committee definerer læringsobjekt som ”... a learning object is defined as any entity, digital or non-digital, that may be used for learning, education or training.” Denne definisjonen er etter vårt syn veldig vid og har liten nytte. At en entitet som enten er digital eller ikke-digital og kan brukes i læring sier veldig lite. Denne definisjonen kan med andre ord omfatte alle entiteter i hele verden. Det er også svært lite

nyttig å ha en definisjon som også omfavner ikke-digitale entiteter, da det her er snakk bruk av datateknologi i forbindelse med læring. Wiley sin definisjon tar bare med digitale ressurser, men er likevel også veldig vid; ”any digital resource that can be reused to support learning.” (Wiley 2000).

Videre er det en rekke definisjoner på læringsobjekt;

”A learning object, narrowly defined, refers to a small, stand-alone unit of instruction that can be tagged with descriptors and stored in repositories for reuse in various instructional contexts.” (Hamel, C. J., & Ryan-Jones, D.).

“A Reusable Learning Object (RLO) is an element of or all of an instructional program that is delivered using technology. RLO’s can be lesson plans, case studies, quizzes, simulations, or interactions.” (Enclyclopedia of Learning Technology).

“The smallest component of an online course that describes and directs learning activity.” (UKeU, London).

Men alle disse definisjonene er relativt vide og konkretiserer lite. De sier ikke entydig hva et læringsobjekt er. Vi i vår gruppe mener at det er viktig å konkretisere mer, og har derfor lagd vår egen definisjon av læringsobjekt. Vi har også valgt å skille mellom kunnskapsobjekt og læringsobjekt. Det er derfor definert at et læringsobjekt består av en eller flere kunnskapsobjekt (tilknyttet læringsmål og læringsaktiviteter) som igjen består av en eller flere ressurser. Disse inndelingene blir nærmere beskrevet i underkapitlene som følger, før det hele beskrives som en metafor kalt togmodellen.

4.1 Ressurser

En ressurs i læringssammenheng er den minste byggesteinen i et læringsobjekt. Dette kan være en tekstfil, bildefil, lydfil, videofil, animasjon, spill eller annen data. Disse blir sett på som atomiske enheter som ikke kan brytes ned videre (fordi de da vil miste vesentlige deler av innholdet og betydning). Et viktig moment med ressurser er at de er gjenbrukbare, dvs. at

de kan benyttes i andre sammenhenger senere uten at man må forandre på dem. For å oppnå dette bør ressursene være kontekstuavhengige.

Det er ofte slik at for å forklare noe trenger man flere metoder. Man kan for eksempel ønske å forklare noe ved hjelp av både en tekst og et bilde. Det er ofte slik at en ressurs ikke er god nok til en entydig forklaring, og derfor har man behov for å sette sammen flere ressurser. Ressurser kan ut fra vår definisjon settes sammen til et kunnskapsobjekt, beskrevet under.

4.2 Kunnskapsobjekt

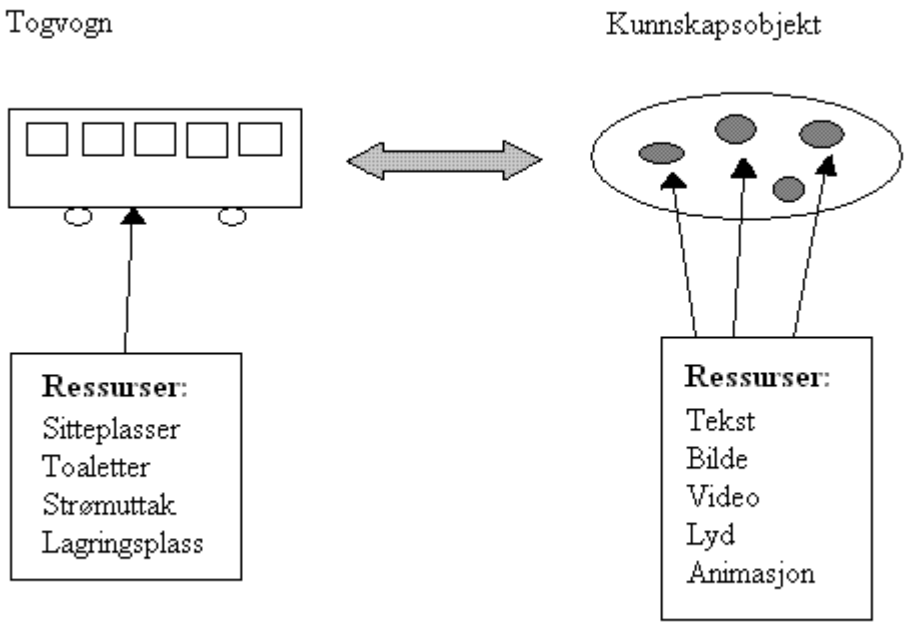
Et kunnskapsobjekt er bygd opp av en eller flere ressurser og skal alene eller sammen med andre kunnskapsobjekter beskrive et konsept. Kunnskapsobjektet skal i likhet med ressursene være gjenbrukbare og kontekstuavhengige. De skiller seg fra ressurser ved at de som oftest er mer selvstendige, dvs. de gir en forklaring på et emne, en oppgave eller lignende. Kunnskapsobjektet skal i prinsippet ikke knyttes til noe læringsmål. Et eller flere kunnskapsobjekt kan igjen settes i en kontekst og få læringsmål knyttet til seg, og dermed danne et læringsobjekt, beskrevet under.

4.3 Læringsobjekt

Et læringsobjekt er et eller flere kunnskapsobjekt kombinert til en større helhet, satt i en kontekst og med et eller flere læringsmål og eventuelle læringsaktiviteter. Læringsobjektene er da i motsetning til kunnskapsobjektene og ressursene, som er kontekstuavhengige, kontekstavhengige. Det er da ofte er læreren som lager slike læringsobjekter ved å velge ut kunnskapsobjekter fra en samling av relevante kunnskapsobjekter (repository, beskrevet i seksjon 7.2.4), og utvikle et læringsopplegg rundt dem. For at den lærende skal kunne oppnå det aktuelle læringsmålet, vil det gjerne være læringsaktiviteter og oppgaver knyttet til læringsobjektet. Disse er det da læreren som har hovedansvaret for.

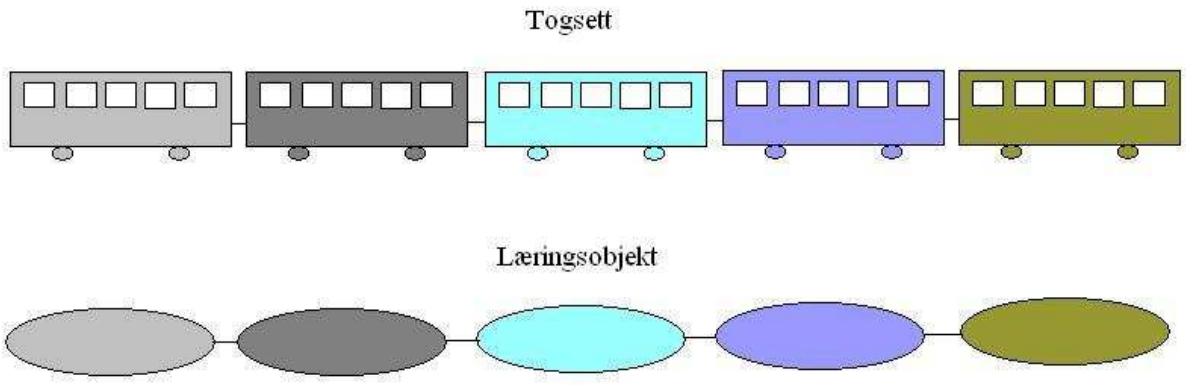
4.4 Togmodell

I vår gruppe har Jarle Larsen lagd en beskrivende metafor for vår måte å se på et læringsobjekt på. Denne metaforen kan forklares som følger. Basisenheten er tognvogn. En togvogn består av en eller flere ressurser som sitteplasser, toaletter, strømuttak, lagringsplass osv. Det finnes ulike typer togvogner som for eksempel tankvogn, passasjervogn og kjølevogn. Man sammenligner altså her en togvogn med et kunnskapsobjekt som består av en eller flere ressurser. En illustrasjon av dette er vist i Figur 1. Ressurser for et kunnskapsobjekt kan som nevnt tidligere være tekst, bilde, video, lyd, animasjon osv.



Figur 1: Togvogn som kunnskapsobjekt.

Flere togvogner, dvs. kunnskapsobjekt skal kunne settes sammen til et togsett. I denne sammenhengen vil et togsett være et læringsobjekt. Man kombinerer ulike kunnskapsobjekter og får en helhet, et læringsobjekt. Til læringsobjektet kan det også som nevnt tidligere knyttes et eller flere læringsmål og læringsaktiviteter. Denne sammenligningen mellom et togsett og et læringsobjekt er vist i Figur 2.

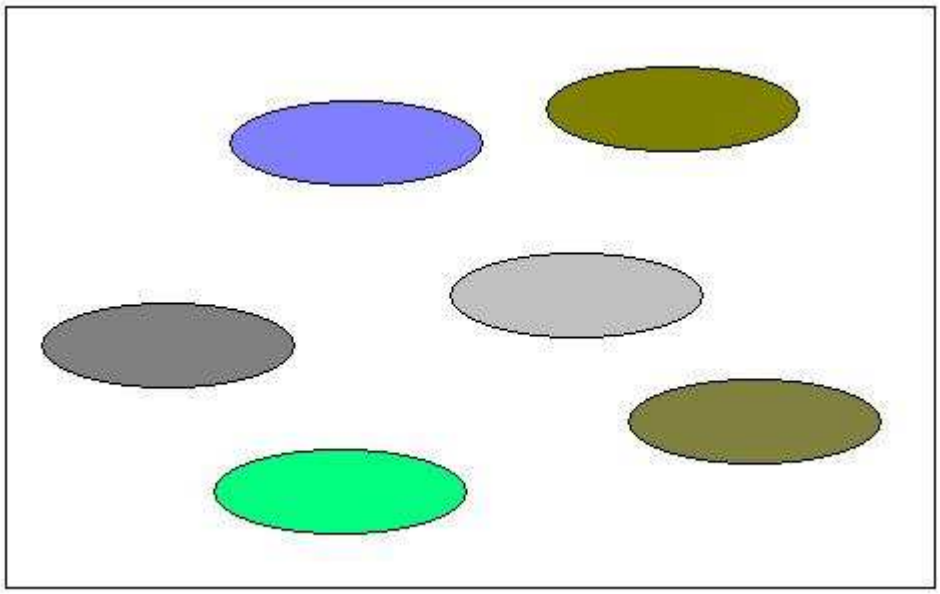


Figur 2: Togsett som læringsobjekt.

Det er også verdt å merke seg at grensesnittet mellom togvognene er det samme for alle togvogner uansett type. Det vil si at man enkelt kan fjerne, legge til, eller bytte ut togvogner og dermed få nye togsett som da vil være forskjellig fra det opprinnelige. På samme måte kan kunnskapsobjekter kombineres på ulike måter og danne ulike læringsobjekt. Målet er å ha læringsobjekter som lett kan tilpasses senere ved å bytte ut kunnskapsobjektene det er konstruert fra.

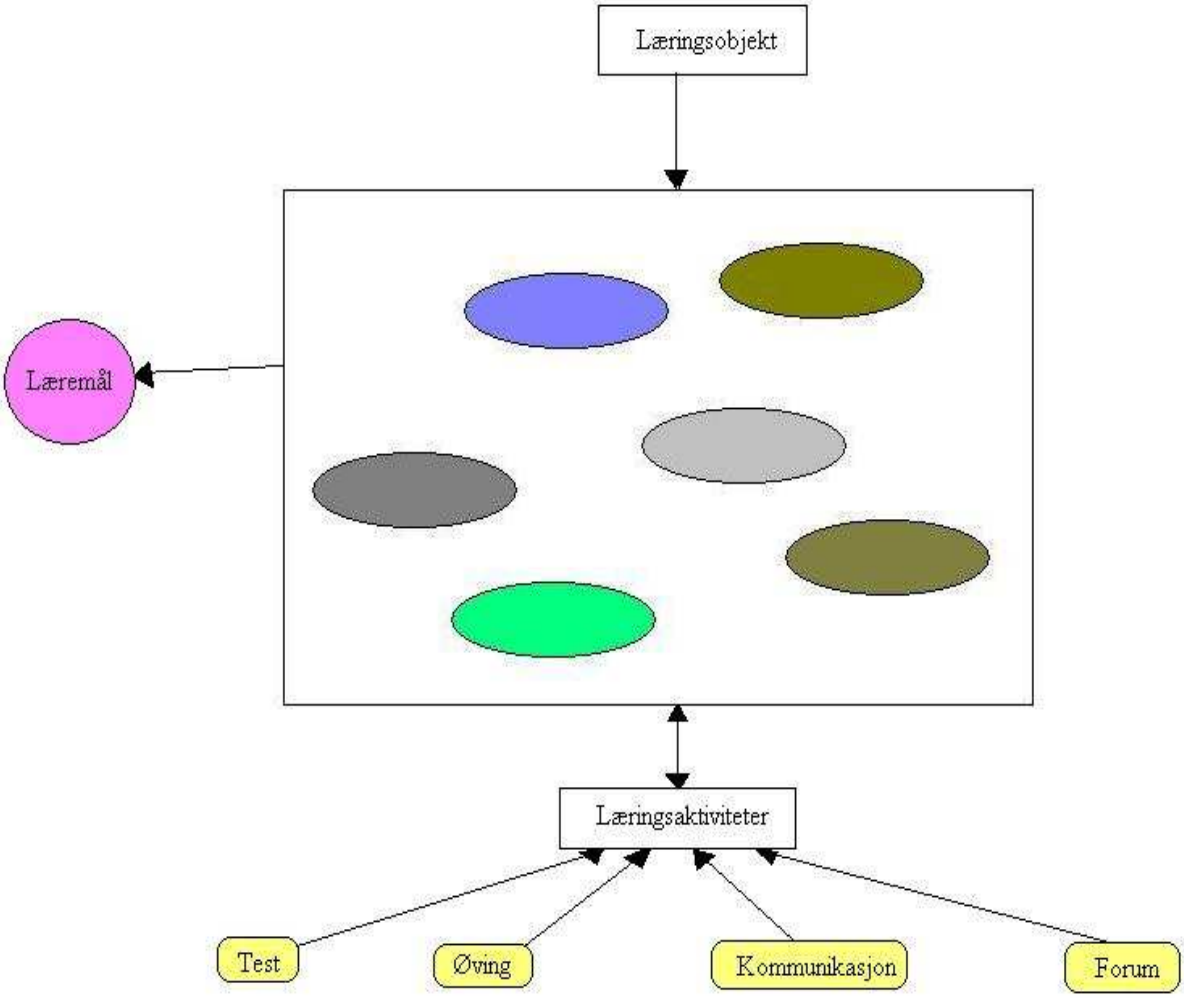
En lærer vil kanskje ofte ønske at kunnskapsobjektene har en bestemt rekkefølge for å få en viss progresjon av kunnskapen, men det kan tenkes tilfeller der dette ikke er nødvendig for å få en helhet. Man kan da gjøre det slik at rekkefølgen for kunnskapsobjektene ikke spiller noen rolle, dvs. de kan bli lest i hvilken som helst rekkefølge, men fortsatt gi mening. Kunnskapsobjektene må da være uavhengige av hverandre og ikke settes sammen til et togsett slik togmodellen tilsier. Dette er da opp til læreren eller den lærende hvordan det blir gjort. Særlig i en arbeidsmetode som problembasert læring (PBL)[1] ønsker man ofte ikke en bestemt rekkefølge satt opp på forhånd. I PBL er det den lærende som er mer aktiv og søker etter relevant informasjon, da etter vår modell kunnskapsobjekter. Da er det opp til de lærende selv å bygge opp rekkefølgen de lærer ting. Et slikt læringsobjekt med fri rekkefølge av kunnskapsobjektene er vist i Figur 3.

[1] ”Problembasert læring (PBL) har sin opprinnelse fra medisinsk utdanning. Den typen oppgaver som presenteres i PBL er vanligvis nært relatert til reelle situasjoner, eller bygger direkte på aktuelle hendelser. Relevansen til den typen situasjoner studentene vil møte i aktiv yrkesutøvelse, er tydelig.” (Problembasert læring).



Figur 3: Læringsobjekt bestående av uavhengige kunnskapsobjekt (fri rekkefølge).

Selv om læringsobjektene altså kan bestå av kunnskapsobjekter i en sekvensiell rekkefølge eller en fri rekkefølge, vil læringsobjektet alltid ha en kontekst, dvs. et læringsmiljø det befinner seg i. Denne konteksten er som nevnt læringsmål og læringsaktiviteter. Dette er illustrert i Figur 4.



Figur 4: Læringsobjekt i en kontekst.

(Larsen 2006).

5 Informasjonsgjenfinning

All jobbing med kunnskapsobjekter innebærer at objektene skal indekseres og organiseres. For at dette skal være mulig benyttes det metoder hentet fra informasjonsgjenfinning. Det blir først i dette kapitlet sett på preprosesseringsdelen, hvor det blir gitt en innføring i leksikalsk analyse, eliminering av stoppord, stemming, valg av indekstermer og bruk av thesaurus. Etter dette blir de mest brukte informasjonsgjenfinningsmodellene, boolsk modell, vektormodellen og probabilistisk modell gjennomgått. Til slutt ses det på ulike måter å vekte et dokument på, før bruk av metadata blir beskrevet.

5.1 Preprosessering av dokumenter

Preprosessering av et dokument er viktig før man skal bygge opp en indeks. Dette fordi dokumenter ofte inneholder termer og tegn som ikke egner seg som identifikatorer for dokumentet og som man derfor ikke ønsker i indeksen. Et annet argument for å gjennomgå en preprosessering er ønsket om å ha en relativt komprimert indeksfil slik at den tar mindre plass, samt gir bedre ytelsesevne.

Preprosesseringen kan deles inn i fem deler, og består av; leksikalsk analyse, eliminering av stoppord, stammeanalyse, valg av indekstermer og konstruksjon av termkategoriseringsstrukturer (thesaurus). (Baeza-Yates & Ribeiro-Neto 1999).

Det informasjonssystemet sitter igjen med etter en slik preprosessering er ofte kalt logical view (brukerutsnitt) av dokumentet. Denne representasjonen er illustrert i Figur 5. Her kan dokumentet først gjennomgå en strukturgjenkjenning[2]. Dette er ikke en nødvendig prosess men kan være gunstig slik at systemet for eksempel automatisk kan velge type stoppordliste som skal bli brukt (for å for eksempel få fjernet termer som er syntaksbundet til strukturen og ikke har noe og si om innholdet i dokumentet). Etter dette eventuelt er gjort er det leksikalsk analyse som er neste del. Denne delen er nødvendig for at man skal kunne delt opp

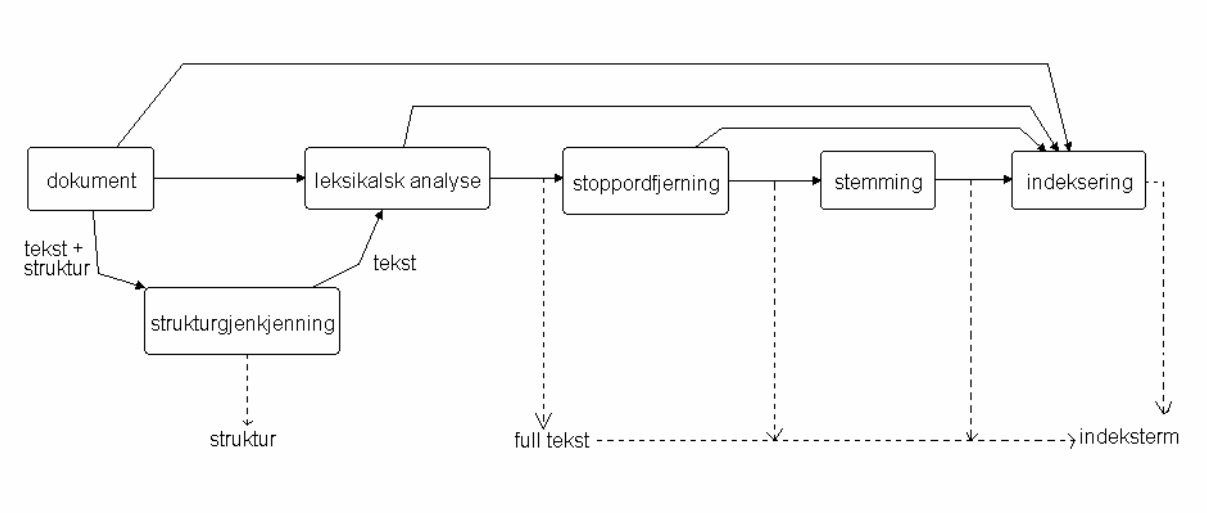
[2] Det er tre hovedtyper strukturer; fast struktur, hypertekststruktur og hierarkisk struktur. Eksempel på fast struktur er e-post (med mottaker, emne, hoveddel, avsender osv.), hyperstruktur er typisk HTML-sider (med lenker til andre sider) og eksempel på hierarkisk struktur kan være bøker inndelt i kapitler og avsnitt.

dokumentet i termer. Så kan man utføre stoppordfjerning og stemming. Dette er ikke noe man må ha med for å få lagd en indeks. Til slutt kommer man til indekseringsdelen, og det er her det endelige valget av indekseringstermer eller nøkkelord blir gjort. Det man står igjen med etter hele denne prosessen er indekseringstermer.

Den vanligste formen er altså å representere hele teksten i dokumentet med et sett av indekseringstermer eller nøkkelord. Disse indekseringstermene eller nøkkelordene er samlet i en indeks. En slik indeks er ofte organisert i et *invertert filsystem*. Det vil si at man har en liste over alle indekseringstermene (eller nøkkelordene) i dokumentsamlingen. Denne lista er gjerne sortert alfabetisk. Til hver term tilhører det en peker til en ny liste som inneholder de dokumentene den aktuelle termen forekommer i. På denne måten får man altså representert alle indekseringstermene samt hvilke dokumenter de forekommer i. Denne måten å indeksere på er den mest utbredte blant kommersielle systemer. (Faloutsos & Oard)

Et eksempel på utsnitt fra en invertert fil kan se slik ut:

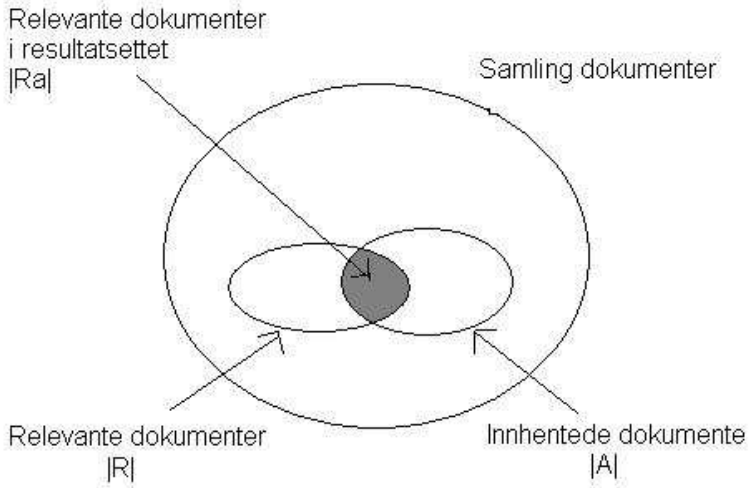
jente → dokument4, dokument7, dokument10
bil → dokument4, dokument5
bamse → dokument1, dokument2, dokument6, dokument10



Figur 5: Logical view av et dokument.

5.1.1 Presisjon og recall

Når man snakker om informasjonsgjenfinning (IR) vil det ofte være nødvendig å ha et mål for å evaluere søkestrategiene. Her snakker man om presisjon (precision) og recall. Presisjon er et mål på hvor mange av de innhentede dokumentene (|A|) som er relevante. Recall er et mål på hvor mange av de relevante dokumentene som eksisterer i dokumentsamlingen (|R|) har blitt funnet. Både recall og presisjon er vanligvis oppgitt i prosent. Presisjon og recall er illustrert i Figur 6.



Figur 6: Precision og recall.

Recall = |Ra|/|R|

Presisjon = |Ra|/|A|

5.1.2 Leksikalsk analyse

Leksikalsk analyse er prosessen å konvertere en strøm av tegn til en strøm av termer (ord). Det er altså identifiseringen av termer i et dokument. Den enkleste form for en slik analyse er å finne alle mellomrom (space) som termseparatorer og dermed har man alle termene (mellomrommene er det som skiller termene). (Holme 2000). Men det er vanligvis litt mer komplisert enn dette. Man ønsker ofte å ”rense” dokumentet for ugyldige tegn og termer. I denne delen er det i tillegg vanlig å bestemme minimum ønsket lengde for lovlige termer. For eksempel kan man bestemme at alle termer må bestå av minst to tegn fordi termer med bare

en bokstav som regel ikke er særlig nyttige i en indekseringssammenheng. Poenget med denne analysen er altså å få fjernet tegn og termer som blir sett på som dårlige identifikatorer for dokumentet, altså tegn og termer man ikke ønsker i indeksen.

Selv om denne analysen virker grei i første omgang, er den som nevnt ovenfor litt mer komplisert. Om man følger automatiserte regler, kan det være at man utelukker termer og uttrykk som er meget viktige for dokumentet. Et eksempel på dette er bruken av tall. Tall er som regel ikke gode indekseringstermer uten en kontekst. Om man for eksempel har et dokument som handler om trafikkulykker mellom 1995 og 2000, vil ikke tallene 1995 og 2000 si så veldig mye, fordi det er et intervall det er snakk om. Et søk med tallet 1997 vil da ikke gi noe resultat. Generelt vil derfor tall vanligvis være ugunstige som indekseringstermer. Men, et unntak kan være tall som er satt sammen med en term, for eksempel termen Boeing747. Dette kan være en viktig term i visse dokumenter, og det ville være dumt å fjerne det om man har en dokumentsamling som omhandler ulike flytyper og man ønsker å søke etter dem.

På samme måte som at tall kan være viktige identifikatorer har man også uttrykk bestående av flere termer som kan være viktige identifikatorer. Her er et eksempel uttrykket ”state of the art”. Her vil det være dumt å bryte opp uttrykket til fire termer. Hver term alene sier lite, men satt sammen kan det være et viktig uttrykk i visse sammenhenger. Man bør også kunne ta høyde for ulike måter å skrive uttrykket på, for eksempel ”state-of-the-art” bør behandles identisk med ”state of the art”.

Normalt vil også punktum (og andre skilletegn) være et tegn som blir fjernet, fordi det som oftest ikke har noen annen betydning enn å skille setninger fra hverandre (slik som andre skilletegn). Men i visse tilfeller kan dette være et tegn som er viktig for betydningen av en term og som man ikke ønsker å fjerne. For eksempel hvis en tekst inneholder en programmeringsbit så kan det være lurt å skille mellom variablene ”x.id” og ”xid”. Her bør man ikke fjerne punktum da det kan senke gjenfinningsmulighetene.

Vanligvis blir hele dokumentet også gjort om til kun små bokstaver (eller i visse tilfelle store bokstaver) i denne prosessen. Dette blir gjort fordi det som regel ikke spiller noen rolle for betydningen av termen om tegnene er store eller små, og man slipper å skille mellom store og små bokstaver senere i preprosesseringen av dokumentet eller i en søkesammenheng. Selv om

dette vanligvis blir gjort, er det også her problemer som kan dukke opp. I visse tilfeller kan det være at betydningen av en term blir utvidet om man endrer fra store til små bokstaver eller motsatt. Dette kan igjen føre til at et eventuelt søk etter termen gir mange flere urelevante treff i forhold til hva brukeren er på jakt etter. Et eksempel her er UNIX-termer. Dette er kommandoer som blir skrevet med store bokstaver. Om disse blir endret til små bokstaver i indekseringsprosessen tar man vekk muligheten for en bruker å kun søke etter UNIX-termene (angitt med store bokstaver), men får derimot treff fra andre dokumenter som har samme term, men som ikke er en UNIX-term dvs. har en annen betydning. Det samme gjelder også med navn.

5.1.3 Eliminering av stoppord

Det er ikke alle termer som egner seg som indekstermer. Stoppord er termer som ikke skal benyttes som indekseringstermer, og man ønsker derfor å fjerne de før man gjør en indeksering av dokumentet(ene). En liste over alle stoppordene kalles stoppordliste, og dette er en slags ”negativ ordliste”. Artikler, preposisjoner og konjunksjoner er ofte de termene som havner i stoppordlista, fordi dette er termer som egner seg dårlige som identifikatorer for et dokument, da dette er termer man vanligvis finner i de fleste dokumenter. Man kan her skille mellom domeneavhengige og generelle stoppordlister. Bestemmelsen av hva som skal være stoppord kan skje manuelt, automatisk generert (alle en- og to-stavelsesord, frekvensfordeling) eller en kombinasjon over manuelle og automatisk genererte stoppordlister. (Holme 2000).

En slik eliminering av stoppord er veldig viktig for å redusere størrelsen av indeksen; man kan her typisk redusere indeksen 40% ved denne stoppordfjerningen (Baeza-Yates & Ribeiro-Neto 1999: 167). Selv om stoppordfjerning har fordeler, kan det også være med på å redusere recall for et søk. For eksempel setningen ”to be or not to be” kan være vanskelig. Etter en stoppordeliminasjon vil vanligvis bare ordet ”be” være igjen, og da vil det være så og si umulig å skille ut de dokumentene som inneholder hele uttrykket.

Zipfs lov

For automatisk stoppordlistegenerering kan man bruke Zipfs lov. Denne loven, oppkalt etter George Kingsley Zipf, sier at man ut fra en liste av termer rangert etter forekomst kan forutsi

meget nøyaktig hvor hyppig en gitt term forekommer. Sannsynligheten for at termer skal forekomme starter høyt og avtar gradvis. Derfor vil noen få termer forekomme veldig often, mens veldig mange andre vil forekomme sjelden. Loven kan defineres som følger:

> *Hyppigheten av den i-te mest frekvente termen er* $1/i^{\theta}$ *ganger med den mest frekvente termen.*

Dette betyr at i en tekst av n termer med et vokabular av V termer, vil den i-te mest frekvente termen forekomme $n/(i^{\theta}H_V(\theta))$ ganger hvor $H_V(\theta)$ er definert som

$$H_V(\theta) = \sum_{j=1}^{V} \frac{1}{j^{\theta}}$$

Summen av alle hyppigheter er n. Dette vil si at hvis i er term nummer 1 på listen rangert etter forekomst, vil term nummer 2 forekomme $\frac{i}{2}$ ganger, term nummer 3 vil forekomme $\frac{i}{3}$ ganger osv. (Baeza-Yates & Ribeiro-Neto 1999).

5.1.4 Stemming (Rotlemmatisering)

For at man skal slippe å indeksere termer som har samme betydning men som er skrevet eller bøyd ulikt kan man utføre stemming (eller noe brukt norsk ord; rotlemmatisering). Man ønsker altså å en ordstamme for termer som egentlig har samme betydning, men som er bøyd ulikt, for eksempel jente, jentene, jenter, disse har en felles ordstamme: jente. En ordstamme er altså den delen av et termen som er igjen etter man har fjernet affiksene (prefikser og suffikser).

Stemming er med på å øke effektiviteten fordi man får en mindre indeks. Man får sannsynligvis bedre søkeresultater også, fordi man får normalt flere treff ved søk (fordi man ikke trenger å vite det eksakte ordet; hvilken bøyingsform det er i) og man kan finne dokumenter som omhandler samme tema selv om man ikke brukte nøyaktig riktig form av termen.

Det er vanlig å skille mellom fire strategier for å utføre stemming; affiksfjerning, oppslag i tabell, variasjon i etterfølger (successor variety) og n-grams. Tabelloppslag fungerer ved at det slås opp i en forhåndsdefinert tabell hvor stammen av termen finnes. Dette er en enkel metode, men er avhengig av å ha stammer for alle termer i hele språket, og kan dermed bli veldig plasskrevende. Et eksempel er vist under. Her er stammen av den aktuelle termen ”kompositt”, og en ser hvordan ulike flertallsformer av termen blir kuttet ned til samme term ved hjelp av tabelloppslag.

Term	**Stamme**
kompositt	kompositt
kompositten	kompositt
kompositter	kompositt

Her vil da for eksempel termen ”kompositten” bli forkortet til stammen av termen som nevnt altså er ”kompositt”.

Variasjon i etterfølger er basert på bestemmelsen av morfemene og bruker kunnskap fra strukturell lingvistikk. N-gram-stemming er basert på identifiseringen av digram og trigram og kan sees på som en klyngeanalyse. Affiksfjerning er som nevnt ovenfor at man fjerner perfikser og suffikser og sitter igjen med ordstammen av termen. En av de mest kjente og brukte algoritmen for denne strategien er Porteralgoritmen. (Holme 2000).

Porteralgoritmen

Porteralgoritmen ble opprinnelig skrevet av M. F. Porter i 1979 i Computer Laboratory (Cambridge, England), som en del av et større IR-prosjekt. Stemmingsalgoritmen (eller Porterstemmeren) er en prosess for å fjerne de vanlige bøyningsformer fra ord i engelsk språk. Algoritmen ble opprinnelig skrevet i Basic Combined Programming Language[3] (BCPL), og som nevnt tilpasset engelsk språk, men er senere skrevet i mange ulike programmeringsspråk og tilpasset flere språk.

[3] “BCPL (Basic Combined Programming Language) is a computer programming language that was designed by Martin Richards of the University of Cambridge in 1966; it was originally intended for use in writing compilers for other languages. Although not widely used now, it was very influential, because Dennis Ritchie would later develop the widely-used C programming language from BCPL.” (Wikipedia: BCPL).

Algoritmen bruker en suffiksliste for fjerning av suffikser. Man har altså en rekke regler for å fjerne siste del av termer. Disse reglene ble opprinnelig skrevet for engelsk språk. Under følger noen eksempler på slike regler.

sses → *ss*
ies → *i*
ss → *ss*
s → *Φ*

Her anger algoritmen at termer som ender på ”sses” skal erstattes med endingen ”ss”. Tilsvarende skal termer som ender med ”ies” erstattes med å ha ending på ”i”. På denne måten vil man indeksere termer med lik ordstamme og semantisk betydning med samme indeksterm selv om de har ulik bøyning/ending. (Baeza-Yates & Ribeiro-Neto 1999).

Denne algoritmen kan også tilpasses norsk språk. For eksempel kan man ha en regel

ene → *Φ*

som blir brukt for å angi at alle termer som slutter på ”ene” skal erstattes med ingen ending (altså den tomme mengde Φ).

5.1.5 Valg av indekstermer

Man kan velge mellom å indeksere alle termene i en tekst (fulltekstrepresentasjon), eller man kan velge å utelukke termer i indekseringen. Hvis man skal utelukke termer, er dette valg man på et tidspunkt må foreta (index terms selection). Dette valget kan man la en spesialist gjøre, eller man kan få systemet til å gjøre det automatisk.

En måte å gjøre dette automatisk på er å starte med å identifisere substantiv. Dette kan så brukes til å systematisk eliminere verb, adjektiv, adverb, bindeord, artikler og pronomen. Det er også vanlig at flere substantiv forekommer sammen, så man kan derfor lage klynge av ord som forekommer nær hverandre slik at man får et enkelt komponent for indekseringen. Hensikten med å velge ut termer på denne måten er å forhåpentligvis bygge en bedre og mindre indeks.

Valg av termer man enten ønsker å ha med i indeksen eller ikke, er vanligvis relatert til den syntaktiske naturen av termen. For eksempel har som regel har substantiv mer semantikk enn adjektiv, adverb og verb og egner seg bedre som en indekseringsterm. (Baeza-Yates & Ribeiro-Neto 1999).

5.1.6 Bruk av thesaurus

I sin enkleste form er thesaurus en synonymordliste (altså en liste over ord med en tilhørende liste over relaterte ord). En mer generell thesaurus kan også inneholde fraser istedet for bare enkeltord. Dette gjør det mulig å utvide indeksen for dokumentsamlingen, slik at det inneholder termer og uttrykk som ikke direkte er med i dokumentene, men som har samme semantiske betydning. Det vil da øke muligheten for å få flere relevante treff (man øker recall) fordi man ikke er avhengig at queryet inneholder de eksakte termene. Man får på denne måten fokusert mer på semantikken av termene i stedet for bare stavelsen eller hvilken term man velger (av termer med samme semantisk betydning). (Baeza-Yates & Ribeiro-Neto 1999).

5.2 Informasjonsgjenfinningsmodeller (IR-modeller)

Det finnes flere måter å utføre et søk på. Felles for alle søkemetodene er at man har et søkeuttrykk og en samling (for eksempel et sett med dokumenter eller objekter) man søker i. Søkeresultatet vil avhenge av hvilken metode man benytter, og presisjon og recall vil også variere ut fra søkestrategiene som blir brukt.

De mest vanlige og kjente informasjonsgjenfinningsmodellene er som nevnt boolsk modell, vektormodellen og probabilistisk modell. Disse er nærmere beskrevet i etterfølgende seksjoner.

5.2.1 Boolsk modell

Den boolske modellen er en enkel gjenfinningsmodell basert på mengdelære og boolsk algebra. Modellen indikerer hvorvidt en term befinner seg i et dokument eller ikke. Som et

resultat av dette vil termvektene være binære, enten 0 (om termen ikke forekommer) eller 1 (om termen forekommer).

Modellen benytter seg av de boolske operatorene AND, NOT og OR. Et boolsk søk er et søk hvor man kombinerer en eller flere termer ved bruk av disse boolske operatorene (slik som AND, NOT og OR). Dette vil si at søket gir treff på dokumenter som gir et sant resultat (true)[4] av søkeuttrykket (queryet). I en spørring kan man for eksempel ha uttrykket ”bil AND hus”. Da vil resultatet av et boolsk søk inneholde dokumenter som både inneholder termen bil og termen hus. De dokumentene som eventuelt kun inneholder den ene termen, enten bil eller hus vil da ikke bli gjenfunnet. Om man ønsker å finne igjen dokumenter som enten har termen bil eller hus eller begge, må man angi søkeuttrykket med operatoren OR i stedet for operatoren AND. Uttrykket vil da se slik ut; ”bil OR hus”. Ved å bruke denne enkleste form for boolsk søk vil man ved et søk få treff på dokumenter som gir sannhetsverdien sann på søkeuttrykket uavhengig av antall ganger termen eller termene forekommer i dokumentene.

En ulempe ved denne modellen er at den ikke har noen form for rangering av dokumentene, slik en skal se vektormodellen har. Det vil si at man ikke vil kunne si i hvor stor grad dokumentene som er gjenfunnet er relevante eller ikke. Denne eksakte matchingen kan igjen føre til at gjenfinningen enten blir for upresis og dermed får man veldig mange treff, eller motsatt tilfelle at man får veldig få treff.

En annen faktor ved denne modellen som kan anses som problematisk er at selv om boolsk algebra har en veldig streng og presis semantikk, vil det ikke alltid være like enkelt for en bruker å angi sitt informasjonsbehov som et boolsk uttrykk. Det viser seg at mange brukere syns dette er veldig vanskelig.

På tross av disse ulempene er denne modellen fortsatt den dominerende modellen som blir brukt i ulike informasjonsgjenfinningssystemer.

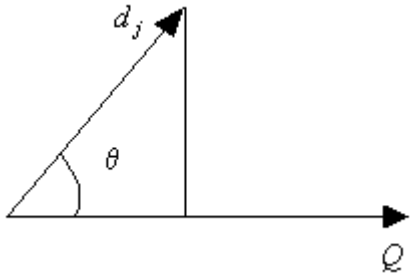
[4] Boolsk logikk er et komplett system for logiske operatorer, hentet fra logikk. ”Logic, from Classical Greek λόγος (logos), originally meaning the word, or what is spoken, (but coming to mean thought or reason) is most often said to be the study of criteria for the evaluation of arguments, although the exact definition of logic is a matter of controversy among philosophers. However the subject is grounded, the task of the logician is the same: to advance an account of valid and fallacious inference to allow one to distinguish good from bad arguments.” (Wikipedia: Logic).

5.2.2 Vektormodellen

I denne modellen lager man en vektor (K) over alle index-termer i dokumentsamlingen man ønsker å ha med. Dette kan ses på som en liste med termer hvor hver term kun forekommer én gang. Så har man en vektor for hvert dokument i dokumentsamlingen. Denne vektoren er like lang som K, men denne inneholder tall. Dette tallet sier hvor mange ganger den gitte termen er med i dokumentet, eller om det er med i det hele tatt (0 – termen er ikke med). Hver term har altså en vekt. Hvordan denne vekten bestemmes er beskrevet nærmere i seksjon 5.3.

Når man så skal søke i dokumentsamlingen, omformes også søkeuttrykket (queryet) til en vektor. Denne består av tall på samme måte som dokumentvektorene. Man kan da måle likheten mellom dokumentvektoren og queryvektoren og på denne måten finne likheten. Slik kan man rangere ulike dokument for å finne hvilket som gir best resultat (passer best med) til søkeuttrykket.

Et dokument d_j og et søkeuttrykk (query) er representert som n-dimmensjonale vektorer som vist i Figur 7.



Figur 7: Cosinus av θ, sim(d_j, q).

Vektormodellen foreslår å evaluere graden av similaritet mellom dokumentet d_j med søkeuttrykket q som korrelasjonen mellom de tilsvarende vektorene $\vec{d}_j$ og $\vec{q}$. Denne similariteten kan for eksempel angis som cosinus av vinkelen mellom disse to vektorene. Denne similariteten angis slik:

$$sim(q, d_j) = \frac{\sum_{k=1}^{n} w_{kj} r_k}{\sqrt{\sum_{k=1}^{n} w_{kj}^{\ 2} \sum_{k=1}^{n} r_k^{\ 2}}}$$

Dette cosinusmålet angir størrelsen på vinkelen mellom to vektorer. Denne størrelsen varierer fra 0 til 1;

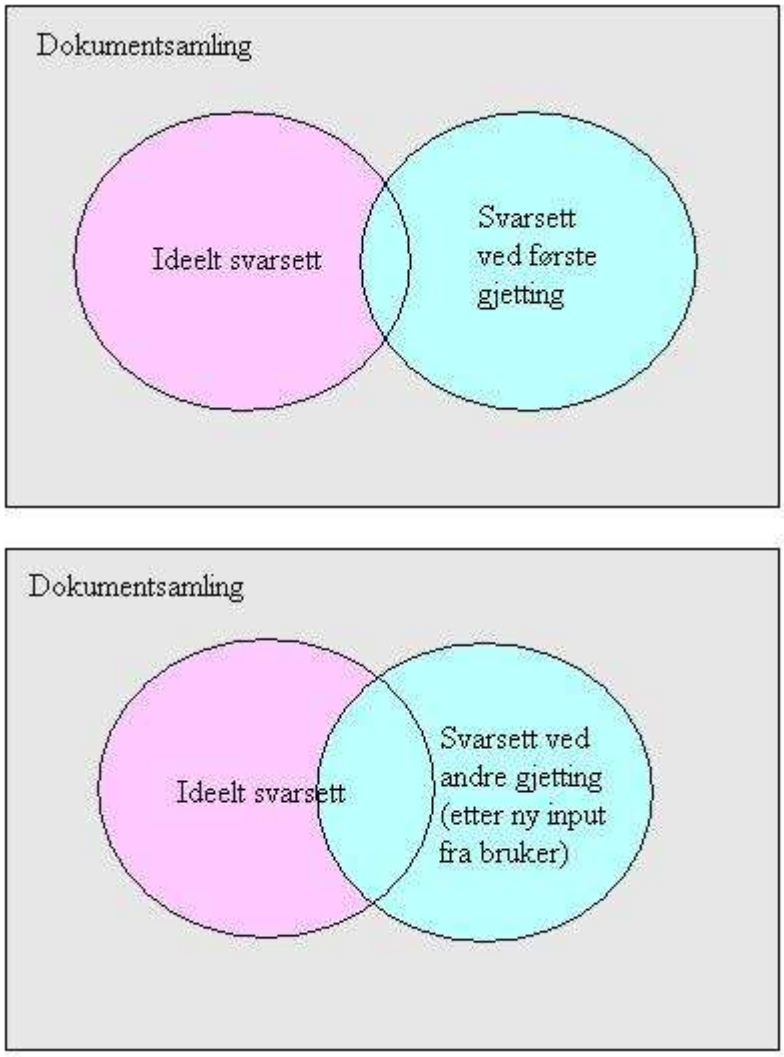
$$0 \leq \text{sim}(q, d_j) \leq 1.$$

Dette vil si at jo nærmere to vektorer er hverandre, jo mindre blir vinkelen mellom dem og jo større blir cosinusmålet. Om to vektorer er identiske med hverandre blir coninusmålet lik 1.0. For to vektorer som ikke har noen likhet vil de stå vinkeltrett på hverandre og ha cosinusmål lik 0.

I motsetning til den boolske modellen som angir om en term er med i et dokument eller ikke og derfor kun sier om et dokument er relevant eller ikke, gjør vektormodellen det mulig å angi grad av likhet mellom ulike dokumenter og derfor kan man også angi grad av relevans. Man kan på denne måten rangere søkeresultatet etter relevans.

5.2.3 Probabilistisk modell

Den probabilistiske modellen baserer seg på følgende ide: Gitt et query er det gitt et ideelt sett (ideelt svarsett) ut fra dokumentsamlingen som brukeren vil finne relevant. Dette settet er ikke kjent ved starten, så systemet må gjette et initielt sett for å få første svarsett med dokumenter. Dette vil være en midlertidig probabilistisk (sannsynlig) beskrivelse av det ideelle svarsettet. Brukeren må så se på svarsettet og velge ut de som er relevante og de som ikke er det. Systemet vil da bruke denne informasjonen fra brukeren til å forbedre svarsettet. Denne interageringen mellom brukeren og systemet er en iterativ prosess som må gjentas flere ganger. På denne måten vil svarsettet utvikle seg og sannsynligvis etter hvert komme nærmere den reelle beskrivelsen. Denne interageringen er vist i Figur 8.



Figur 8: Svarsett etter første og andre gjetting i en probabilistisk modell.

Antakelsen for denne modellen er som følger: Gitt et brukerquery *q* og et dokument d_j i samlingen. Den probabilistiske modellen vil prøve å estimere sannsynligheten for at brukeren vil finne dokument d_j relevant. Modellen antar at denne sannsynligheten kun avhenger av queryet og dokumentrepresentasjonen. Modellen antar også at det er et subset av alle dokument som brukeren vil foretrekke som svarsett for queryet *q*. Et slikt ideelt svarsett skal maksimere den totale sannsynligheten av relevans til brukeren. Dokument som befinner seg i dette svarsettet er antatt å være relevant til queryet, og dokumentene som ikke befinner seg i svarsettet er antatt å være irrelevant. (fritt oversatt fra Baeza-Yates & Ribeiro-Neto 1999: 31).

Hovedfordelene med denne modellen er at i teorien blir dokumentene rangert i forhold til sannsynligheten av å være relevante. Dette kan være veldig nyttig for brukeren, og spare tid ved å bedømme hvor relevante dokumentene er. Ulemper med denne modellen er at systemet

må gjette den initielle separeringen av relevante og irrelevante dokumenter, modellen tar ikke med i betraktingen termfrekvenser (altså hvor ofte en term forekommer i et dokument) og den baserer seg på at indekstermene er uavhengige av hverandre (selv om dette ikke nødvendigvis alltid er en ulempe). (Baeza-Yates & Ribeiro-Neto 1999).

5.2.4 Andre IR-modeller og søkemetoder

Fritekstsøk er en metode som ofte blir benyttet i ulike søkemodeller. Her søker man i teksten til dokumentene og man matcher søkeuttrykket med denne teksten. Søkeuttrykket kan være et enkeltord, en frase eller en setning. Om man finner dokument som matcher søkeuttrykket (dvs. er likt) vil dokumentet være med i resultatsettet.

Kombinasjon av et boolsk søk (beskrevet i seksjon 5.2.1) og fritekstsøk, kalles ofte *hybrid søk*.

Om man ønsker å søke i bestemte felt, dette kan være gunstig om man for eksempel skal søke i en samling med e-post, kan man benytte *feltsøk.* Dette betyr at man kun søker i bestemte felt i dokumentet. E-post inneholder som kjent blant annet felt som mottaker, emne og mottaker. Man kan da for eksempel søke i mottakerfeltet uten å bry seg om resten av e-posten (om man ønsker å finne alle e-poster som har blitt sendt til en bestemt person).

Fuzzy søk er en informasjonsgjenfinningsmodell som gjør det mulig å søke på termer og uttrykk som ligner på de i søkeuttrykket, man er altså ikke avhengig av at termen eller uttrykket er helt likt som søkeuttrykket. Man kan i denne modellen velge om man ønsker å søke direkte i dokumentet eller via metadata eller annen informasjon som beskriver dokumentet. Denne modellen baserer seg på fuzzy sett-teori, hvor man kan ha delvis klassetilhørighet. Klassetilhørighetene er altså ikke fast bestemt, og et element kan ha klassetilhørighet til flere klasser. Dette vil si at elementet har gradvis tilhørighet til flere klasser. På samme måte kan man altså søke på termer og få treff på termer som ikke er helt like, men som delvis tilhører samme ”klassen” (altså samme termen).

For å få definert denne klassetilhørigheten kan man bygge opp en thesaurus (se seksjon 5.1.6). Denne thesaurusen kan bli konstruert ved å definere en term-term-korrelasjonsmatrise hvor radene og kolonnene er assosiert til indekstermene i dokumentsamlingen. Man kan dermed

bruke disse termkorrelasjonene for å definere et fuzzy sett assosiert til hver indeksterm. I dette fuzzy settet vil hvert dokument i samlingen ha en grad av tilhørighet. Denne tilhørigheten kan beregnes som en algebraisk sum over alle termene i det bestemte dokumentet. Et dokument vil ha tilhørighet til det gitte fuzzy settet assosiert til den gitte termen hvis det har egne termer som er relatert til den gitte termen. (Baeza-Yates & Ribeiro-Neto 1999).

Det er ofte slik at flere termer har en tilknytning til hverandre, men man vet ikke hvilken rekkefølge de forekommer i, eller om de står skrevet rett etter hverandre i teksten man ønsker å søke i. Et eksempel kan være et dokument som inneholder termene ”mat” og ”hund”. Man ønsker så å finne igjen de dokumentene som har disse termene i nærheten av hverandre, altså at mat er i forbindelse med hund. Man ønsker derimot ikke å finne dokument som tilfeldigvis inneholder begge termene, men hvor de ikke har noen forbindelse med hverandre (for eksempel et dokument som handler om hund, og så står det tilfeldigvis noe om mat helt på slutten, men ikke i forbindelse med hund). Man kan da benytte *proximity search* (nærhetssøk). Denne metoden gjør at man kan finne igjen dokumenter som har søketermene i nærheten av hverandre, ofte angitt slik ”søketerm1 NEAR søketerm2”. Andre slike queryoperatorer som NOT NEAR, FOLLOWED BY, NOT FOLLOWED BY, SENTENCE og FAR kan også brukes for å angi ulike nærheter. (Wikipedia: Proximity Search).

Til slutt har man metoder som *lingvistisk søk* og *parametrisk søk.* I lingvistisk søk søker man på betydningen av termene og ikke selve termene. Alle dokumentene i en slik samling må da ha blitt utført stemming på før en indeksering. Dette vil si at man kan søke på termen ”komponenter” og få treff på dokument som inneholder termen ”komponent”. Parametrisk søk gir muligheten for å søke på objekter som inneholder en bestemt numerisk verdi eller attributt slik som datoer, heltall (integer) eller andre numeriske datatyper.
(IBM Glossary).

5.3 Vekting

Den enkleste måten å ”vekte” et dokument på, er boolsk vekting. Her tilordnes vektene etter om termen er med i dokumentet eller ikke. Er termen med i dokumentet blir vekten 1, og er den ikke med blir vekten 0.

Antakelsen om at hvis en term forekommer ofte i et dokument, er dette en term som sier mye om dokumentets innhold og som bør tilordnes en høy vekt, noe som indikerer en høy signifikans. Termfrekvensvekting styres på samme måte som ved boolsk vekting etter om en term er med i dokumentet eller ikke, men her tar man også med hvor mange ganger en term forekommer i beregningen av vekten. Det vil si vekten blir det samme som antallet termen forekommer i det gitte dokumentet.

Et mer nyansert syn på dette vil være antakelsen om at dersom en term forekommer i mange av dokumentene i samlingen, vil ikke termen anses som så viktig, og dermed er en lavere vekting ønsket, selv om termen forekommer mange ganger i et bestemt dokument. Man kan da operere med termfrekvens (TF, antall ganger termen forekommer i et bestemt dokument) og invers dokumentfrekvens (IDF). Invers dokumentfrekvens er definert som:

$$\log_2(\frac{N}{nt})$$

hvor N er det totale antallet dokumenter i samlingen og nt er antall dokumenter som termen forekommer i. Man kan da angi vektingen med TF * IDF. TF for en term som forekommer hyppig i et dokument vil som nevnt bli høy, men jo flere dokumenter termen forekommer i, jo lavere blir IDF, og dermed også termens vekt.

5.4 Metadata

Metadata kan defineres som ”data om data”. Dette er attributter som beskriver et objekt, vanligvis inndelt i kategorier eller deler. Metadata inneholder informasjon om objektet (dokumentet), dets (eventuelle) domeneavhengighet og relasjoner mellom ulike objekter. Et vanlig eksempel for å beskrive metadata er bibliotekskort. Hver bok i et bibliotek har et tilhørende kort hvor det står forfatter, tittel, kategori osv. All dataen som trengs for finne igjen boka står på kortet. (Wikipedia: Metadata).

Metadata er ofte organisert etter ulike standarder. Dublin Core og MARC er eksempler på ulike standarder for metadata. Slike standarder gjør at man får en felles måte å beskrive

objektene på, slik at man lett kan bruke samme regler for gjenfinning senere. Noen av bruksområdene til metadata er oppdaging, lokalisering, relevansvurdering og annen evaluering, utvelgelse og dokumentasjon. Et av målene ved bruk av metadata er å forbedre presisjonen ved søking.

Fordelene med metadata er at man som sagt får samlet data om objektene i samlingen på en felles måte og dermed lett kan kategorisere objektene og klassifisere innholdet. En annen fordel er at det gjør det mulig å innlemme metadataen som en del av selve primærdokumentet (for eksempel vha. HTML eller SGML). (Husby 1997)

I dagens systemer må som regel metadataene skrives inn manuelt for hvert objekt. De fleste vil mene at metadatametodene er relativt enkle så det kreves ingen ekspertise. Andre vil derimot mene at selv om selve konseptet er enkelt, vil noen av standardene for metadataene være så kompliserte og tidkrevende at det vanskelig lar seg gjøre. Noen objekter kan være veldig vanskelig å klassifisere på denne måten også (fordi man føler at elementene i standarden ikke passer). Selv om det eksperimenteres med metoder for å automatisk bestemme metadata, er dette noe som langt fra er ferdigutviklet og etablert i dagens informasjonssystemer.

6 Klyngeanalyse

Klyngeanalyse er en vanlig teknikk for statistisk dataanalyse som er brukt i mange felt, inkludert maskinlæring, datamining, mønstergjenkjenning, bildeanalyse og bioinformatikk. Klyngeanalyse er en teknikk for å klassifisere eller organisere like objekter i samme grupper, slik at dataen i hver gruppe (ideelt sett) deler felles egenskaper, og at de er nære hverandre i forhold til definerte likhetsmål (similaritet). (Wikipedia: Data clustering). Det man oppnår ved en slik analyse som i vårt tilfelle er i en søkesammenheng er altså å automatisk organisere treffene i ulike klynger, hvorav ordet klyngeanalyse.

De mest vanlige og mest brukte søkemotorene i dag representerer søket som en sekvensiell liste med alle treffene. Dette kan ofte være tungvint og til lite hjelp for brukeren. Det viser seg at 90 % av brukerne (de som utfører spørringer) undersøker bare treffene på første side i listen og bryr seg lite om resten. (Holme 2000). Det man derimot kan oppnå ved å utføre klyngeanalyse er å få organisert resultatene på en slik måte at brukeren enkelt og intuitivt kan navigere seg fram til det som er ønskelig. Dette resultatet kan for eksempel representeres grafisk i et tredimensjonalt rom hvor hver node i grafen representerer en klynge eller et objekt. Så kan brukeren trykke på nodene og på denne måten navigere seg i søkeresultatet; få opp nye klynger, se relasjonene mellom objektene osv.

Klyngene kan være med på å gjøre det enklere for brukeren ved brukeren raskt får dannet seg et overblikk over søket, og hvordan resultatene henger sammen med hverandre. Så i stedet for å presentere søkeresultatet som en liste, ser man på sammenhengen mellom de ulike treffene og om de kan organiseres i ulike emneområder. Ikke bare i søk kan klyngeanalyse benyttes til noe nyttig, men også i organiseringen av kunnskapsobjektene slik vi ser for oss, og som nevnt tidligere i oppgaven. En bruker kan ved hjelp av denne metoden automatisk få organisert et datasett hvis ønskelig.

Det finnes mange ulike teknikker for å utføre en klyngeanalyse, og disse kan hovedsakelig deles inn i to typer. Dette er hierarkiske algoritmer og ikke-hierarkiske algoritmer. Ved hierarkiske algoritmer finner man etterfølgende klynger ved å bruke de allerede etablerte klyngene. Disse algoritmene kan konstruere klyngene ved å bygge opp fra bunnen av

(bottom-up) eller ved splitting av en klynge (top-down). Ved ikke-hierarkiske algoritmer organiserer man alle klyngene på en gang i en iterativ prosess.

Det er i denne oppgaven gjort et utvalg av teknikker som er presentert. Under ikke-hierarkiske klyngemetoder er det i etterfølgende seksjoner lagt hovedvekt på K-means, mens under hierarkisk er det lagt hovedvekt på bottom-up måte å utføre klynging på. Det blir i tillegg gjennomgått to andre teknikker for klynging, nevrale nett og spektralklyning. Disse kan ikke klassifiseres som en av hovedtypene. Til slutt gis det et eksempel på hvordan man kan velge antall klynger.

6.1 Ikke-hierarkisk klynging

I en ikke-hierarkisk klyngemetode blir alle klyngene organisert på en gang. Dette foregår vanligvis i en iterativ prosess og man stopper når klyngene lenger ikke forandrer seg eller man har kjørt et visst antall ganger (etter et forhåndsdefinert kriteria).

6.1.1 K-means

Denne algoritmen organiserer et gitt datasett i et antall klynger som på forhånd er kjent. Hvert element blir tilordnet den klyngen som har det senteret (klyngerepresentanten) som ligger nærmest. Dette senteret er et gjennomsnitt av alle elementene (punktene) i klyngen. Målet er altså å dele en vektormengde i K klynger. Algoritmen er en iterativ prosess og kan beskrives som følgende:

- Steg 1: Velg ut K forskjellige klyngerepresentanter (denne utvelgelsen skjer tilfeldig) c_1, c_2, c_3, ..., c_k
- Steg 2: For hvert element i mengden, regn ut likheten til alle klyngerepresentantene. Legg elementet i den klyngen som den nærmeste klyngerepresentanten representerer
- Steg 3: For hver klynge, regn ut ny klyngerepresentant ved å ta gjennomsnittet for hele klynga.
- Steg 4: Gjenta fra steg 2 helt til klyngene ikke lenger flytter på seg (dvs. klyngerepresentantene ikke endres), man har da oppnådd konvergens.

En av ulempene med k-means-algoritmen er at man på forhånd må vite hvor mange klynger man ønsker. Det kan i mange tilfeller være veldig vanskelig å vite på forhånd hva det optimale antallet klynger vil være, særlig hvis man vet lite om dokumentsamlingen. I vår modell anses ikke dette som et problem, fordi det jobbes med et område hvor brukeren oftest vil vite hvor mange klynger som er ønskelig. Det antas med andre ord at brukeren som oftest har en viss kjennskap til dokumentsamlingen.

Et annet problemområde med k-means-algoritmen er at den ofte vil være svært sensitiv i forhold til hva som blir valgt som klyngerepresentanter i den initielle fasen. Det mest vanlige (og som algoritmen tilsier) er å velge disse tilfeldig. Men hvis disse blir valgt tilfeldig vil ikke algoritmen gi samme resultat ved hver analyse siden resultatklyngene avhenger av de initielle tilfeldige tilordningene. Det blir senere i oppgaven vist at de mest optimale resultatene blir oppnådd om man velger ut representanter manuelt ut fra kjennskapen man har til dokumentsamlingen på forhånd.

Selv om det er bevist at denne algoritmen alltid vil terminere, så er det ikke sikkert at den vil finne den mest globale optimale klyngefordelingen. Den maksimerer inter-klynging (eller minimerer intra-klynger) forandringer, men forsikrer altså ikke at resultatet har et globalt minimum av forandringer. (K-means Clustering).

Hovedfordelene med denne algoritmen er dens enkelhet og hastighet som gjør det mulig å la den kjøre på store datasett.

6.1.2 Andre ikke-hierarkiske klyngemetoder

QT Clust algorithm

QT (Quality Threshold) klyngeanalyse (Heyer et al, 1999) er en alternativ metode for ikke-hierarkisk klynging. Denne ble utviklet for genklynging. Metoden ligner på K-means, men her trenger man ikke vite antall klynger man ønsker på forhånd. Denne metoden krever mer ressurser enn K-means, men vil også alltid gi samme resultatklynger ved flere kjøringer av en datasamling, noe K-means ikke gjør (fordi K-means i utgangspunktet starter med tilfeldige klyngerepresentanter).

Algoritmen kan beskrives som følgende:

- Steg 1: Brukeren velger en maksimal diameter for klynger.
- Steg 2: Bygg en kandidatklynge for hvert element (punkt) ved å inkludere det nærmeste elementet, det nest nærmeste og så videre til diameteren av klyngen overgår det gitte thresholdet (angitt av brukeren i steg 1).
- Steg 3: La den kandidatklyngen med flest element i seg være den første klyngen, og fjern alle element i klyngen slik at de ikke blir tatt med i videre beregninger.
- Steg 4: Gjenta fra steg 2 med det reduserte antall element.

Avstanden mellom et element og en gruppe av element blir beregnet ved å bruke complete link, nærmere beskrevet i seksjon 6.4.2. (Wikipedia: Data clustering)

Fuzzy K-means-klynging

I fuzzy klyngeanalyse har hvert element en grad av tilhørighet til alle klynger (som i fuzzy-logikk) i stedet for å kun tilhøre èn klynge, som de fleste andre klyngemetodene baserer seg på. På denne måten er elementene som ligger ytterst i en klynge, med i klyngen i en mindre grad enn de som ligger i senteret av klyngen. For hvert element *x* er det en koeffisient som angir graden av klyngetilhørigheten til klynge nummer *k,* $u_k(x)$. Vanligvis er summen av disse koeffisientene definert til å være 1, slik at $u_k(x)$ angir sannsynligheten for å tilhøre en bestemt klynge:

$$\forall x \sum_{k=1}^{\text{num. clusters}} u_k(x) = 1 .$$

I denne algoritmen er senteret (klyngerepresentanten) i en klynge gjennomsnittet av alle element, vektet med graden av deres tilhørighet til klyngen:

$$\text{center}_k = \frac{\sum_x u_k(x) x}{\sum_x u_k(x)} .$$

Graden av tilhørigheten er relatert til den inverse avstanden til klyngen

$$u_k(x) = \frac{1}{d(\text{center}_k, x)}$$

så er koeffisientene normalisert og fuzzyfisert med en parameter $m > 1$ slik at summen blir 1. Så

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d(\text{center}_k, x)}{d(\text{center}_j, x)} \right)^{1/(m-1)}}.$$

Hvis m er lik 2 er dette ekvivalent med å normalisere koeffisientene lineært for at de skal ha sum 1. Når m er nærme 1 så er klyngesenteret nærmest elementet gitt mye høyere vekt enn de andre, og algoritmen er derfor svært lik k-means.

Hele algoritmen er som sagt veldig lik k-means-algoritmen og kan beskrives som følgende:

- Steg 1: Velg et antall klynger.
- Steg 2: Tilordne tilfeldige koeffisienter for hvor stor grad de tilhører klyngene til hvert element.
- Steg 3: Gjenta til algoritmen konvergerer (dvs. koeffisentene mellom to iterasjoner er ikke større enn ε, det forhåndsdefinerte thresholdet):
    - Beregn senteret (klyngerepresentant) for hver klynge ved å bruke formelen beskrevet ovenfor.
    - For hvert element, beregn koeffisienten for hvor stor grad de tilhører klyngene ved å bruke formelen beskrevet ovenfor.

Denne algoritmen har samme problem som k-means, den minimerer intra-klynge-forandringer, men dette minimumet er et lokalt minimum, og resultatet avhenger av det initielle valget av vekter. (Wikipedia: Data clustering)

6.2 Hierarkisk klynging

Hierarkisk klynging bygger opp (bottom-up) eller splitter (top-down) et hierarki av klynger. Den tradisjonelle representasjonen av dette hierarkiet er en tredatastruktur, også kalt *dendrogram*, med individuelle elementer i en ende av treet og en klynge med hvert element i den andre.

Et viktig moment ved hierarkisk klyngeanalyse er å velge avstandsmål. Et enkelt avstandsmål er *Manhattan distance*. Denne er lik summen av de absolutte avstandene for hver variabel. Dette målet er også kjent som ”city block distance” eller ”taxi-cab distance”, fordi det er den korteste veien en bil ville kjørt fra et sted til et annet i en by med kvartaler som i Manhattan. Variablene kan altså plottes i et rutenett som kan sammenlignes med bygater. For eksempel i planet vil Manhattan-avstanden mellom punkt P1 med koordinater (x_1, y_1) og punkt P2 på (x_2, y_2) være

$$|x_1 - x_2| + |y_1 - y_2| .$$

Et mer vanlig avstandsmål er *Euklidisk distance*. Denne blir beregnet ved å finne avstanden mellom hver variabel, summere kvadratene og finne kvadratroten av den summen. Den euklidiske avstanden mellom to punkt P = (p_1, p_2, …, p_n) og Q = (q_1, q_2, …, q_n) i et euklidisk n-rom er definert som:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Altså den korteste veien (i luftlinje) mellom to punkt. I et tilfelle med to variabler vil da avstanden være det samme som å finne lengden av hypotenus i en trekant.

En undersøkelse av klyngeanalyse i forskning innenfor helsepsykologi fant at det mest vanlige avstandsmålet i publiserte studier i det forskningsområdet er Euclidean distance eller den kvadratiske Euclidean distance (Clatworthy et al, 2005).

6.2.1 Bottop-up (Agglomerative[5] clustering)

Ved bottom-up starter man med like mange klynger som dokumenter i samlingen. Dette er en rekursiv prosess hvor man starter ved å beregne avstanden mellom de ulike klyngene, og finner de to klyngene som ligger nærmest hverandre. Disse to klyngene erstattes med en ny klynge. Denne nye klyngen blir så sammenlignet med en annen klynge. Slik fortsetter prosessen til man har én klynge, organisert i et hierarkisk tre.

Algoritmen kan beskrives stegvis slik:

[5] ”agglomerere v. ((agglomererer - agglomererte - agglomerert)) dynge sammen” (Clue for Windows v.6.3, Clue International Corporation, 1991-2005).

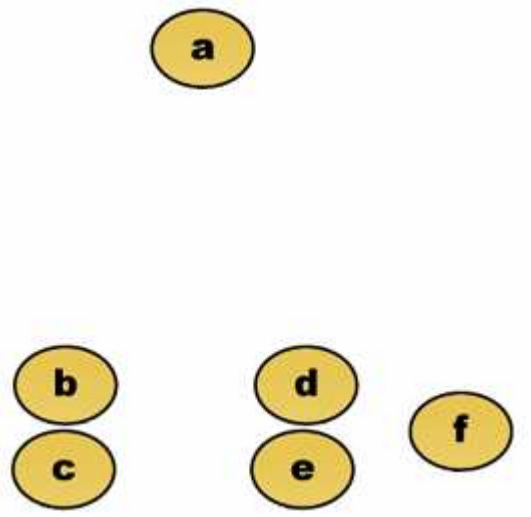
- Steg 1: Tilordne hvert element til en klynge, slik at om du har N elementer har du N klynger (hver klynge består altså av kun ett element).
- Steg 2: Finn det paret som er nærmest hverandre (kortest avstand), og la de bli til en klynge, slik at mengden nå inneholder en klynge mindre.
- Steg 3: Beregn avstandene mellom den nye klynga og hver av de andre klyngene.
- Steg 4: Gjenta steg 2 og 3 til alle elementene er klynget sammen til en stor klynge med størrelse N.

Ønsker man k antall klynger, kan man bare kutte den k-1 lengste linken.

Steg 3 kan utføres på flere forskjellige måter, men de mest vanlige er single-linkage, complete-linkage and average-linkage klynging. Disse er nærmere beskrevet i seksjon 6.4.1, 6.4.2 og 6.4.3.

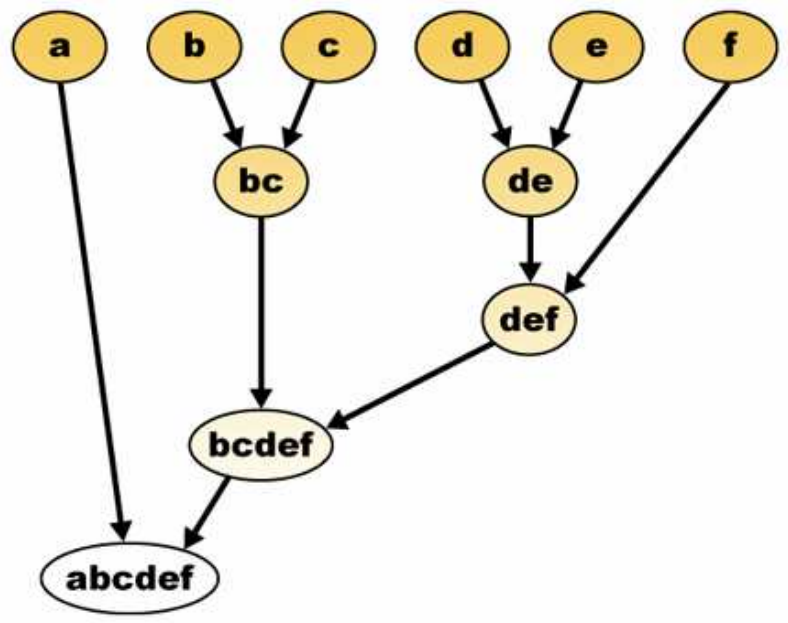
Å kutte et tre ved en gitt høyde vil gi en klyngesamling med den valgte presisjonen. I det følgende eksemplet, vil en kutting etter den andre raden gi klyngene {a} {b c} {d e} og {f}. Hvis man derimot kutter treet etter den tredje raden vil man få klyngene {a} {b c} og {d e f}, som da er en ”grovere” klynging med færre antall av større klynger.

For eksempel har man følgende data (Figur 9) som skal klynges og man bruker euklidisk avstand som avstandsmål.



Figur 9: Datasett som skal klynges (Wikipedia: Data Clustering).

Et hierarkisk dendrogram av disse dataene vil da se slik ut:



Figur 10: Hierarkisk dendrogram (Wikipedia: Data Clustering).

Denne metoden bygger altså hierarkiet fra de individuelle elementene (klyngene) ved å gradvis flette sammen klynger. Vanligvis starter man som sagt ved å flette de to klyngene (elementene) som ligger nærmest hverandre. Man trenger her et avstandsmål og vanligvis er, som nevnt tidligere single link, complete link eller average link de metodene som blir brukt.

Hver fletting (økning av klyngene) vil forårsake en større avstand mellom klyngene enn ved de tidligere økningene, og en kan bestemme å stoppe klyngingen enten når klyngene er for langt fra hverandre til å bli flettet (etter et forhåndsdefinert kriterium), eller når det er et tilstrekkelig lite antall klynger (etter et forhåndsdefinert kriterium). (Wikipedia: Data clustering).

6.2.2 Top-down (Divisive clustering)

Ved top-down-klynging starter man med å se på hele settet som én klynge. Denne blir så stegvis splittet opp i flere klynger, til man har de klyngene man ønsker. En illustrasjon av dette er å se på Figur 10 og følge pilene motsatt vei.

Algoritmen er som følger:

- Steg 1: La alle elementene være i en eneste klynge, og bestem en terskelverdi på avstand.

- Steg 2: Beregn avstanden mellom alle par i en gruppe, og velg det paret med størst avstand.
- Steg 3: Sammenlign denne avstanden med den forhåndsbestemte terskelverdien.
    - Hvis avstanden er større enn terskelverdien er denne gruppen delt i to. Dette blir gjort ved å plassere det valgte paret i to ulike grupper og bruke dem som seed. Undersøk alle andre elementer i gruppa og plasser elementene i den nye gruppa som er nærmest (i avstand) som seedet. Gjenta fra steg 2.
    - Hvis avstanden mellom det valgte paret er mindre enn terskelverdien stopper algoritmen.

6.3 Andre klyngealgoritmer

I tillegg til de ikke-hierarkiske og hierarkiske klyngeanalysealgoritmene kan andre metoder benyttes. Nevrale nett og spektralklynging er eksempel på dette. Disse er beskrevet i etterfølgende seksjoner.

6.3.1 Nevrale nett

Kunstige nevrale nett er inspirert fra biologiske nevrale nett. Disse biologiske nettene finnes i hjernen. Det har blitt en vanlig oppfatning at hjernen er bygd opp av nerveceller som er knyttet sammen i en meget komplisert nettstruktur.

Hver nervecelle kan bli sett på som en basis prosesseringsenhet. Denne enheten kan sende ut utputtsignaler som en reaktiv aksjon om den mottar bestemte inputsignaler. Disse signalene som så blir sendt ut blir sendt som input til andre nerveceller (gjennom synaptiske forbindelser) som igjen kan sende ut nye utputtsignaler. Denne prosessen kan gjentas mange ganger gjennom mange lag av nerveceller og blir referert til som ”spread activation prosess”. Som et resultat av denne prosessen blir inputinformasjonen analysert og tolket og kan gjøre så hjernen kommanderer fysiske reaksjoner (motoraksjoner) i respons.

De kunstige nevrale nettene er forenklede matematiske modeller (grafrepresentasjoner) av tenkte, biologiske nett. Nodene i grafen representerer nerveceller og kantene i grafen representerer de synaptiske forbindelsene.

Et nevralt nett er en konstruksjon som kan lagre kunnskap som det mottar gjennom eksempler. Dette betyr at et kunstig nevralt nett må læres opp for å kunne brukes til problemløsning. Læringen foregår ved å gi inn eksempler med løsninger til nettet. Etter hvert vil nette justere seg og har på denne måten ”kunnskap” om emnet. Den kunnskapen som gjennom denne læringsprosessen blir lagret i nettet, kan senere benyttes på tilsvarende problemer og oppgaver. Et nevralt nett kan ses på som en rettet graf hvor det er knyttet vekter til kantene i grafen. Den kunnskapen som nettet inneholder ligger lagret i vektene. (Holme 2000).

Adaptive Resonance Theory

Stephen Grossberg introduserte i 1976 det som ble kalt Adaptive resonance theory (ART). Dette omslutter et vidt spekter av ulike nevrale nett basert eksplisitt på nevrofysiologi. ART-nett er definert algoritmisk i form av detaljerte differensialligninger med intensjon om å være plausible modeller av biologiske nerveceller. I praksis er ART-nett implementert ved å bruke analytiske løsninger eller tilnærminger til disse differensialligningene. Den modellen som Grossberg lanserte kalles ART-1. Senere har det blitt utviklet andre ART-modeller, blant annet ART-2, ARTMAP, fuzzy ART og fuzzy ARTMAP. (ART).

I klyngeanalyse kan også ART-nett benyttes. Det som skiller ART-nett fra de fleste andre kunstige nett er at ART-nett har evnen til å stadig tilegne seg ny kunnskap uten at det går på bekostning av allerede tilegnet kunnskap. Et ART-nett inneholder altså et minne der tidligere klassifiserte vektorer ligger lagret. Når så en ny vektor skal klassifiseres legges den klyngen med størst likhet til vektoren. Hvis ingen av de eksisterende klyngene har tilstrekkelig similaritet (etter en forhåndsdefinert grenseverdi) opprettes en ny klynge med inputvektoren som foreløpig eneste medlem. Hvis derimot vektoren tilordnes en allerede eksisterende klynge, vil vektene til den aktuelle klyngen for at den bedre skal matche inputvektoren. Algoritmen for et slikt nett kan i grove trekk beskrives slik:

- Steg 1: Initialisering. La mengden prototypvektorer være tom.

- Steg 2: Presenter ny inputvektor og finn nærmeste prototypvektor (om mulig).
- Steg 3: Sjekk om prototypvektoren er for langt fra inputvektoren.
    - Dersom de er for langt fra hverandre eller det ikke finnes noen klyngevektorer, opprett en ny klynge med klyngevektor lik inputvektoren. Gå til steg 2.
- Steg 4: Oppdater klyngevektor ved å flytte den nærmeste inputvektoren.

6.3.2 Spektralklynging

For et sett av datapunkt (elementer) kan similaritetsmatrisen beskrives som en matrise S hvor S_{ij} representerer et mål for likheten mellom punkt i og j. Spektralklyngeteknikker bruker spektrumet av similaritetsmatrisa for å klynge punktene. Man kan også benytte samme teknikk for å utføre en dimensjonreduksjon for å klynge i færre dimmensjoner enn hva det opprinnelig er. Her er algoritmene Shi-Malik og Meila-Shi ofte benyttet. (Wikipedia: Data clustering)

6.4 Avstandsmål mellom klynger

For at man skal kunne sammenligne og se på avstanden mellom elementer i klyngene trenger man et avstandsmål. Dette blir gjort forskjellig, men de mest vanlige er som nevnt tidligere single link, complete link og average link. Disse er beskrevet under.

6.4.1 Single link (nearest neighbour)

Single link er den minimale avstanden mellom et element fra hver klynge. Dersom det er to klynger, A og B er denne avstanden definert som:

$$\min\{d(x,y) : x \in \mathrm{A}, y \in \mathrm{B}\}$$

6.4.2 Complete link (furthest neighbour)

Complete link er den maksimale avstanden mellom et element fra hver klynge. Dersom det er to klynger, A og B er denne avstanden definert som:

$$\max\{d(x,y) : x \in \mathrm{A}, y \in \mathrm{B}\}$$

6.4.3 Average link

Average link er den gjennomsnittlige avstanden mellom et element fra hver klynge. Dersom det er to klynger, A og B er denne avstanden definert som:

$$\frac{1}{card(\mathrm{A})card(\mathrm{B})}\sum_{x\in \mathrm{A}}\sum_{y\in \mathrm{B}} d(x,y)$$

6.4.4 Andre avstandsmål

Andre avstandsmål enn de beskrevet ovenfor er:

- Summen av alle intraklynge-variasjoner. Det vil si den totale summen av variasjoner innenfor de ulike klyngene.
- Økningen av variasjonen for klyngene som blir flettet (Wards kriteria).
- Ved at man bruker sannsynligheten for at kandidatklynger oppstår fra samme distribusjonsfunksjon (Vaccardo-link).

(Wikipedia: Data clustering).

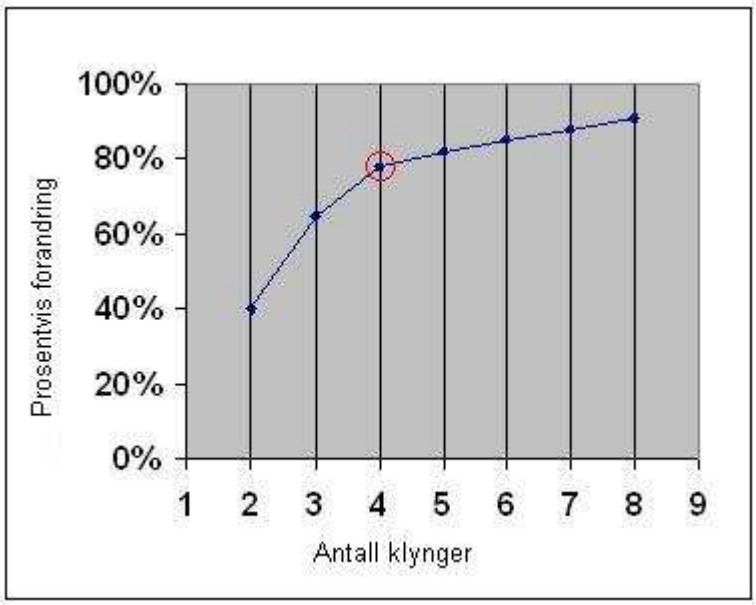
6.5 Antall klynger

For algoritmer hvor man må bestemme antall klynger dukker spørsmålet ”hvor mange klynger bør man velge?” opp. For eksempel ved bottom-up hierarkisk klyning, må man på et tidspunkt bestemme når man skal stoppe itereringen slik at man har et visst antall klynger (om itereringen ikke stoppes vil den fortsette til det bare er en klynge igjen). For å bestemme dette kan man bruke en regel, albueregelen, som beskrevet under.

6.5.1 Albueregel (Elbow criterion)

Albueregelen, eller “Elbow criterion” er en vanlig regel for å bestemme antall klynger som bør bli valgt, for eksempel ved bruk av k-means-algoritmen eller bottom-up hierarkisk klynging. Denne regelen sier at man bør velge et antall klynger slik at om man legger til en ekstra klynge blir det ikke lagt til tilstrekkelig informasjon. Hvis man tegner en graf over forandringene ved at man legger til klynger vil første klyngen legge til mye informasjon, men etter et visst punkt vil dette avta, og man får en vinkel (en knekk) i grafen (albuen). Dette er

vist i Figur 11. Her ser en at 4 er det antallet klynger man bør velge etter denne regelen, fordi det er her grafen får en knekk.



Figur 11: Albueregel.

7 Vår modell

Vår gruppe har valgt å basere seg på den konstruktivistiske tankegangen (beskrevet i seksjon 2.3). Med vår gruppe menes som tidligere nevnt alle mastergradsstudentene som har amanuensis Arvid Holme som veileder.

Begrunnelsen for å basere seg på den konstruktivistiske modellen, er at vi mener at det er den lærende som aktivt skal bygge opp sin kunnskap når det er ønskelig eller etter behov. For at dette skal bli mest mulig optimalt, er man avhengig av at det er en indre motivasjon som er til stede, slik at den lærende selv har lyst til å lære og skape kunnskap. Vi ønsker å lage et rammeverk slik at det blir enklere både for den lærende og læreren å organisere kunnskapen samt å skape egen kunnskap i form av egne lærings- og kunnskapsobjekter. Målet er altså å også kunne skape og jobbe med både kunnskapsobjekter og læringsobjekter i et slikt rammeverk, men ettersom læringsobjektene har momenter som læringsmål og læringsaktiviteter i seg (se seksjon 4.3), krever dette mer pedagogisk kompetanse. Det vil også ikke være like lett å indeksere disse læringsobjektene da en ikke enkelt kan lage en tekstlig beskrivelse av disse på samme måte som vi skal se at man kan med kunnskapsobjekt. Dette fordi læringsobjektene som tidligere nevnt er mye mer komplisert og har andre ting enn bare ressurser knyttet til seg. Vi har derfor valgt å begrense området til å jobbe med kunnskapsobjekter.

Først i dette kapitlet blir vår målsetting beskrevet. Videre følger en beskrivelse av kunnskapsobjektene slik vi tenker oss dem og en beskrivelse av et framtidig rammeverk. Dette rammeverket innebærer indeksering av kunnskapsobjekter, organisering av objektene ved bruk av leksjonsmodellen eller arbeidsbok samt presentasjon av objektene ved bruk av klyngeanalyse og et tredimensjonalt konseptkart. Til slutt følger en kort beskrivelse av bruk av latent semantisk indeksering som det også jobbes med i vår gruppe.

7.1 Vår målsetting

Intensjonen er at når en lærende er ferdig med et kurs skal den lærende sitte igjen med sin egen kunnskapsbase. Denne basen er det derfor viktig at den lærende selv bygger opp på sin måte, fordi folk har veldig forskjellig måte å lære på og måten man ønsker å organisere kunnskapen på vil sannsynligvis variere veldig fra person til person.

Vi ønsker å følge en konstruktivistisk modell og gjøre læringen bedre i form av en mer tilpasset og interaktiv læring hvor den lærende er i hovedfokus. Dagens læringssystemer ser stort sett på læring som en avsluttende prosess, dvs. at den lærende ikke sitter igjen med noe håndfast når et kurs er avsluttet. De er utviklet slik at når et kurs er avsluttet er også læringsprosessen avsluttet. Vi ønsker å se på læring som en viktig del av kunnskapsforvaltning, hvor læring er den delen som produserer ny kunnskap. (Holme 2006).

Vårt mål er derfor å lage et system for kunnskapsforvaltning hvor nettbasert læring inngår som en del for produksjon av individuell kunnskap som kan organiseres etter eget ønske. For å oppnå dette målet, benytter vi i vår gruppe metoder fra fagområdene informasjonsgjenfinning, matematikk, visualisering og kunstig intelligens. Vi ønsker ikke å utvikle en ny læringsplattform, men et opplegg av komponenter som kan benyttes i allerede eksisterende læringsplattformer.

7.2 Kunnskapsobjektene

For å kunne jobbe med kunnskapsobjekter må man vite hvordan de skal representeres og gjort klar for indeksering. I etterfølgende seksjoner er dette nærmere beskrevet. I tillegg blir det sett nærmere på hvordan en ser for seg bruk og deling av disse objektene ved bruk av repositories.

7.2.1 Beskrivelse av kunnskapsobjektene

De fleste dokumenter og tekstsamlinger har metadata knyttet til seg. Metadata er som nevnt tidligere i oppgaven informasjon om organiseringen av dataen (i dokumentet),

domeneavhengighet og relasjonene mellom dem. Veldig mange mener at metadata er det som bør brukes og satses på i forbindelse med informasjonsgjenfinning. Man ønsker en mal for å kunne beskrive alle dokumenter og man benytter standarder for å beskrive alle dokumentene på samme måte (dvs. sette opp metadataene). Dette blir så brukt i søkesammenheng og det er i disse feltene man søker etter dokumentene.

Vi mener derimot at dette ikke er veien å gå. Det viser seg at metadata er veldig subjektivt. Disse dataene vil derfor variere veldig avhengig av hvem som skriver dem og også når det blir gjort. Det er undersøkelser som viser at selv samme person kan skrive forskjellige metadata om samme dokument til ulike tidspunkt (Holme 2000). Dette fordi metadata vanligvis er noe som må gjøres manuelt, og ofte vil det være vanskelig å fylle inn etter en bestemt mal, fordi man ikke føler det passer eller vet hva man skal fylle inn. Det kan være at dette er noe som blir oppfattet som vanskelig og tidkrevende og derfor noe som blir gjort veldig unøyaktig (ved at man bare fyller inn noe veldig raskt uten å tenke særlig over det). Dette kan føre til veldig dårlig metadata som igjen kan føre til at de er ubrukelige i en informasjonsgjenfinningssammenheng.

Derfor er et av målene vi jobber etter, det at en bruker skal kunne sette seg ned og skrive i naturlig språk hva som er ønskelig. For eksempel vil en lærer utarbeide et kurs og ønsker hjelp til å utarbeide en del som om handler et bestemt tema. Læreren kan da skrive en beskrivelse av temaet, hvilke temaer og konsepter som ønskes og hva man ønsker å oppnå med kurset. Systemet skal da kunne gi tilbake kunnskapsobjekter som forhåpentligvis passer med det læreren var på jakt etter.

7.2.2 Tekstlig beskrivelse av kunnskapsobjektene

For å kunne søke blant kunnskapsobjektene, må man ha noe å søke i som identifiserer objektene. I veldig mange søkemotorer søker man direkte i teksten i dokumentene. Det er da veldig vanlig å benytte fritekstsøk (beskrevet i seksjon 5.2.4). Å søke direkte i et dokument vil ofte fungere veldig greit dersom det er snakk om rene tekstdokumenter eller tekstdokumenter som består for det meste av ren tekst. Men, har man dokumenter eller objekter bestående av andre elementer som for eksempel bilde, video, animasjon, lyd og lignende har man kanskje ikke en slik tekst å søke i, eller den vil være veldig mangelfull som identifikator på objektet.

Slik som våre kunnskapsobjekt er oppbygd vil de sannsynligvis ofte inneholde andre elementer enn ren tekst, eller det som er kalt ressurser (beskrevet i seksjon 4.1). Ettersom objektene da ofte kan inneholde lite eller i noen tilfeller ingen tekst, vil dette være en mangelfull måte å identifisere objektet på, og dermed er fritekstsøk direkte på objektet uaktuelt. En kunne gjort som mange andre systemer gjør, og brukt metadata for objektene som identifikatorer, men som nevnt tidligere mener vi denne metoden er utilstrekkelig og fører ofte til en subjektiv identifisering.

For at man likevel skal kunne identifisere kunnskapsobjektene, slik at det blir mulig å søke på dem er det bestemt at til hvert kunnskapsobjekt skal det være en tekstlig beskrivelse av objektet. Denne skrives manuelt og vil være fra en halv til en a4-side cirka. Ved å bruke en slik beskrivelse får man en unik beskrivelse av hvert kunnskapsobjekt og alle ressursene de består av. Hvis det for eksempel er et kunnskapsobjekt som er en animasjon, har man en tekstlig beskrivelse av den. Denne beskrivelsen kan sammenlignes med det en lærer ville sagt dersom en animasjon ble demonstrert for en lærende. Intensjonen med dette er ikke å påføre noen ekstraarbeid med en slik beskrivelse, men siden en lærer mest sannsynlig ville lagt slike beskrivelser til bruk i undervisning uansett, medfører dette ikke ekstraarbeid. For eksempel til et lysark vil veldig mange lærere ha egne stikkord og tekst. Dette er teksten de sier til den lærende. Selve lysarket inneholder ofte bare stikkord, korte tekster, setninger og bilder.

7.2.3 Vektorisering av kunnskapsobjekter

Som nevnt i forrige seksjon, er det bestemt at til et hvert kunnskapsobjekt er det knyttet en følgetekst. Denne Følgeteksten gjør det mulig å framstille kunnskapsobjektet som en vektor. I vårt system vil altså alle kunnskapsobjekter bli framstilt som vektorer i et *n*-dimensjonalt vektorrom. *n* angir antallet forskjellige termer som finnes i det bestemte domenets termmengde. Denne termmengden blir bestemt ved at alle følgetekstene til kunnskapsobjektene blir lest og man henter ut alle unike termer som man ønsker å ha med i indekseringen (stoppord blir mao. fjernet). Når man så har disse kunnskapsobjektene representert som en vektor kan man som nevnt tidligere bruke statistiske metoder for å organisere disse. Her kan man bruke redskap som lineær algebra, nevrale nett, klyngeanalyse osv. (Holme 2006).

7.2.4 Repository

Repository er samlinger av kunnskapsobjekter. Som nevnt tidligere i oppgaven (seksjon 3.1.4), ser vi for oss at kunnskapsobjektene blir produsert i en produksjonslinje av profesjonelle aktører. Når kunnskapsobjektene da er produsert vil det være samlinger med disse, repositories hvor de ligger tilgjengelige. Man kan tenke seg at både faglig veileder samt de lærende kan abonnere på tilgang til en slik database, eller repository. Faglig veileder skal kunne søke i dette for å lage læringsobjekter, og de lærende for å kunne løse oppgaver.

Dette repositoriet vil da altså inneholde ferdige kunnskapsobjekter, som abonnenter kan jobbe med og bruke som en selv ønsker. Om abonnentene kun jobber med referanser til objektene uten å selv laste de ned, vil man kunne opprettholde man regler for opphavsrett, kopibeskyttelse osv.

7.3 Rammeverket

Det ønskes som nevnt tidligere, et helhetlig rammeverk for organiseringen av kunnskapsobjektene i en læringssammenheng. Dette rammeverket skal kunne brukes av både lærer og den lærende, men hovedfokus er kunnskapsorganiseringen fra den lærendes ståsted. Det finnes veldig mange læringsplattformer i dag (it's:learning, ClassFronter, FirstClass, PedIT osv.), men det som er felles for de fleste er at de er organisert ut fra lærerens ståsted, og oppdelt i kurs. En lærende vil sannsynligvis ikke ha samme behov for å dele inn kunnskapen etter kurs, fordi det kan være emner innenfor ulike kurs som omhandler samme tema (helt eller delvis overlapper), og som det derfor er ønskelig å knytte sammen på en annen måte, enn hva som er gjort i kursene eller på tvers av kurs.

Det blir her først sett på indeksering av kunnskapsobjektene, organisering av objektene samt presentasjon av objektene.

7.3.1 Indeksering av objektene: Vektormodellen

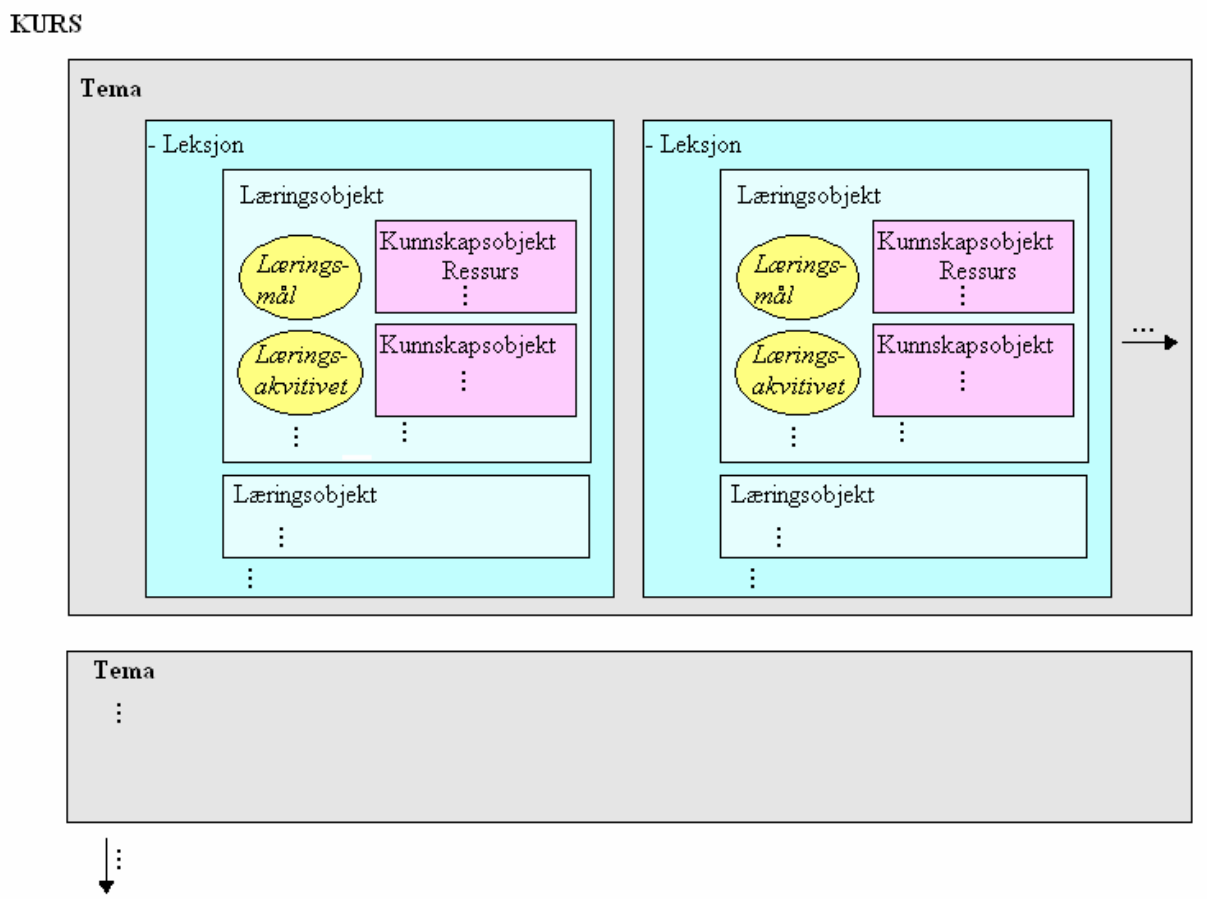
Vår gruppe har basert seg på vektormodellen for gjenfinning av kunnskapsobjektene. Dette fordi dette er den modellen som passer best til vårt formål. Som nevnt tidligere, er det påpekt

at det til et hvert kunnskapsobjekt er knyttet en følgetekst. Det er denne teksten som igjen lar seg vektorisere, slik at alle objektene blir framstilt som vektorer. Vektormodellen har den egenskapen at den kan rangere dokumenter ut fra likhet og dermed gjør det mulig å organisere søkeresultatet på en bedre måte enn for eksempel ved bruk av en ren boolsk modell (som bare sier om et dokument er relevant eller ikke, men ikke noe om *i hvor stor grad* et dokument er relevant eller ikke). Vektormodellen sies derfor ofte å være den ”beste” søkemodellen, men i og med at den er så ressurskrevende er den ikke så utbredt, fordi søkemotorer ofte er nødt til å behandle ekstremt store mengder data (som for eksempel ved et søk på Internett). Til vårt system derimot, som er tenkt domeneavhengig, vil det være snakk om mindre datamengder og dermed kan vektormodellen og dens sterke sider benyttes.

7.3.2 Organisering av objektene

Leksjonsmodellen

Denne modellen er under utvikling i vår gruppe og ser på organisering av læringsmateriell fra et kurs ståsted. Her deles læringsmateriellet opp i kurs som består av et eller flere tema. Temaer består av en eller flere leksjoner. Hver leksjon består av et tema som igjen består av en eller flere læringsobjekt. Læringsobjekt består av et eller flere kunnskapsobjekt tilknyttet læringsmål og ulike læringsaktiviteter. Læringsobjekt og kunnskapsobjekt er nærmere beskrevet i seksjon 4. En oversikt over hvordan denne leksjonsmodellen er tenkt er vist i Figur 12.



Figur 12: Leksjonsmodellen.

Fordelen med en slik organisering, er at en faglig veileder kan bygge opp et kurs med allerede eksisterende materiell. Det kan dermed hentes inn hele ferdige temaer fra et repository (beskrevet i seksjon 7.2.4) i stedet for å bruke tid på å konstruere alt på nytt. Dette kan være temaer fra andre lignende kurs, eller tidligere kurs som samme veileder har holdt. Dersom det er ønskelig å bygge opp et undervisningsopplegg for et tema kan leksjoner hentes inn eller lages på nytt.

Om leksjoner skal konstrueres helt fra bunnen av, må en ved denne modellen først hente inn ferdiglagde eller selvlagde kunnskapsobjekt og ressurser. Læringsobjekt konstrueres som nevnt tidligere ved å knytte kunnskapsobjekter sammen med læringsmål og læringsaktiviteter. Det er som nevnt tidligere disse læringsobjektene en leksjon består av. Med denne modellen oppnår en altså at kunnskapen blir inndelt i objekter og dermed lar seg gjenbruke og knyttes sammen på ulike måter uten behov for mye tilpasning.

Arbeidsbok

En arbeidsbok som også er under utvikling i vår gruppe, er tenkt til å være et miljø sett fra den lærendes ståsted. Her skal den lærende selv kunne organisere og jobbe med læringsmateriell og samtidig utvikle stoff selv. Man tenker her at læringsmateriellet, dvs. kunnskapsobjektene skal kunne organiseres på tvers av kurs og sammensetninger av faglig veileder. På denne måten kan en lærende lage nye koblinger og knytte sammen deler fra ulike kurs etter ønske.

Man tenker seg at en slik arbeidsbok skal jobbe med referanser. Det er videre tenkt at det hele tiden jobbes opp mot Internett, slik at man kan se på, og jobbe med objekter som ligger tilgjengelig i ulike repositories, eller man kan utvikle egne objekter selv dersom det er ønskelig.

7.3.3 Presentasjon av objektene

Klyngeanalyse

Klyngeanalyse gjør det mulig å se på sammenhengen mellom objektene, se hvordan de er relatert til hverandre og hvor nærme de ligger hverandre (likhet); for eksempel om de omhandler samme tema. Resultatene fra en slik klyngeanalyse er et veldig godt utgangspunkt for å gi en grafisk framstilling av en organisering av kunnskapsobjektene, eller et søk. Ved å bruke tredimensjonale konseptkart vil brukeren få helt nye muligheter og forhåpentligvis bedre måter å organisere kunnskapen og læringsmateriellet på.

Tredimensjonalt konseptkart

Vektormodellen gir muligheter for grafisk organisering av kunnskapsobjekter. I det endelige systemet skal det benyttes en tredimensjonal graf hvor nodene er kunnskapsobjekter og kantene viser hvor sterkt de foreskjellige kunnskapsobjektene er knyttet til hverandre. Kantene vil med andre ord markere likheten mellom objektene. Nodene i grafen kan enten representere enkelte kunnskapsobjekter eller klynger av kunnskapsobjekter. Om en node representerer en klynge, skal man da kunne klikke seg inn i noden og få opp en ny himmel med den nye klyngen. Denne nye klyngen kan bestå av noder som representerer kunnskapsobjekt, eller igjen nye klynger av objekter. Man vil altså her ha en hierarkisk

framstilling hvor brukeren kan klikke seg fram etter ønske. En klynge som er en samling av kunnskapsobjekter med stor likhet, er et konsept.

En slik tredimensjonal graf kan vise hvordan alt læringsmateriellet i et kurs er organisert, det vil si hvordan de forskjellige kunnskapsobjektene eller konseptene er forbundet med hverandre. Ikke bare fra et kurs ståsted, men også fra en lærendes ståsted skal denne framstillingen kunne benyttes. Man kan tenke seg at en lærende vil organisere det som er lagd i forbindelse med et kurs, og/eller knytte dette sammen med læringsmateriellet fra et eller flere kurs. En lærende skal altså selv kunne organisere det som er lært på sin egen unike måte. Flere lærende vil da sannsynligvis sitte igjen med helt forskjellige organiseringer av læringsmateriellet når et kurs er ferdig. Ettersom det er en kjent sak at ulike personer lærer på ulike måter og ved bruk av ulike metoder, er det viktig å ta vare på denne individualiteten, og derfor bør en kunne organisere og framstille læringsmateriellet som man selv vil. Et tredimensjonalt konseptkart vil være med på å gjøre læringen mer individtilpasset.

Grafen skal også kunne benyttes for å representere resultat fra et søk i en gitt samling av objekter på en oversiktlig måte. Spørringen (queryet) kan representeres som en node i grafen og plasseres i sentrum av grafen. Resultatnodene (de nodene som ga treff på søket) vil da orienteres rundt spørringen etter hvor nære de ligger i likhet. I motsetning til resultatet fra en vanlig søkemotor som representerer resultatene i en sekvensiell liste, gir den grafiske presentasjonen oss muligheten til å se hvordan kunnskapsobjektene som returneres ikke bare forholder seg til spørringen, men også til hverandre. (Holme 2006).

7.4 Latent semantisk indeksering

Et problem som ofte oppstår i læringssammenheng er bruk av fagterminologi. Ofte er det slik at det blir brukt veldig mye fagtermer som en novise vil ha problemer med å forstå eller å bruke. Det vil da bli problematisk for brukeren å søke i et domene ettersom brukeren ikke kjenner så godt til fagtermene som blir brukt. Vår gruppe jobber i denne forbindelse med å kunne overkomme dette problemet ved å benytte latent semantisk indeksering (LSI). LSI plasserer dokumenter og termer i et vektorrom som vanligvis har færre dimensjoner enn fagdomenets vektorrom. Det man da oppnår er at dokumenter som er sterkt assosiert med

hverandre vil få relativ lik vektorrepresentasjon i det nye rommet. Det blir foretatt en dimensjonsreduksjon av det opprinnelige vektorrommet, det vil si at en spørring kan benytte andre termer enn de som ble brukt ved indeksering av kunnskapsobjektene, men samtidig gi treff på objekter som er relevante.

LSI er også noe det endelige systemet skal ha med for å bedre kunne tilrettelegge læringen for den lærende. Særlig i startfasen av en læringsprosess vil dette sannsynligvis være meget nyttig, da en lærende ikke er så kjent med fagtermene som blir brukt i emnet. Målet er altså å knytte LSI til systemet slik at en novise enkelt kan kommunisere med sitt eget språk med systemet. Systemet fungerer da som en slags ”oversetter” mellom novise og fagspråk.

8 Prototyp og implementering

I dette kapitlet blir det først sett på hvilken preprosessering som er nødvendig for å utføre en klyngeanalyse, hvilke innputt prototypene er avhengige av, hvordan objektene blir representert ved hjelp av vektorer og dokument-term-matrise, likhetsmål, kort om hvordan resultatene av analysen blir presentert i prototypene samt en oversikt over implementeringen ved hjelp av klassediagram med forklaring.

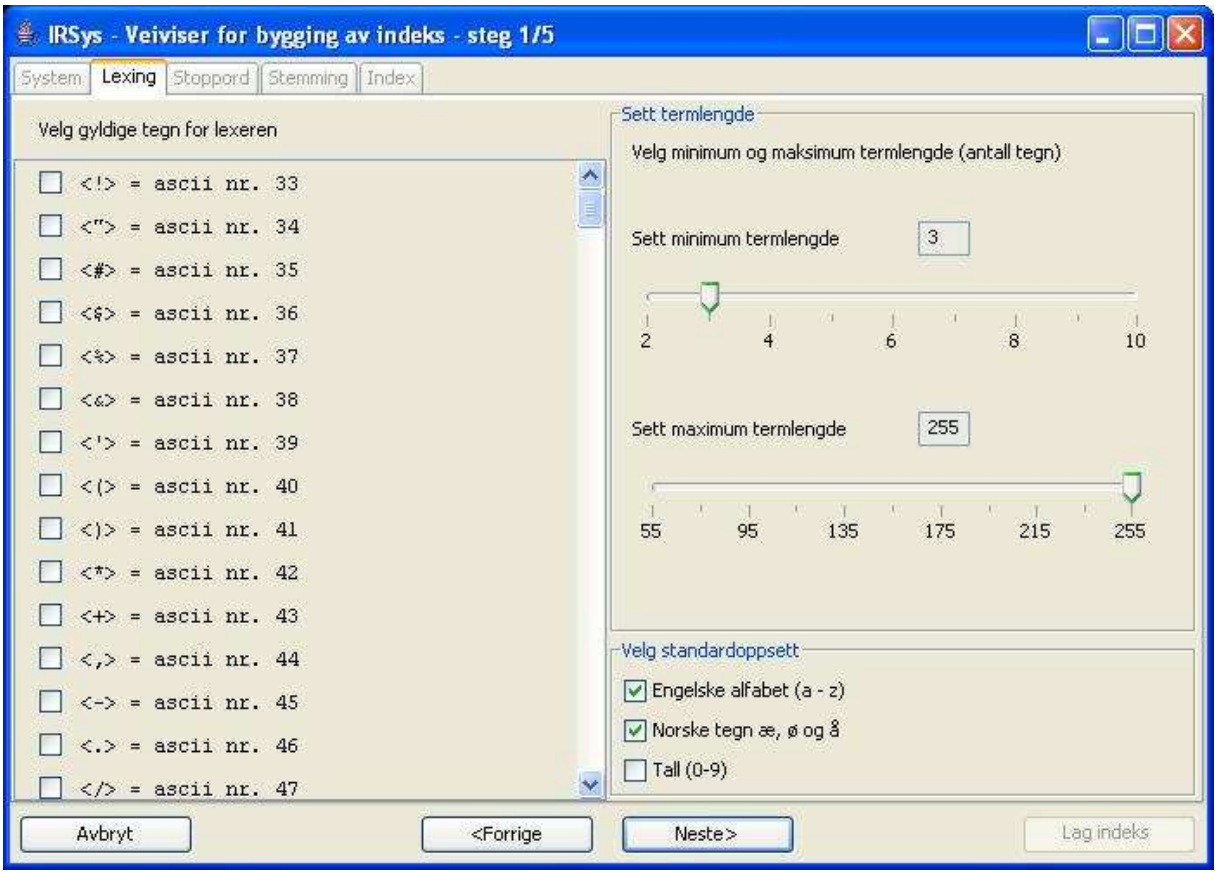
8.1 Preprosessering

Kunnskapsobjektene som skal klynges, dvs. de tekstlige beskrivelsene av objektene går først igjennom en leksikalsk analyse og stoppordfjerning før de er klare for selve klyngeanalysen. Dette for at objektene skal bli behandles på samme måte, og fordi man får en bedre indeksering av hvert objekt. Det er viktig at man bruker samme stoppordliste for både objektene som skal klynges samt de initielle klyngerepresentantene.

Denne delen er i de tilhørende prototypene for denne oppgaven ikke blitt implementert, men bruker en applikasjon utviklet tidligere som samarbeidsprosjekt i vår gruppe. Grunnen til dette er hovedsaklig at man ønsker et modulært system bestående av frittstående komponenter slik at man senere skal kunne knytte dette sammen slik man måtte ønske. Det ønskes som nevnt tidligere i oppgaven ikke å lage et system hvor alt avhenger av hverandre. Modulariteten vil føre til at man enklere kan inkludere komponentene i ulike systemer.

Applikasjonen for leksikalsk analyse og stoppordfjerning leser tekstfiler, utfører leksikalsk analyse og produserer nye tekstfiler. Disse nye tekstfilene blir så lest og utført stoppordfjerning på før det igjen produseres nye filer. Det er i testingen av klyngeanalyse valgt å ikke utføre stemming på de tekstlige beskrivelsene av objektene som skal klynges, da denne funksjonaliteten er vurdert til å kreve en uforholdsmessig stor mengde ressurser for å få til å fungere optimalt i den eksisterende applikasjonen, sammenlignet med prioritert funksjonalitet. I et endelig system som skal benyttes for vanlige brukere vil dette selvfølgelig være en modul som burde inngå, men dette er ikke en kritisk del som må være med for å kunne se resultater av en klyngeanalyse.

I denne applikasjonen, skrevet hovedsaklig av Per Kristen Fredlund, må brukeren først angi hvilke tekstfiler som skal gjøres klare for indeksering. Når dette er gjort må det angis innstillinger for leksikalsk analyse. Skjermbilde på denne delen er vist i Figur 13.



Figur 13: Skjermbilde av valg for leksikalsk analyse.

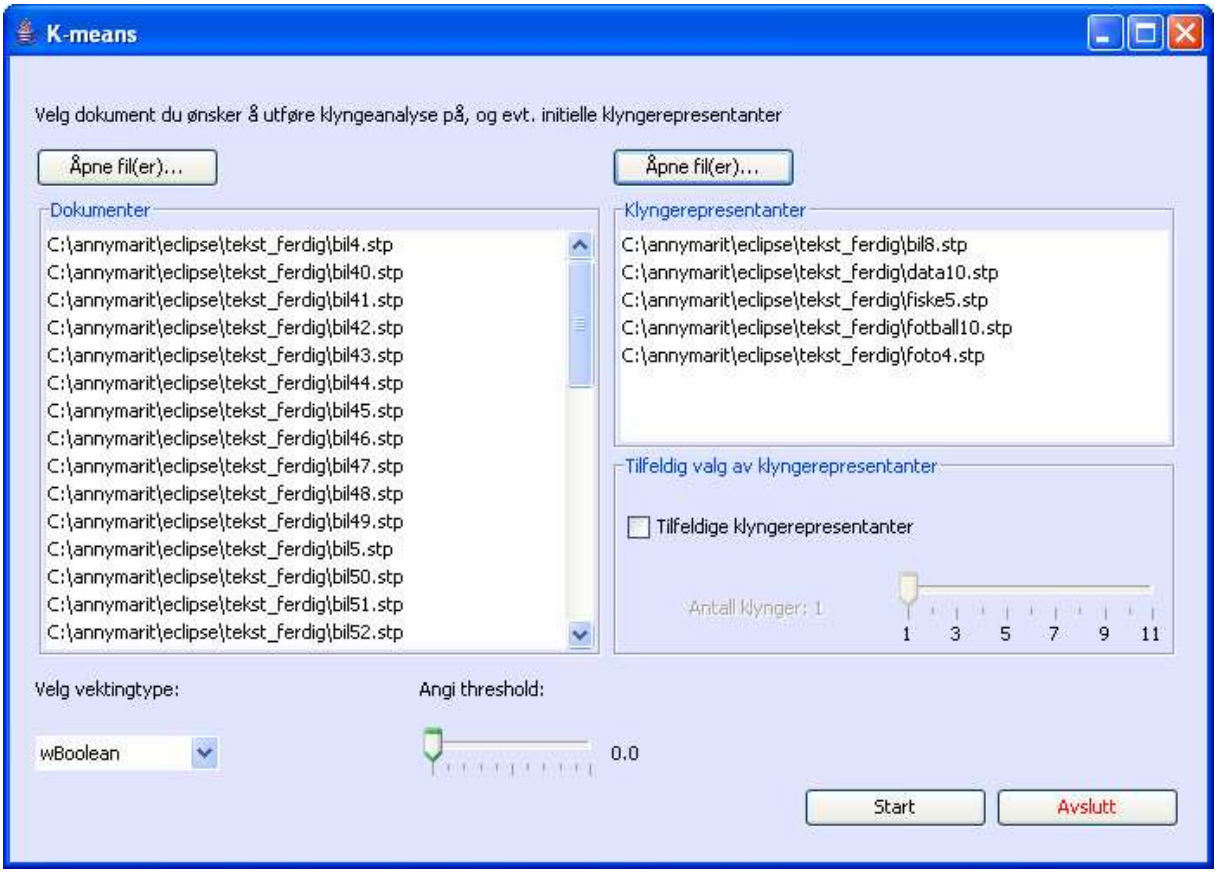
Her må brukeren angi hvilke termer som skal være gyldige, samt minimum og maksimum termlengde. Når dette er gjort, må brukeren velge stoppordliste dersom det skal benyttes. Det er i denne testingen, jobbet med norske filer og har derfor valgt en forhåndsdefinert stoppordliste. Stemming blir som nevnt ikke utført. Til slutt i denne applikasjonen blir det bygget en indeks. Denne indeksen blir i prototypene for denne oppgaven ikke brukt, det er bare brukt tekstfilene som blir lagd etter at leksikalsk analyse og stoppordfjerning er utført.

8.2 K-means

8.2.1 Innputt

Brukeren må velge hvilke kunnskapsobjekter som skal klynges ved å åpne de tilhørende beskrivelsene av objektene. Disse beskrivelsene er lagret som rene tekstfiler.

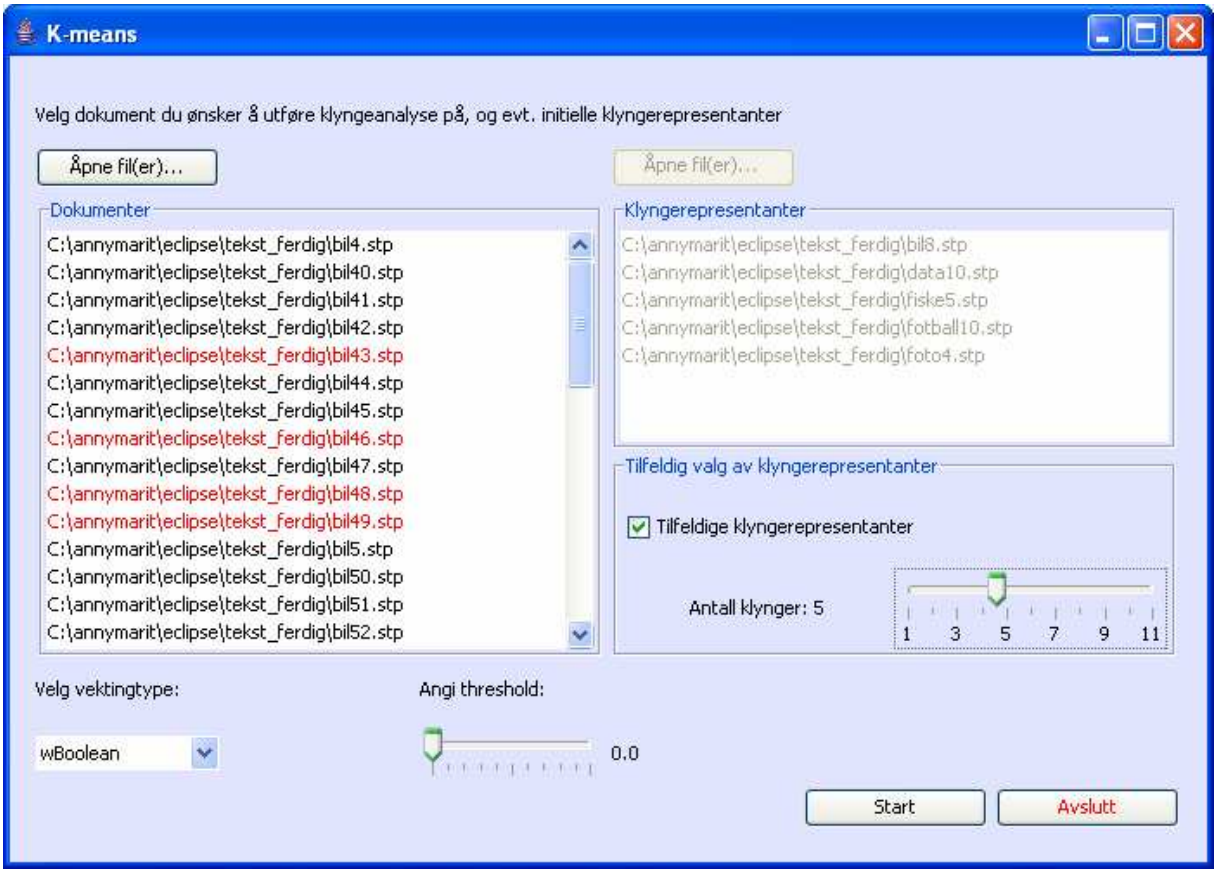
Prototypen for klyngeanalyse ved bruk av K-means er implementert slik at brukeren kan velge mellom å la systemet velge initielle klyngerepresentanter tilfeldig, eller at brukeren selv velger hvilke klyngerepresentanter som skal være de initielle. Velger brukeren å selv bestemme initielle klyngerepresentanter, disse filene åpnes på samme måte som selve filene som skal klynges. Eksempel på brukervalgte initielle klyngerepresentanter er vist i Figur 14.



Figur 14: Skjermbilde for K-means med initielle klyngerepresentanter valgt av bruker.

Om brukeren derimot velger å la systemet velge initielle klyngerepresentanter tilfeldig, må brukeren gi inn antall klynger det er ønskelig å klynge samlingen i forhold til. Systemet vil så

følge vanlig K-means algoritme og iterativt klynge objektene helt til klyngerepresentantene ikke lenger forandres. Eksempel på skjermbilde hvor dette blir gjort er vist i Figur 15.



Figur 15: Skjermbilde for K-means med initielle klyngerepresentanter tilfeldig valgt av systemet.

Denne prototypen på K-means skiller seg altså fra den vanlige K-means-algoritmen ved at brukeren selv kan velge initielle klyngerepresentanter dersom ønskelig. Dette kan gjøres fordi som nevnt tidligere i oppgaven, en antar at brukeren har en viss kjennskap til objektsamlingen. Brukeren vil sannsynligvis ha en viss formening om hvilke emner samlingen kan grupperes etter. Det blir senere sett på at testing viser at man får et bedre resultat av klyngeanalysen dersom man selv velger hvilke objekter som skal være initielle klyngerepresentanter.

8.2.2 Dokument-term-matrise

Ved bruk av K-means-algoritmen på tekstdokumenter, må alle dokumenter som skal klynges være representert som vektorer. Man vil da ha en dokument-term-matrise hvor hver term er listet opp med vektene i alle dokumentene, enten termen befinner seg i dokumentet eller ikke

(befinner termen ikke seg i dokumentet vil vekten være 0). Denne matrisa er vist i tabellen under.

Tabell 1: Dokument-term-matrise.

	dok_1	dok_2	dok_3	…	dok_m
t_1	w_{11}	w_{12}	w_{13}	…	w_{1m}
t_2	w_{21}	w_{22}	w_{23}	…	w_{2m}
t_3	w_{31}	w_{32}	w_{33}	…	w_{3m}
…	…	…	…		…
t_n	w_{n1}	w_{n2}	w_{n3}	…	w_{nm}

Her er dok_j dokument nummer j, t_i term nummer i, w_{ij} vekt som angir indeksterm t_i sin betydning for dokument d_j. Vekten w_{ij} angir om et dokument inneholder en term eller ikke. Dvs. dok_i inneholder t_j (da vil $w_{ij} = 1$) eller ikke (da vil $w_{ij} = 0$). Vektene kan beregnes etter et gitt kriterium (se seksjon 5.3).

Dette vil si at for hvert dokument har man en vektor hvor alle vektene til hver term befinner seg. Som beskrevet tidligere krever algoritmen for K-means (se seksjon 6.1.1) at man må sammenligne vektorer for å finne likheter slik at de kan bli tilordnet til en klynge. Dokumentvektorene må sammenlignes med klyngerepresentantene. Vektoren for en klyngereepresentant kan noteres slik;

$q = (r_1, r_2, r_3, \ldots r_n)$.

8.2.3 Liketsmål

Som likhetsmål er det i denne prototypen benyttet cosinus-similaritet. Denne formelen angis som vist i seksjon 5.2.2 slik:

$$sim(q, d_j) = \frac{\sum_{k=1}^{n} w_{kj} r_k}{\sqrt{\sum_{k=1}^{n} w_{kj}^2 \sum_{k=1}^{n} r_k^2}}.$$

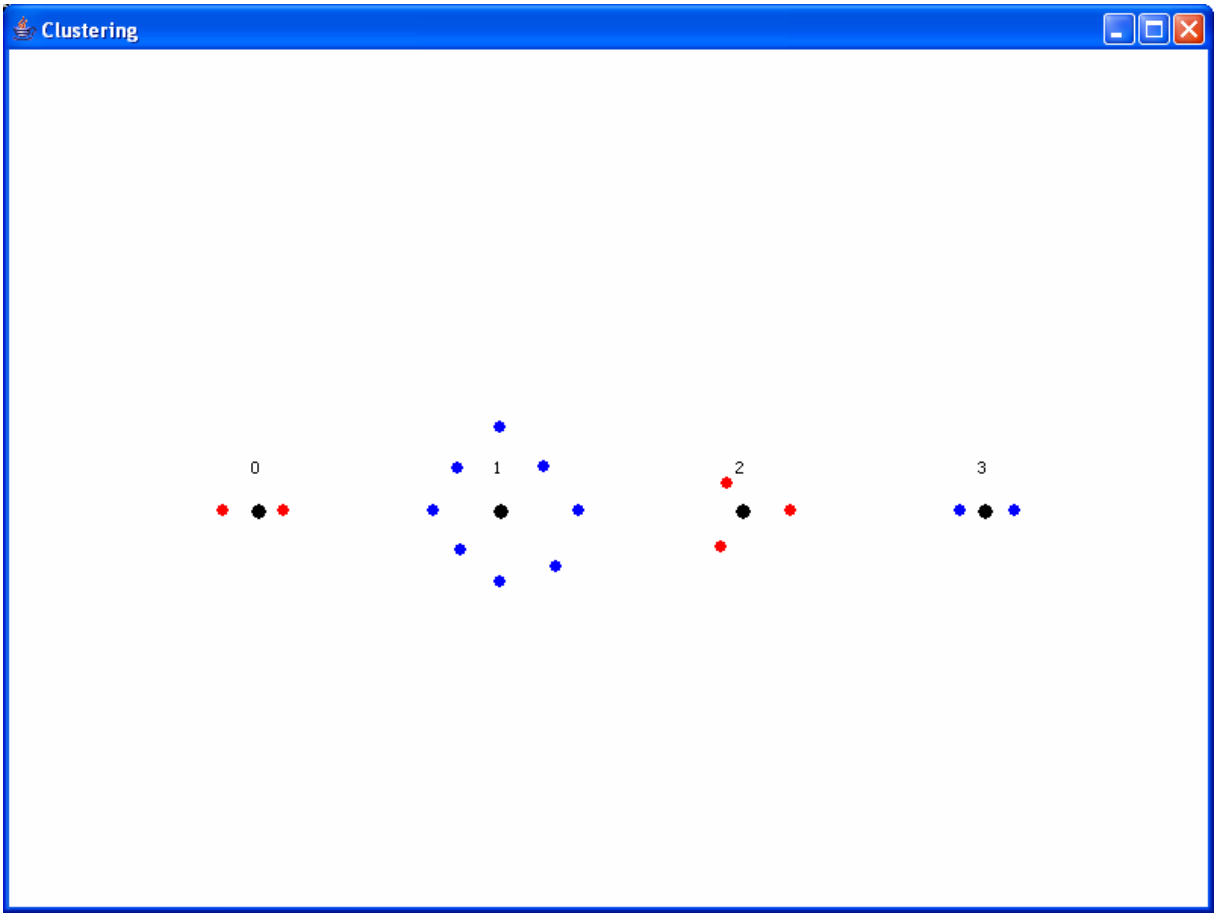
Her er det altså vektorene q (angir klyngerepresentanten) og d_j (den aktuelle vektoren i dokumentsamlingen) som sammenlignes. Dette cosinusmålet angir størrelsen på vinkelen mellom to vektorer. Denne størrelsen varierer fra 0 til 1;

$$0 \leq \text{sim}(q, d_j) \leq 1.$$

Dette vil si at jo nærmere to vektorer er hverandre, jo mindre blir vinkelen mellom dem og jo større blir cosinusmålet. Om to vektorer er identiske med hverandre blir coninusmålet lik 1.0. For to vektorer som ikke har noen likhet vil de stå vinkeltrett på hverandre og ha cosinusmål lik 0.

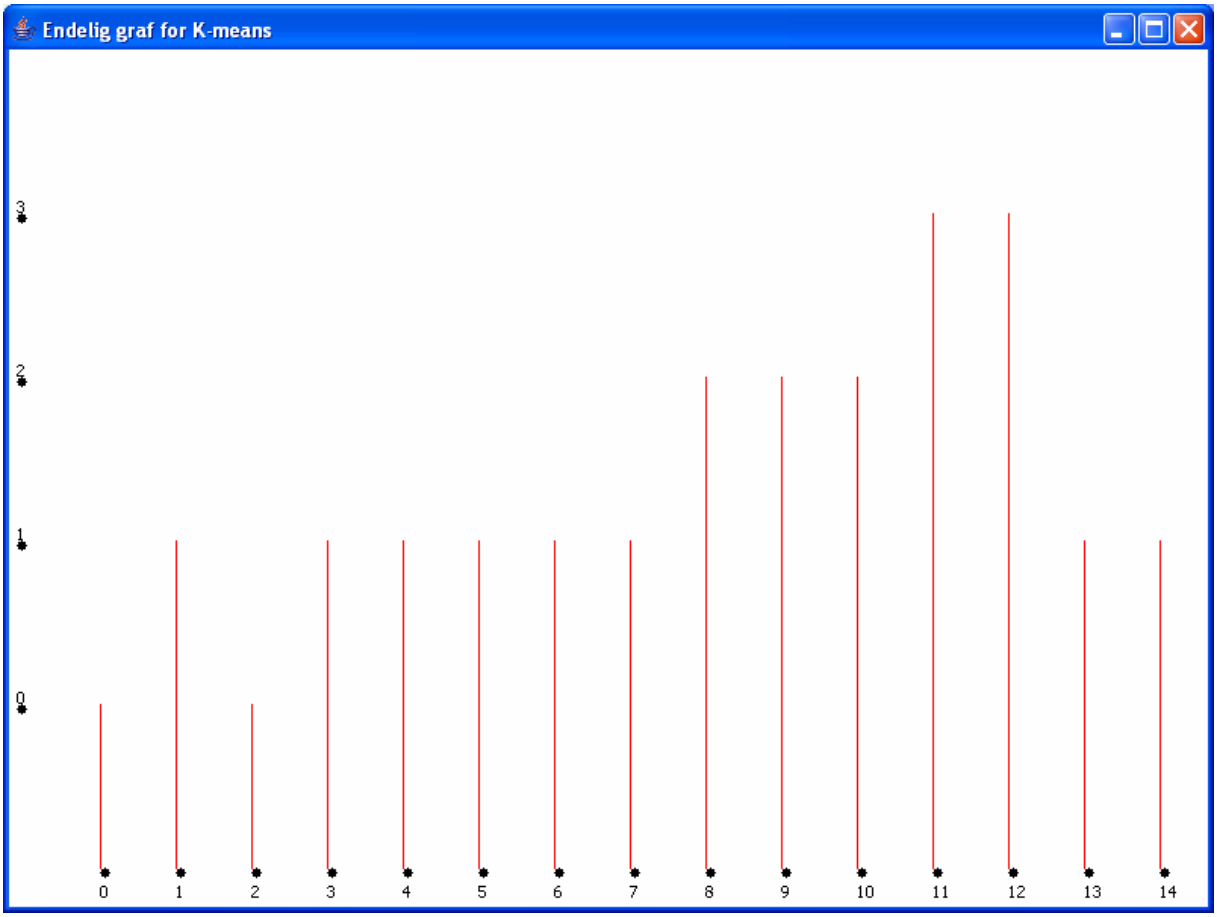
8.2.4 Presentasjon grafisk

Programmet for K-means vil under kjøring hele tiden oppdatere grafisk hvordan klyngefordelingen ser ut. Dette blir gjort ved å la alle objektene være representert som noder i en todimensjonal graf. Klyngerepresentantene blir tegnet opp midt i skjermbildet på en linje og punktene (vektorene) fordeler seg rundt de respektive klyngerepresentantene. Hver node vil altså tilhøre en klyngerepresentant. Avstanden inn til sentrum (klyngerepresentanten) angir similariteten til den aktuelle klyngerepresentanten. De nodene som blir liggende langt unna representanten har altså lav likhet med klyngerepresentanten, mens de helt nære har stor likhet. Likhet 1.0 som er maksimal likhet, dvs. vektorene for objektene står vinkelrett på hverandre og er helt like vil bli tegnet midt over klyngerepresentantnoden. Denne representasjonen er vist som et eksempel i Figur 16. Her er det et utsnitt fra en kjøring med fire klynger hvor initielle klyngerepresentanter er valgt av bruker. Fargene for nodene varierer (annenhver rød og blå) for å kunne skille hvilke noder som tilhører hvilken klynge.



Figur 16: Klyngefordeling ved bruk av K-means.

I tillegg til denne grafiske framstillingen som oppdateres underveis, blir det tegnet opp en graf som viser resultatet av klyngefordelingen.



Figur 17: Eksempel på endelig graf for K-means.

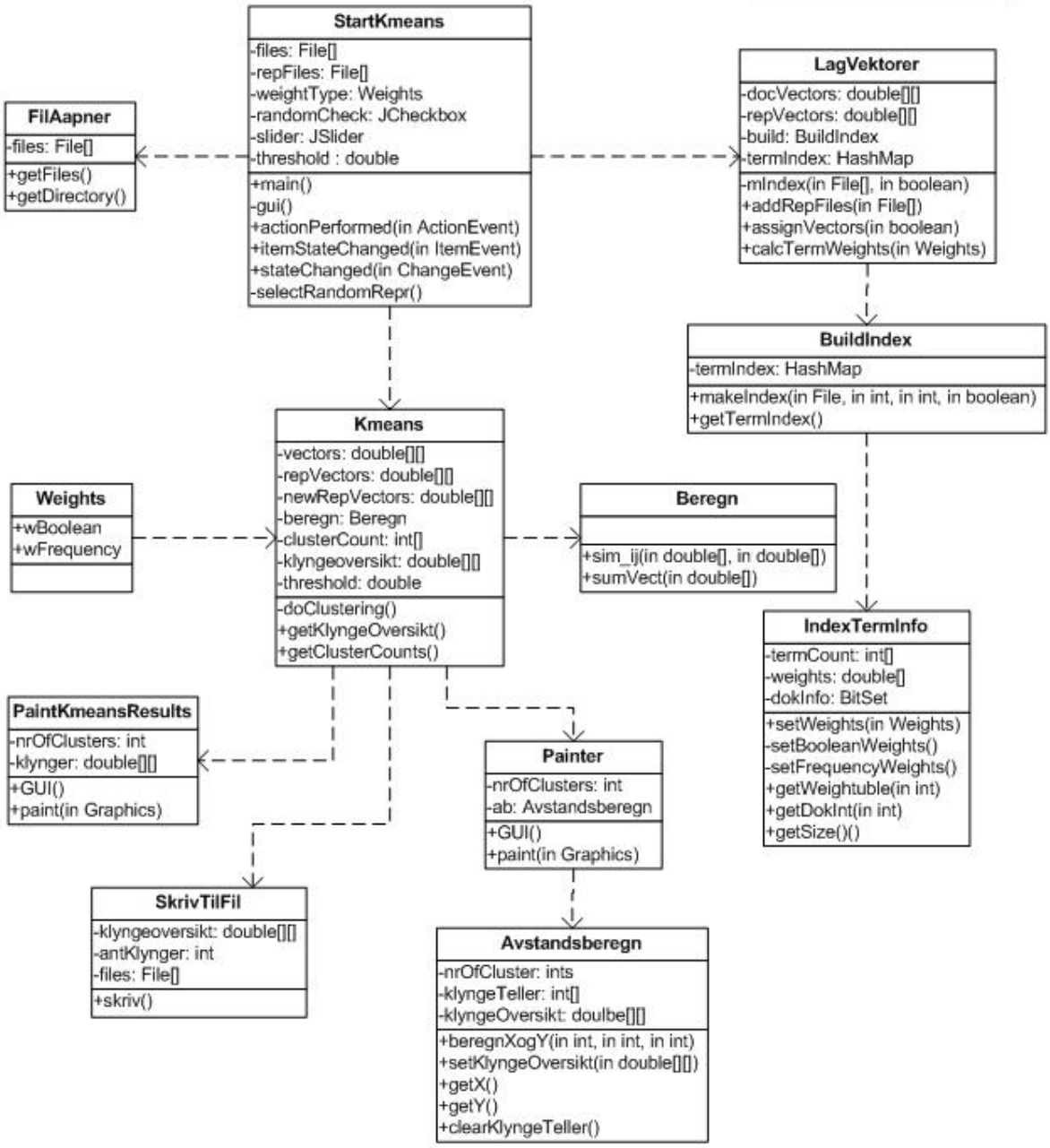
I denne grafen, som eksemplet i Figur 17 viser, betegnes dokumenter langs x-aksen og klynger langs y-aksen. Det vil si at hvert dokument er representert som en stolpe. Stolpen viser hvilken klynge dokumentet tilhører. Stopper for eksempel stolpen ved 7 på y-aksen, betyr dette at det aktuelle dokumentet tilhører klynge 7. Dokumentene er videre sortert alfabetisk slik at hver emnegruppe kommer etter hverandre sammenhengende.

8.2.5 Klasseoversikt

Prototypen for K-means er skrevet i Java og organisert i ulike klasser. En oversikt over disse klassene er vist som klassediagram i Figur 18. Programmet starter ved å kjøre StartKmeans. Her blir brukergrensesnitt tegnet opp og lyttere til knapper aktivert.

Om startknapp blir aktivert, opprettes det først objekt av LagVektorer som leser dokumentfilene og oppretter vektorer for hvert dokument samt bygger indeks ved kall til klassen BuildIndex.

BuildIndex benytter seg av klassen IndexTermInfo som er skrevet av Per Kristen Fredlund. Deretter opprettes det et objekt av type Kmeans. Det er i denne klassen selve klyngeanalysen foregår. Her blir nødvendige beregninger gjort og sluttresultat tegnet opp og skrevet til fil ved bruk av diverse hjelpeklasser som vist i figuren.



Figur 18: Klassediagram for K-means-implementering.

Organisering av klyngene

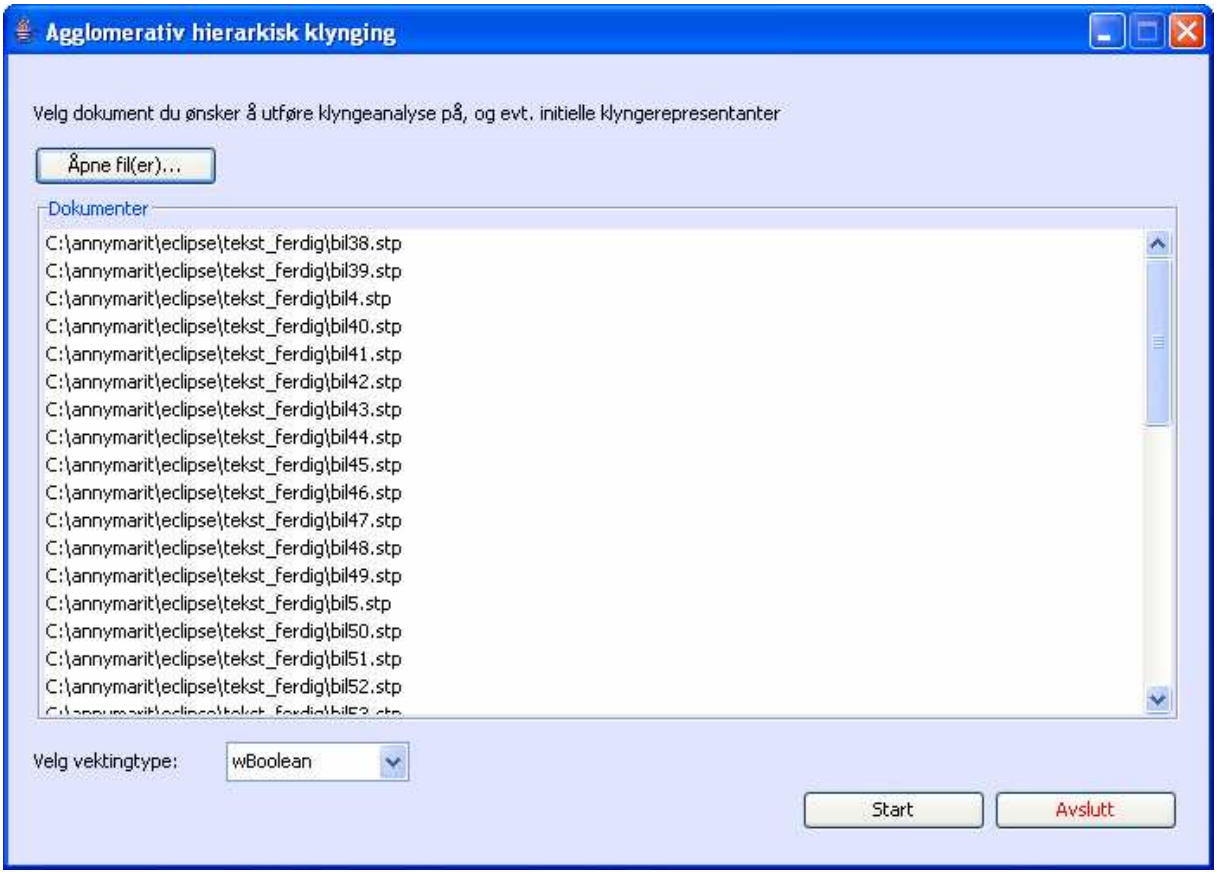
Programmet opererer i denne algoritmen med matriser for å holde styr på hvilket dokument som tilhører hvilken klynge. Det er tre slike matriser, en matrise for selve dokumentvektorene, en matrise for klyngerepresentantvektorene og en matrise for de nye klyngerepresentantvektorene (som blir beregnet på nytt for hver iterasjon). Matrisen for klyngerepresentantvektorene blir etter hver iterasjon satt til å være lik den nye klyngerepresentantvektor-matrisa.

8.3 Agglomerativ hierarkisk klynging

På samme måte som ved bruk av K-means-algoritmen, er man også i denne algoritmen avhengig av at alle tekstdokumenter som skal klynges er representert som vektorer. Dette blir her gjort på samme måte som beskrevet i seksjon 8.2.2.

8.3.1 Innputt

Brukeren må også her, i likhet med bruk av K-means velge hvilke dokumenter som skal klynges ved å åpne de tilhørende beskrivelsene av objektene. I motsetning til K-means opererer ikke hierarkisk klyngeanalyse med noen klyngerepresentanter og trenger derfor heller ikke initielle klyngerepresentanter. Et eksempel på et skjermbilde fra en kjøring med bruk av agglomerativ hierarkisk klyngeanalyse er vist i Figur 19.



Figur 19: Skjermbilde for kjøring av agglomerativ hierarkisk klyngeanalyse.

8.3.2 Likhetsmål

Algoritmen for denne klyngeanalysen (se seksjon 6.2.1) sier at man skal måle avstanden mellom punktene og dermed bygge klynger ut fra den avstanden. Man velger da de to klyngene som ligger nærmest hverandre ved å måle de to punktene som ligger lengst fra hverandre i de to klyngene (complete link, se seksjon 6.4.2), de som ligger nærmest hverandre (single link, se seksjon 6.4.1) eller ved bruk av gjennomsnittet (average link, se seksjon 6.4.3) fra de to klyngene. Avstanden mellom punktene er det vanlig å måle ved bruk av enten Manhattan distance eller Euklidisk distance (se sekjon 6.2). Uansett hva man benytter vil man sitte igjen med de to klyngene som ligger nærmest hverandre ut fra hva man definerte som mål, og lar disse klyngene forme en ny klynge. For hver iterasjon vil man altså ha en klynge mindre.

Det er i denne oppgaven jobbet med vektorer (dvs. hvert dokument er representert som en vektor), og man kan derfor ikke måle avstand på samme måte som man kan med punkter. Derfor er det også i denne prototypen valgt å bruke cosinus-similaritet som likhetsmål i stedet

for vanlige avstandsmål (se 6.2). Som nevnt tidligere er dette et mål på cosinus til størrelsen av vinkelen mellom to vektorer, og dette vil variere fra 0 til 1.

8.3.3 Single link

Min prototyp benytter single link som avstandsmålsmetode mellom to klynger. Dette vil si at den velger de to klyngene som inneholder de to punktene som ligger nærmest hverandre til å forme en ny klynge. Alle nodene i den første klyngen blir flyttet over til den andre klyngen. Den første klynga blir så slettet, fordi man ikke lenger har behov for denne.

8.3.4 Presentasjon

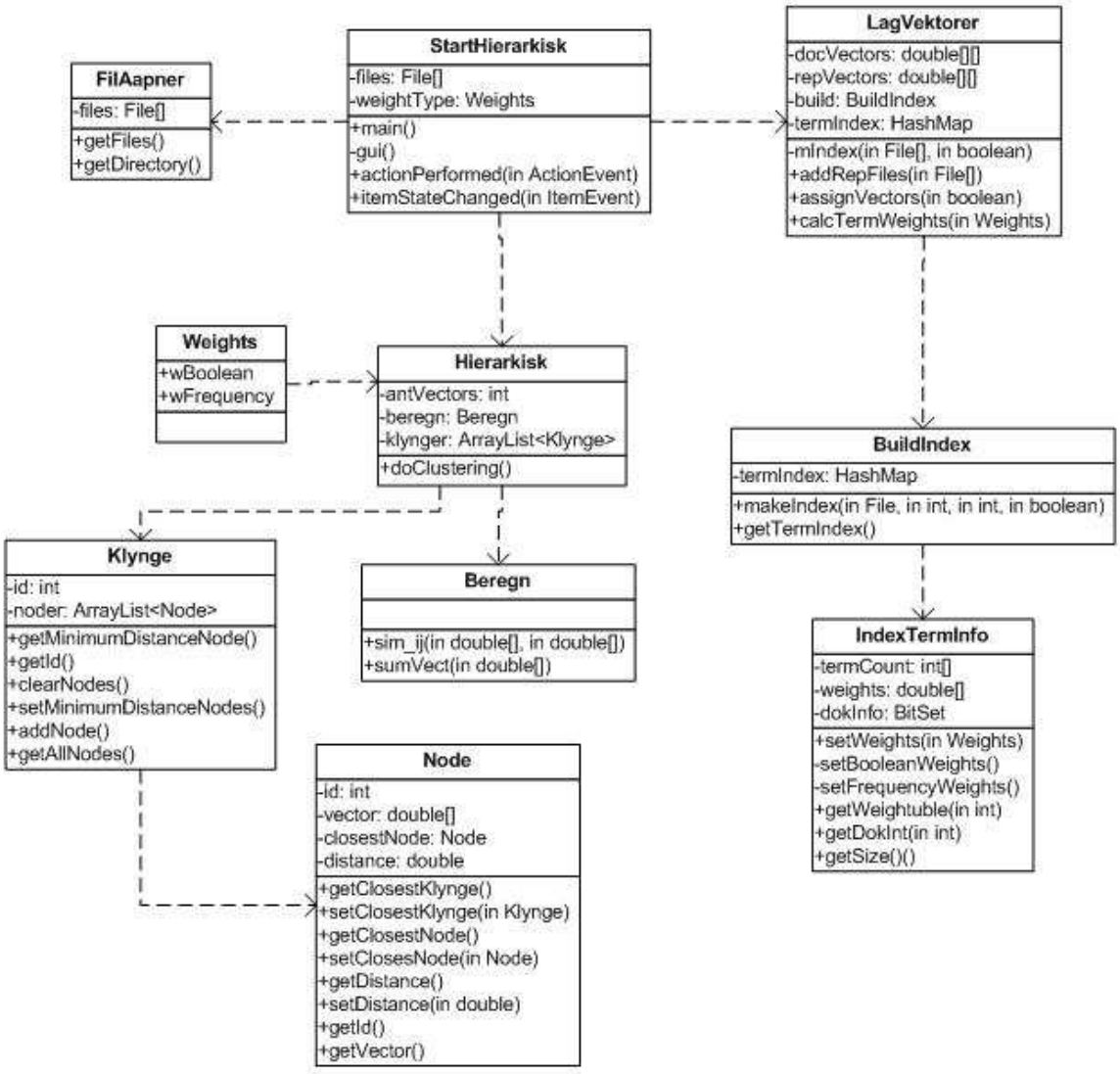
Det er ikke implementert noen grafisk framstilling av denne klyngeanalysen, resultatene skrives ut tekstlig. Man vil etter hver iterasjon få en oversikt over alle klyngene og hvilke noder som tilhører dem. For hver iterasjon vil det som sagt bli en klynge mindre, og itereringen foregår til man har igjen en klynge, dvs. alle dokumentene ligger i samme klynge.

8.3.5 Klasseoversikt

På samme måte som ved K-means er også algoritmen for agglomerativ hiearkisk klynging skrevet i java og organisert i klasser. En oversikt over klassene er vist i klassediagrammet i Figur 20. Også her startes programmet i StartHierarkisk. Her blir brukergrensesnitt tegnet opp og det lyttes til knappene.

Om starknapp blir aktivert, opprettes det også her, i likhet med algoritmen for K-means, først objekt av LagVektorer som leser dokumentfilene og oppretter vektorer for hvert dokument samt bygger indeks ved kall til klassen BuildIndex.

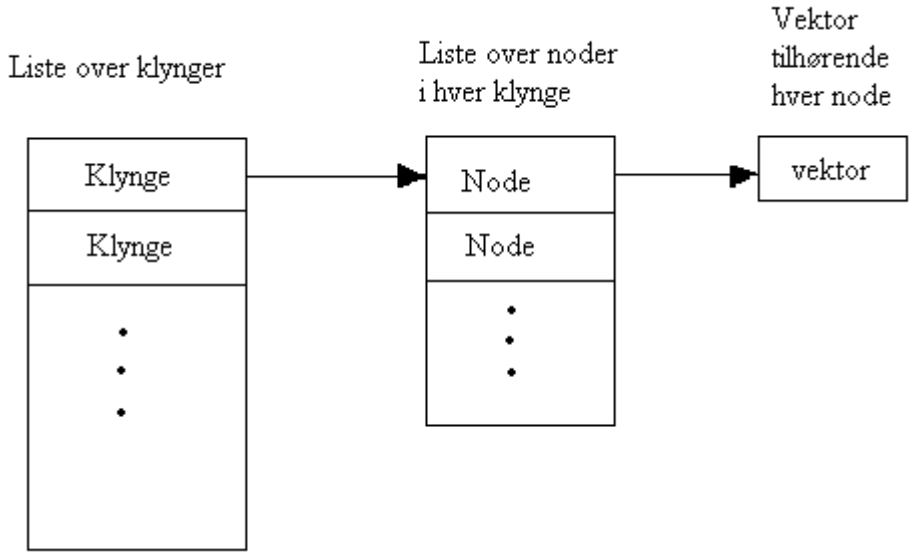
Deretter blir det opprettet objekt av type Hierarkisk, og det er i denne klassen klyngeanalysen foregår. Det er som nevnt ingen grafisk presentasjon underveis, men klyngeoversikten blir skrevet ut til fil kontinuerlig (denne metoden er inkludert i metoden doClustering() i klassen Hiearkisk).



Figur 20: Klassediagram for implementering av agglomeartiv hiearkisk klyngeanalyse.

Organisering av klyngene

Ved bruk av denne algoritmen, vil man som beskrevet i seksjon 6.2.1, starte med at alle objektene er en klynge. Prototypen for denne modellen er implementert slik at det er en liste som inneholder alle klyngene. Hver klynge inneholder igjen en liste over tilhørende noder, dvs. noder som er tilordnet klyngen. Ved initiering består hver klynge altså kun av et objekt, eller node som det er kalt i denne prototypen. Hver node inneholder vektoren for det aktuelle dokumentet samt en id så man vet hvilket dokument det er. En framstilling over denne organiseringen er vist i Figur 21.



Figur 21: Organisering av agglomerativ hierarkisk klyngeanalyse.

9 Testing av prototyper

Det er i denne oppgaven lagt hovedfokus på bruk av prototypen for K-means-algoritmen ved testing av dokumentene i samlingen. Dette begrunnes med algoritmens effektivitet, samt at det er tenkt at en bruker kjenner til kunnskapsobjektsamlingen, slik at klyngerepresentanter kan velges. Ved å la bruker velge initielle klyngerepresentanter oppnås det, som beskrevet senere i dette kapitlet meget bra klyngefordeling i forhold til disse. Prototypen for hierarkisk klyngeanalyse ble lagd for å vise at andre klyngealgoritmer også kan benyttes, men denne er ikke vektlagt som testalgoritme. Hovedgrunnen til at denne er presentert er at med denne klyngeanalysealgoritmen trenger man ikke seeds. Man vil på denne måten kunne organisere objektsamlinger hierarkisk, og det kan tenkes at kunnskapen også er organisert hierarkisk, slik at en vil få en ”ferdiglagd” presentasjon av læringsmateriellet. I og med at denne ikke ble prioritert, ble det heller ikke implementert grafisk representasjon av resultatene til denne algoritmen.

Det blir i dette kapitlet først sett på hvilke tekstdokumenter som ble brukt for testingen, før selve testingen av K-means med de aktuelle resultatene blir presentert. Til slutt blir resultatene av hierarkisk klyngeanalyse presentert.

Som testoppsett ble følgende hardware og software benyttet:

- Windows XP Pro SP2, Java JDK 1.5.0_06
- Hardware: Intel(R) Pentium(R) M prosessor 2GHz med 1024 MB RAM.
- Ved testkjøringer på store samlinger ble det tilordnet mer minne til java: -Xmx500M

9.1 Tekstdokumenter

Ettersom det å utvikle egne lærings- og kunnskapsobjekt er en meget tidkrevende oppgave, som både krever kompetanse innenfor et fagområde samt en viss pedagogisk kompetanse, er det ikke utviklet egne objekter for testing. Det er derfor i denne oppgaven utført tester for algoritmene for klyngeanalysen på tekstdokumenter for ulike emner. Disse tekstdokumentene handler om et bestemt emne, så kan derfor sammenlignes med de tekstlige beskrivelsene av lærings- og kunnskapsobjekter.

I samlingen av testdokumenter er det 340 dokumenter, gruppert i 11 ulike emner. Dokumentene er inndelt i emnene *bil, data, fiske, fotball, foto, fugl, håndball, økonomi, politikk, psykiatri* og *religion.* Dokumentene er hentet fra Internett på ulike nettaviser, nyhetsgrupper og diskusjonsfora.

Størrelsen på dokumentene varierer veldig. Det minste inneholder under 50 ulike termer, mens det største inneholder over 800 ulike termer. Men som vist i Tabell 2, ligger de fleste dokumentene i størrelsesorden 50-250 unike termer. Det er cirka i den størrelsesorden de tekstlige beskrivelsene av kunnskapsobjekt vil være.

Tabell 2: Tekstdokument gruppert etter antall unike termer.

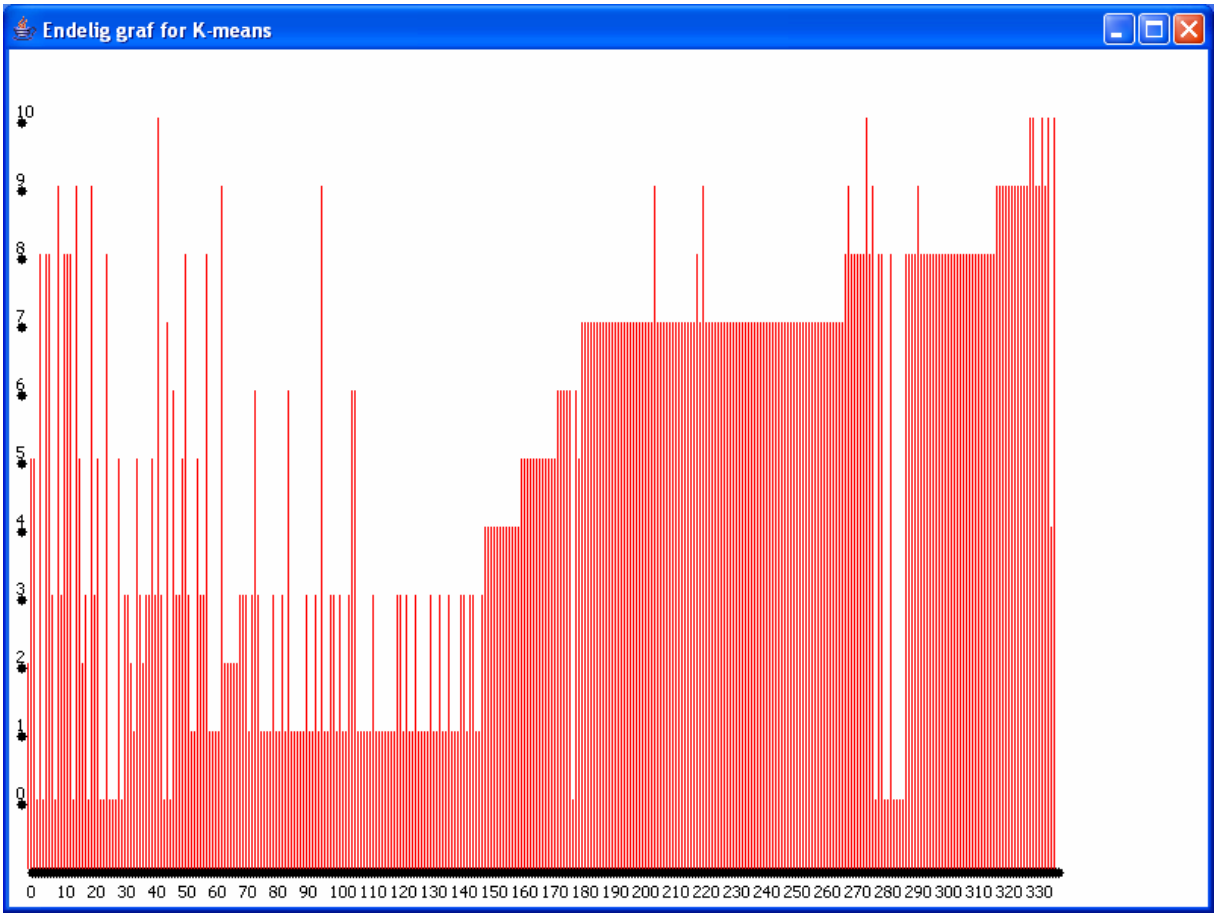
Termer	Dokumenter							
0-49	okonomi29	okonomi34	psyk5	psyk8	psyk9			
50-99	bil13	bil31	bil50	fotball18	fotball56	fotball80	okonomi71	psyk4
	bil16	bil33	data1	fotball25	fotball61	foto1	politikk21	psyk7
	bil18	bil34	data5	fotball30	fotball64	foto10	politikk23	religion1
	bil19	bil41	fotball1	fotball41	fotball65	foto12	politikk38	religion2
	bil23	bil46	fotball11	fotball5	fotball69	foto2	psyk10	religion4
	bil26	bil48	fotball16	fotball52	fotball70	foto5	psyk2	religion5
	bil27	bil49	fotball17	fotball55	fotball74	foto8	psyk3	religion8
100-149	bil12	bil39	fotball21	fotball47	fotball71	håndball1	okonomi24	politikk32
	bil14	bil52	fotball26	fotball48	fotball72	håndball5	okonomi25	politikk33
	bil15	bil6	fotball27	fotball49	fotball75	håndball6	okonomi33	politikk36
	bil17	data10	fotball28	fotball50	foto11	okonomi03	okonomi43	politikk4
	bil2	data2	fotball33	fotball51	foto3	okonomi04	okonomi62	politikk40
	bil24	data4	fotball38	fotball53	foto4	okonomi12	okonomi69	politikk6
	bil28	data9	fotball39	fotball6	foto6	okonomi15	okonomi81	psyk1
	bil29	fotball10	fotball4	fotball60	foto7	okonomi20	politikk19	psyk6
	bil32	fotball14	fotball40	fotball62	foto9	okonomi21	politikk20	religion3
	bil36	fotball15	fotball44	fotball68	fugl5	okonomi22	politikk24	religion6
	bil38	fotball19	fotball45	fotball7	fugl8	okonomi23	politikk30	religion9
150-199	bil1	bil53	fotball3	håndball2	okonomi19	okonomi65	politikk13	politikk43
	bil20	bil7	fotball31	håndball3	okonomi27	okonomi70	politikk14	politikk5
	bil21	data3	fotball35	håndball7	okonomi36	okonomi76	politikk18	psyk11
	bil30	data6	fotball36	okonomi02	okonomi37	okonomi80	politikk25	religion7
	bil35	data7	fotball37	okonomi06	okonomi41	okonomi82	politikk29	
	bil4	fiske	fotball43	okonomi08	okonomi45	okonomi84	politikk34	
	bil44	fotball12	fotball8	okonomi11	okonomi47	okonomi86	politikk37	
	bil45	fotball23	fugl11	okonomi17	okonomi49	okonomi88	politikk39	
	bil5	fotball29	fugl3	okonomi18	okonomi60	politikk1	politikk42	
200-249	bil11	fotball24	fotball81	okonomi13	okonomi52	okonomi67	politikk11	politikk48
	bil42	fotball32	fotball9	okonomi26	okonomi53	okonomi72	politikk12	politikk50
	bil8	fotball54	fugl10	okonomi39	okonomi54	okonomi78	politikk15	
	bil9	fotball59	fugl2	okonomi40	okonomi57	okonomi79	politikk16	
	fiske7	fotball63	håndball4	okonomi44	okonomi59	okonomi83	politikk27	
	fotball2	fotball66	okonomi05	okonomi46	okonomi63	okonomi85	politikk31	
	fotball22	fotball78	okonomi10	okonomi51	okonomi64	okonomi87	politikk45	
250-299	bil10	bil51	fotball20	fotball73	fugl9	okonomi58	okonomi77	politikk3
	bil25	data11	fotball42	fotball76	okonomi16	okonomi61	politikk2	politikk44
	bil3	data8	fotball57	fotball77	okonomi31	okonomi66	politikk26	politikk7
	bil37	fotball13	fotball67	fotball79	okonomi35	okonomi75	politikk28	politikk8
300-349	bil22	fotball34	fotball58	okonomi30	okonomi55	okonomi68	politikk17	politikk46
	bil43	fotball46	fugl12	okonomi38	okonomi56	okonomi73	politikk22	politikk47
350-399	bil47	fiske6	fugl4	okonomi01	okonomi32	politikk41		
	fiske3	fugl1	fugl7	okonomi14	politikk35	politikk49		
400-449	bil40	fiske5	okonomi07	okonomi09	okonomi74			
450-499	fiske4	fugl6	okonomi48	okonomi50				
500-549	politikk9							
550-599								
600-649	okonomi28							
650-699								
700-749	okonomi42							
750-799								
800-849	politikk10							

9.2 Testing av K-means

Det ble utført en rekke tester med prototypen som benytter K-means-algoritmen. Først er det testet med brukervalgte og tilfeldig valgte seeds, det vil si initielle klyngerepresentanter, eller startvektorer, på hele dokumentsamlingen for å se forskjellen og om det spiller noen rolle hva man velger. Deretter ble det testet med likt antall dokumenter fra hvert emne, også nå med både brukervalgte seeds og tilfeldig valgte seeds. Det ble så testet med variasjon av dokumentstørrelse, her med dokumenter med få termer og dokumenter med mange termer. Når dette er gjort utføres noen tester med ulik grenseverdi for å se om dette har påvirkning på resultatene. Til slutt ble det gjort test av hele samlingen med termfrekvens i stedet for boolsk vekting som ble brukt ved de andre testene. Alle disse testene er beskrevet nærmere i etterfølgende seksjoner.

9.2.1 Brukervalgte seeds

Figur 22 viser resultat av en kjøring med K-means på hele dokumentsamlingen hvor seedene, det altså de initielle klyngerepresentanter, er valgt av bruker. Her er det valgt like mange seeds som det er emner i dokumentsamlingen, 11. Det er brukt boolsk vekting (se seksjon 5.3) for vektorene til hvert dokument med grenseverdi på 0,0. Ved brukervalgte seeds vil man få samme resultat hver gang, hvis seedsene er de samme dokumentene.



Figur 22: Resultat av K-means med brukervalgte seeds.

Selv om det er noen feilklassifiseringer ut fra hva slags gruppe dokumentet er definert til å tilhøre, viser kjøring K-means med valgte seed bra resultater. Dokumentene er her organisert slik at dokument nummer 0-52 er bil, 53-63 data, 64-69 fiske, 70-150 fotball, 151-174 fugl, 175-181 håndball, 182-269 økonomi, 270-319 politikk, 320-330 psykiatri og 331-339 religion. Hvilke dokumenter som havnet i hvilken klynge også vist i Tabell 3.

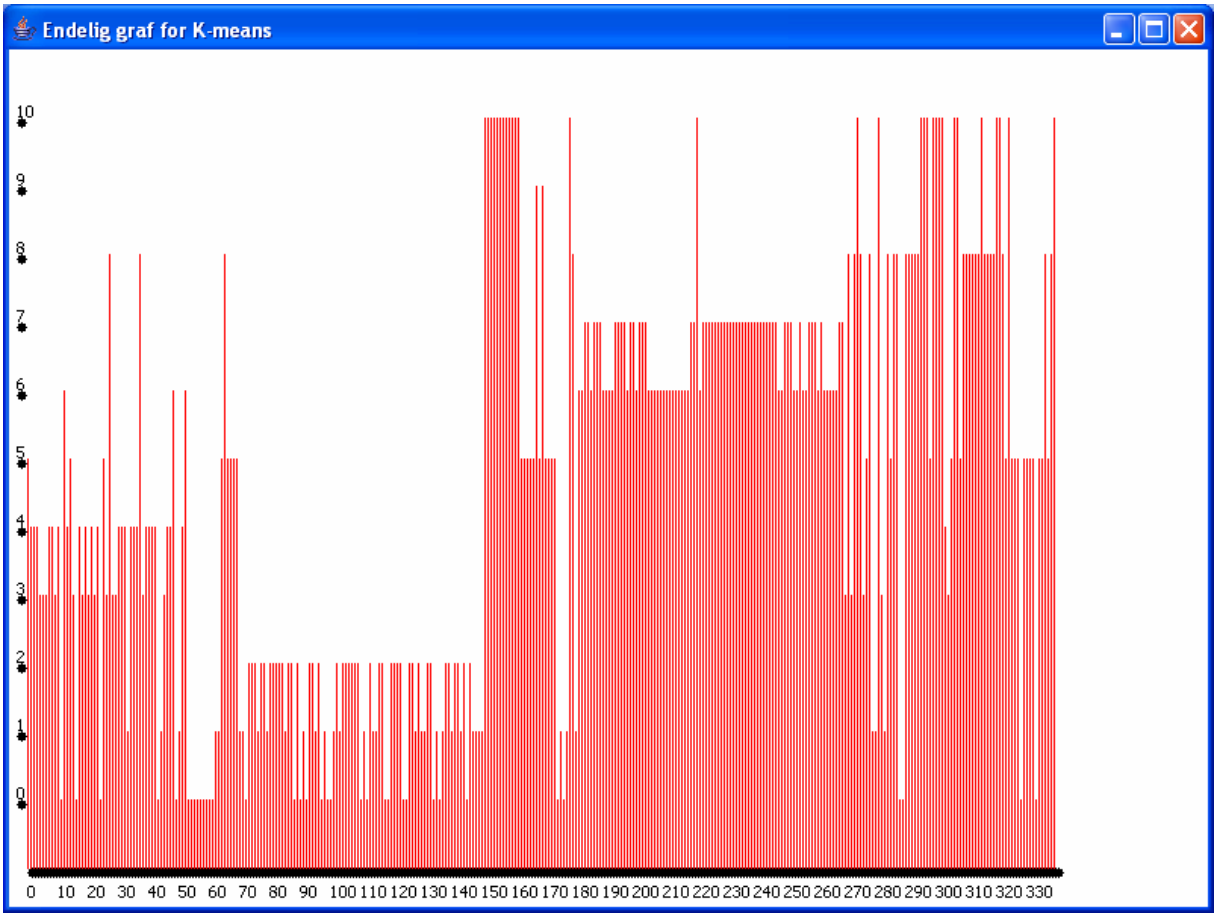
Tabell 3: Dokumentfordeling ved K-means med brukervalgte seeds.

Klynge: 0 bil12, bil14, bil18, bil23, bil28, bil31, bil32, bil34 bil35, bil36, bil38, bil50, bil52, håndball6, politikk19, politikk21 politikk22, politikk24, politikk25, politikk26, politikk27,
Klynge: 1 bil41, data10, data11, data6, data7, data8, data9, fotball12 fotball16, fotball17, fotball18, fotball19, fotball20, fotball21, fotball23, fotball25 fotball26, fotball27, fotball28, fotball29, fotball30, fotball31, fotball33, fotball35 fotball36, fotball39, fotball40, fotball41, fotball45, fotball46, fotball47, fotball48 fotball49, fotball50, fotball51, fotball52, fotball53, fotball54, fotball55, fotball56 fotball59, fotball60, fotball61, fotball63, fotball64, fotball65, fotball66, fotball68 fotball69, fotball70, fotball71, fotball73, fotball74, fotball75, fotball78, fotball80 fotball81,
Klynge: 2 bil1, bil26, bil40, bil44, fiske3, fiske4, fiske5, fiske6 fiske7,
Klynge: 3 bil17, bil2, bil27, bil3, bil39, bil4, bil43, bil45 bil46, bil48, bil5, bil6, bil7, data1, data3, data4 fotball1, fotball10, fotball11, fotball13, fotball15, fotball2, fotball22, fotball3 fotball32, fotball37, fotball38, fotball4, fotball42, fotball5, fotball57, fotball58 fotball6, fotball62, fotball67, fotball7, fotball72, fotball76, fotball77, fotball79 fotball8, fotball9,
Klynge: 4 foto1, foto10, foto11, foto12, foto2, foto3, foto4, foto5 foto6, foto7, foto8, foto9, religion8,
Klynge: 5 bil10, bil11, bil25, bil30, bil37, bil42, bil47, bil8 data2, fugl1, fugl10, fugl11, fugl12, fugl2, fugl3, fugl4 fugl5, fugl6, fugl7, fugl8, fugl9, okonomi01,
Klynge: 6 bil53, fotball14, fotball24, fotball43, fotball44, håndball1, håndball2, håndball3 håndball4, håndball5, håndball7,
Klynge: 7 bil51, okonomi02, okonomi03, okonomi04, okonomi05, okonomi06, okonomi07, okonomi08 okonomi09, okonomi10, okonomi11, okonomi12, okonomi13, okonomi14, okonomi15, okonomi16 okonomi17, okonomi18, okonomi19, okonomi20, okonomi21, okonomi22, okonomi23, okonomi24 okonomi25, okonomi27, okonomi28, okonomi29, okonomi30, okonomi31, okonomi32, okonomi33 okonomi34, okonomi35, okonomi36, okonomi37, okonomi38, okonomi39, okonomi41, okonomi43 okonomi44, okonomi45, okonomi46, okonomi47, okonomi48, okonomi49, okonomi50, okonomi51 okonomi52, okonomi53, okonomi54, okonomi55, okonomi56, okonomi57, okonomi58, okonomi59 okonomi60, okonomi61, okonomi62, okonomi63, okonomi64, okonomi65, okonomi66, okonomi67 okonomi68, okonomi69, okonomi70, okonomi71, okonomi72, okonomi73, okonomi74, okonomi75 okonomi76, okonomi77, okonomi78, okonomi79, okonomi80, okonomi81, okonomi82, okonomi83 okonomi84, okonomi85, okonomi86, okonomi87, okonomi88,
Klynge: 8 bil13, bil15, bil16, bil20, bil21, bil22, bil33, bil9 data5, okonomi40, politikk1, politikk11, politikk12, politikk13, politikk14, politikk15 politikk17, politikk2, politikk20, politikk23, politikk28, politikk29, politikk3, politikk30 politikk32, politikk33, politikk34, politikk35, politikk36, politikk37, politikk38, politikk39 politikk4, politikk40, politikk41, politikk42, politikk43, politikk44, politikk45, politikk46 politikk47, politikk48, politikk49, politikk5, politikk50, politikk6, politikk7, politikk8 politikk9,
Klynge: 9 bil19, bil24, bil29, fiske, fotball34, okonomi26, okonomi42, politikk10 politikk18, politikk31, psyk1, psyk10, psyk11, psyk2, psyk3, psyk4 psyk5, psyk6, psyk7, psyk8, psyk9, religion3, religion4, religion6
Klynge: 10 bil49, politikk16, religion1, religion2, religion5, religion7, religion9,

9.2.2 Tilfeldig valgte seeds

Kjører man K-means slik algoritmen opprinnelig er, dvs. med tilfeldig valgte initielle klyngerepresentanter, seeds, får man også mer tilfeldig resultat enn ved brukervalgte seeds. Et resultat fra en slik kjøring på hele dokumentsamlingen er vist i Figur 23. Her er det på samme måte som ved testkjøringen av brukervalgte seeds, brukt like mange seeds som emner i dokumentsamlingen, nemlig 11. Også her, som i kjøring med brukervalgte seeds, er det brukt boolsk vekting med grenseverdi satt til 0,0. Dokumentene er her organisert slik at dokument nummer 0-52 er bil, 53-63 data, 64-69 fiske, 70-150 fotball, 151-174 fugl, 175-181 håndball, 182-269 økonomi, 270-319 politikk, 320-330 psykiatri og 331-339 religion.

Resultatet av denne kjøringen kan ved første øyekast se veldig rotete ut, men ved nærmere ettersyn ser man at også her er dokumentene oppdelt i klynger. Man vil ikke få den samme ”trappeformen” som med brukervalgte seeds, fordi med brukervalgte seeds, hadde hvert emne et seed i samme alfabetisk rekkefølge som dokumentene. Ved tilfeldige seeds kan det være flere dokument fra samme emne som blir valgt som seed, og derfor blir klyngerekkefølgen også mer tilfeldig.



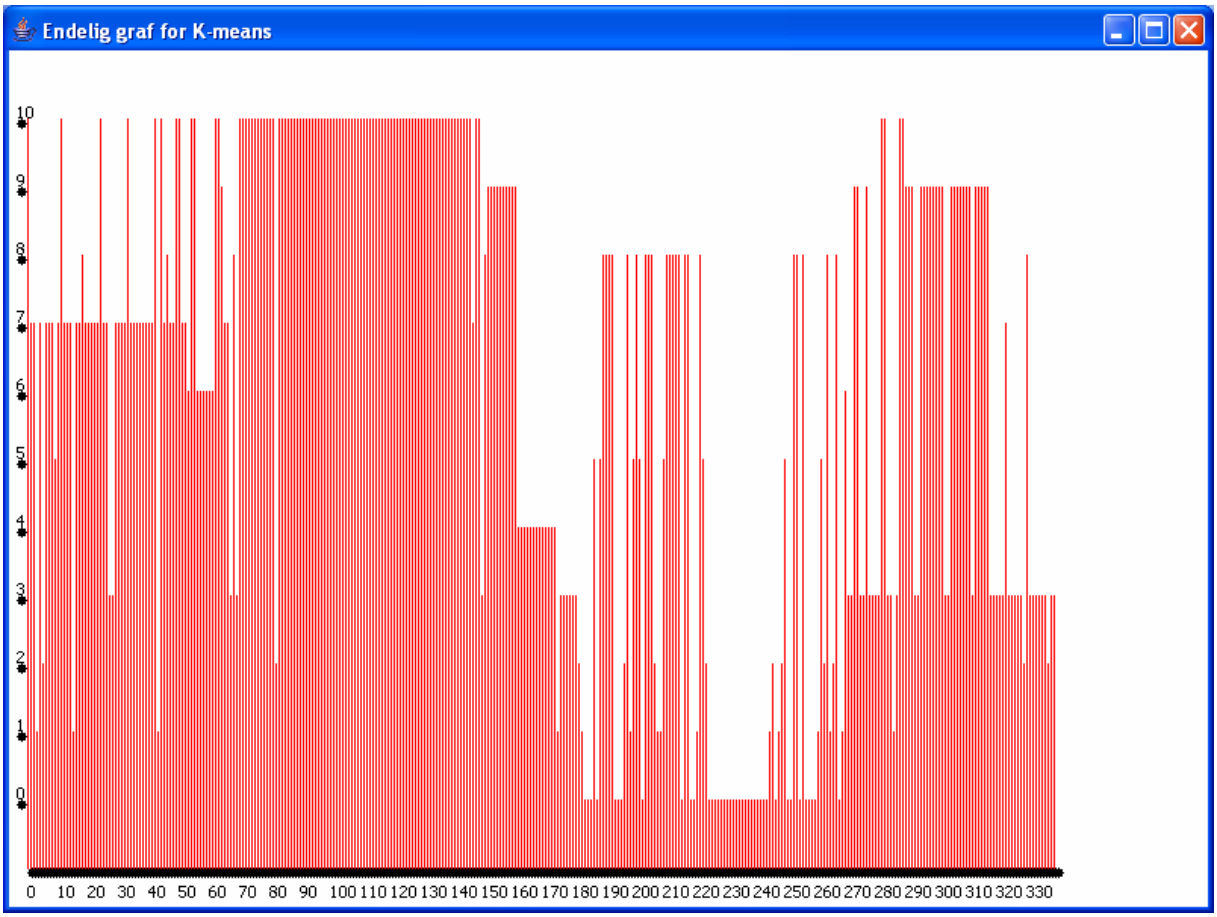
Figur 23: Resultat av K-means med tilfeldig valgte seeds.

Den tilhørende tabellen over dokumentfordelingen (Tabell 1), viser her at dokumentene også her delvis er klynget slik at samme type dokument havner i samme klynge. Men i motsetning til K-means med brukervalgte seeds, splitter den her opp mye mer. Man ser for eksempel at økonomidokumentene i to klynger. K-means med brukervalgte seeds tilordnet så å si alle økonomiobjektene i samme gruppe. Et annet eksempel er fotballdokumentene. Med tilfeldige seeds havner disse hovedsakelig i to klynger, mens med brukervalgte seeds blir disse gruppert i samme klynge.

Tabell 4: Dokumentfordeling ved K-means med tilfeldige seeds.

Klynge: 0 bil2, bil24, bil31, bil49, bil6, data1, data10, data11 data2, data3, data4, data5, data6, data7, fotball11, fotball26 fotball28, fotball3, fotball34, fotball36, fotball37, fotball46, fotball48, fotball53 fotball54, fotball59, fotball6, fotball68, fotball7, fotball78, håndball1, håndball3 politikk26, politikk27, psyk7, religion3,
Klynge: 1 bil4, bil5, bil7, data8, data9, fotball1, fotball10, fotball15 fotball18, fotball23, fotball29, fotball32, fotball35, fotball38, fotball4, fotball47 fotball5, fotball50, fotball62, fotball64, fotball65, fotball69, fotball70, fotball73 fotball76, fotball8, fotball80, fotball81, fotball9, håndball2, håndball4, håndball7 politikk18, politikk19, politikk21,
Klynge: 2 fotball12, fotball13, fotball14, fotball16, fotball17, fotball19, fotball2, fotball20 fotball21, fotball22, fotball24, fotball25, fotball27, fotball30, fotball31, fotball33 fotball39, fotball40, fotball41, fotball42, fotball43, fotball44, fotball45, fotball49 fotball51, fotball52, fotball55, fotball56, fotball57, fotball58, fotball60, fotball61 fotball63, fotball66, fotball67, fotball71, fotball72, fotball74, fotball75, fotball77 fotball79,
Klynge: 3 bil13, bil14, bil15, bil18, bil23, bil26, bil28, bil3 bil33, bil35, bil36, bil44, bil50, politikk1, politikk11, politikk15 politikk20, politikk40,
Klynge: 4 bil10, bil11, bil12, bil16, bil17, bil19, bil21, bil25 bil27, bil29, bil30, bil37, bil38, bil39, bil40, bil41 bil42, bil45, bil46, bil47, bil48, bil51, bil52, bil8 politikk4,
Klynge: 5 bil1, bil22, bil32, fiske, fiske4, fiske5, fiske6, fiske7 fugl1, fugl10, fugl11, fugl12, fugl2, fugl4, fugl6, fugl7 fugl8, fugl9, politikk16, politikk23, politikk35, politikk41, politikk44, psyk2 psyk4, psyk5, psyk6, psyk8, psyk9, religion1, religion2, religion4 religion5, religion7,
Klynge: 6 bil20, bil53, bil9, okonomi01, okonomi02, okonomi05, okonomi09, okonomi10 okonomi11, okonomi12, okonomi17, okonomi20, okonomi24, okonomi25, okonomi26, okonomi27 okonomi28, okonomi29, okonomi30, okonomi31, okonomi32, okonomi33, okonomi34, okonomi35 okonomi36, okonomi37, okonomi41, okonomi67, okonomi68, okonomi72, okonomi73, okonomi75 okonomi76, okonomi80, okonomi82, okonomi83, okonomi84, okonomi85, okonomi86,
Klynge: 7 okonomi03, okonomi04, okonomi06, okonomi07, okonomi08, okonomi13, okonomi14, okonomi15 okonomi16, okonomi18, okonomi19, okonomi21, okonomi22, okonomi23, okonomi38, okonomi39 okonomi42, okonomi43, okonomi44, okonomi45, okonomi46, okonomi47, okonomi48, okonomi49 okonomi50, okonomi51, okonomi52, okonomi53, okonomi54, okonomi55, okonomi56, okonomi57 okonomi58, okonomi59, okonomi60, okonomi61, okonomi62, okonomi63, okonomi64, okonomi65 okonomi66, okonomi69, okonomi70, okonomi71, okonomi74, okonomi77, okonomi78, okonomi79 okonomi81, okonomi87, okonomi88,
Klynge: 8 bil34, bil43, fiske3, håndball6, politikk10, politikk12, politikk14, politikk17 politikk22, politikk24, politikk25, politikk28, politikk29, politikk3, politikk30, politikk31 politikk45, politikk46, politikk47, politikk48, politikk49, politikk5, politikk6, politikk7 politikk8, politikk9, psyk11, religion6, religion8,
Klynge: 9 fugl3, fugl5,
Klynge: 10 foto1, foto10, foto11, foto12, foto2, foto3, foto4, foto5 foto6, foto7, foto8, foto9, håndball5, okonomi40, politikk13, politikk2 politikk32, politikk33, politikk34, politikk36, politikk37, politikk38, politikk39, politikk42 politikk43, politikk50, psyk1, psyk10, psyk3, religion9,

I motsetning til K-means med brukervalgte seeds som gir samme resultat ved hver kjøring, vil kjøring med tilfeldige seeds gi ulikt resultat ved flere kjøringer. Dette er et av de problemene som blir sett på som en svakhet ved vanlig K-means. For å illustrere dette er det her gjort enda en testkjøring på hele dokumentsamlingen, med samme parametere som den forrige. Figur 24 viser at resultatet er ganske forskjellig fra resultatet vist i Figur 23: Resultat av K-means med tilfeldig valgte seeds.Figur 23.



Figur 24: Resultat av K-means med tilfeldig valgte seeds.

Den tilhørende tabellen (Tabell 5) viser også at dokumentene er fordelt noe annerledes enn hva Tabell 4: Dokumentfordeling ved K-means med tilfeldige seeds.Tabell 4 viser. I likhet med forrige kjøring er økonomidokumentene nå også splittet. Ved forrige kjøring havnet disse dokumentene i 2 ulike klynger, mens de nå har havnet i 5 ulike klynger. Ser man derimot på fotballdokumentene var disse splittet i to ulike klynger ved forrige kjøring, mens de nå er samlet i samme klynge. Ellers er resten av dokumentene forholdsvis gruppert, men på langt nær like bra som testkjøringen av brukervalgte seeds viste.

Tabell 5: Dokumentfordeling ved K-means med tilfeldig valgte seeds.

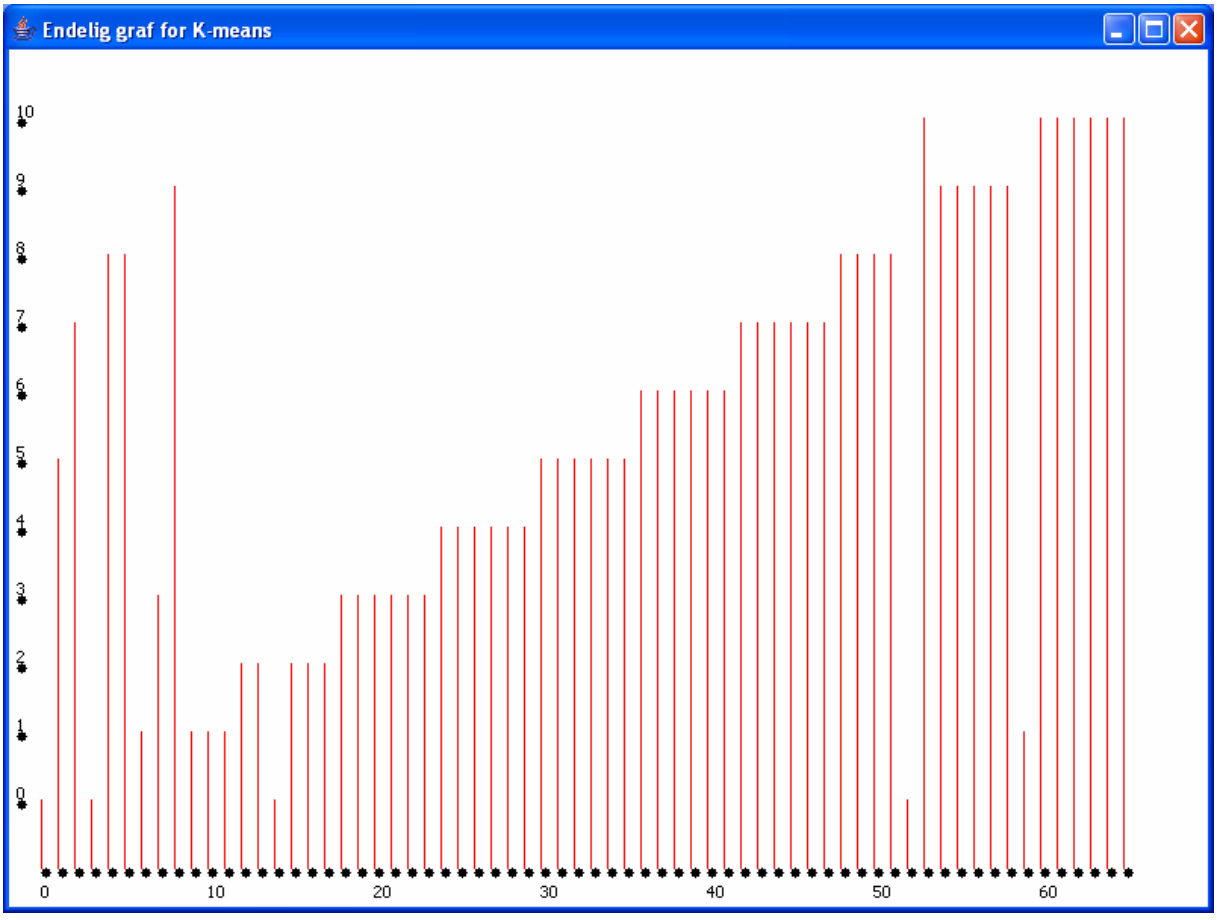
Klynge: 0
okonomi03, okonomi04, okonomi05, okonomi07, okonomi13, okonomi14, okonomi15, okonomi22 okonomi35, okonomi38, okonomi39, okonomi44, okonomi45, okonomi46, okonomi47, okonomi48 okonomi49, okonomi50, okonomi51, okonomi52, okonomi53, okonomi54, okonomi55, okonomi56 okonomi57, okonomi58, okonomi59, okonomi60, okonomi61, okonomi62, okonomi63, okonomi66 okonomi70, okonomi71, okonomi74, okonomi76, okonomi77, okonomi78, okonomi79, okonomi87
Klynge: 1
bil12, bil23, bil49, håndball1, okonomi02, okonomi18, okonomi27, okonomi28 okonomi40, okonomi64, okonomi67, okonomi80, okonomi84, okonomi88, politikk24,
Klynge: 2
bil14, fotball20, okonomi01, okonomi16, okonomi26, okonomi43, okonomi65, okonomi68 okonomi82, okonomi85, psyk8, religion7,
Klynge: 3
bil34, bil35, fiske5, fiske7, fotball9, håndball2, håndball3, håndball4 håndball5, håndball6, håndball7, politikk10, politikk11, politikk14, politikk15, politikk17 politikk18, politikk19, politikk2, politikk22, politikk23, politikk25, politikk30, politikk31 politikk4, politikk40, politikk48, politikk8, politikk9, psyk1, psyk10, psyk11 psyk3, psyk4, psyk5, psyk6, psyk7, religion1, religion2, religion3 religion4, religion5, religion6, religion8, religion9,
Klynge: 4
foto9, fugl1, fugl10, fugl11, fugl12, fugl2, fugl3, fugl4 fugl5, fugl6, fugl7, fugl8, fugl9,
Klynge: 5
bil18, okonomi06, okonomi08, okonomi19, okonomi21, okonomi29, okonomi42, okonomi69 okonomi81,
Klynge: 6
data1, data2, data3, data4, data5, data6, data7, politikk1
Klynge: 7
bil10, bil11, bil13, bil15, bil16, bil17, bil19, bil20 bil21, bil22, bil24, bil25, bil27, bil28, bil29, bil3 bil30, bil32, bil33, bil36, bil37, bil38, bil39, bil40 bil41, bil42, bil43, bil44, bil45, bil46, bil47, bil50 bil52, bil53, bil8, bil9, fiske3, fiske4, fotball8, psyk2
Klynge: 8
bil26, bil51, fiske6, foto1, okonomi09, okonomi10, okonomi11, okonomi12 okonomi17, okonomi20, okonomi23, okonomi24, okonomi25, okonomi30, okonomi31, okonomi32 okonomi33, okonomi34, okonomi36, okonomi37, okonomi41, okonomi72, okonomi73, okonomi75 okonomi83, okonomi86, psyk9,
Klynge: 9
fiske, foto10, foto11, foto12, foto2, foto3, foto4, foto5 foto6, foto7, foto8, politikk12, politikk13, politikk16, politikk28, politikk29 politikk3, politikk32, politikk33, politikk34, politikk35, politikk36, politikk37, politikk38 politikk39, politikk41, politikk42, politikk43, politikk44, politikk45, politikk46, politikk47 politikk49, politikk5, politikk50, politikk6, politikk7,
Klynge: 10
bil1, bil2, bil31, bil4, bil48, bil5, bil6, bil7 data10, data11, data8, data9, fotball1, fotball10, fotball11, fotball12 fotball13, fotball14, fotball15, fotball16, fotball17, fotball18, fotball19, fotball2 fotball21, fotball22, fotball23, fotball24, fotball25, fotball26, fotball27, fotball28 fotball29, fotball3, fotball30, fotball31, fotball32, fotball33, fotball34, fotball35 fotball36, fotball37, fotball38, fotball39, fotball4, fotball40, fotball41, fotball42 fotball43, fotball44, fotball45, fotball46, fotball47, fotball48, fotball49, fotball5 fotball50, fotball51, fotball52, fotball53, fotball54, fotball55, fotball56, fotball57 fotball58, fotball59, fotball6, fotball60, fotball61, fotball62, fotball63, fotball64 fotball65, fotball66, fotball67, fotball68, fotball69, fotball7, fotball70, fotball71 fotball72, fotball73, fotball74, fotball75, fotball76, fotball77, fotball78, fotball79 fotball80, fotball81, politikk20, politikk21, politikk26, politikk27,

9.2.3 Likt antall dokumenter fra hvert emne

Det ble utført testing med likt antall dokumenter fra hvert emne for å se om det har noe å si om det er mange dokumenter fra et emne og få fra et annet emne. I og med at det er et emne i samlingen som bare har 6 dokumenter, ble disse testene utført med 6 dokumenter fra hvert emne. Dokumentene er også her sortert alfabetisk, og det er brukt boolsk vekting med grenseverdi på 0,0.

Brukervalgt seeds

Resultat for denne testingen med brukervalgte seeds er vist i Figur 25. Her ser en at klyngefordelingen er meget jevn, og de aller fleste dokumentene havner i riktig klynge, etter hva som er forhåndsdefinert som emner for dokumentene.



Figur 25: Resultat av K-means med 6 dokumenter fra hvert emne og brukervalgte seeds.

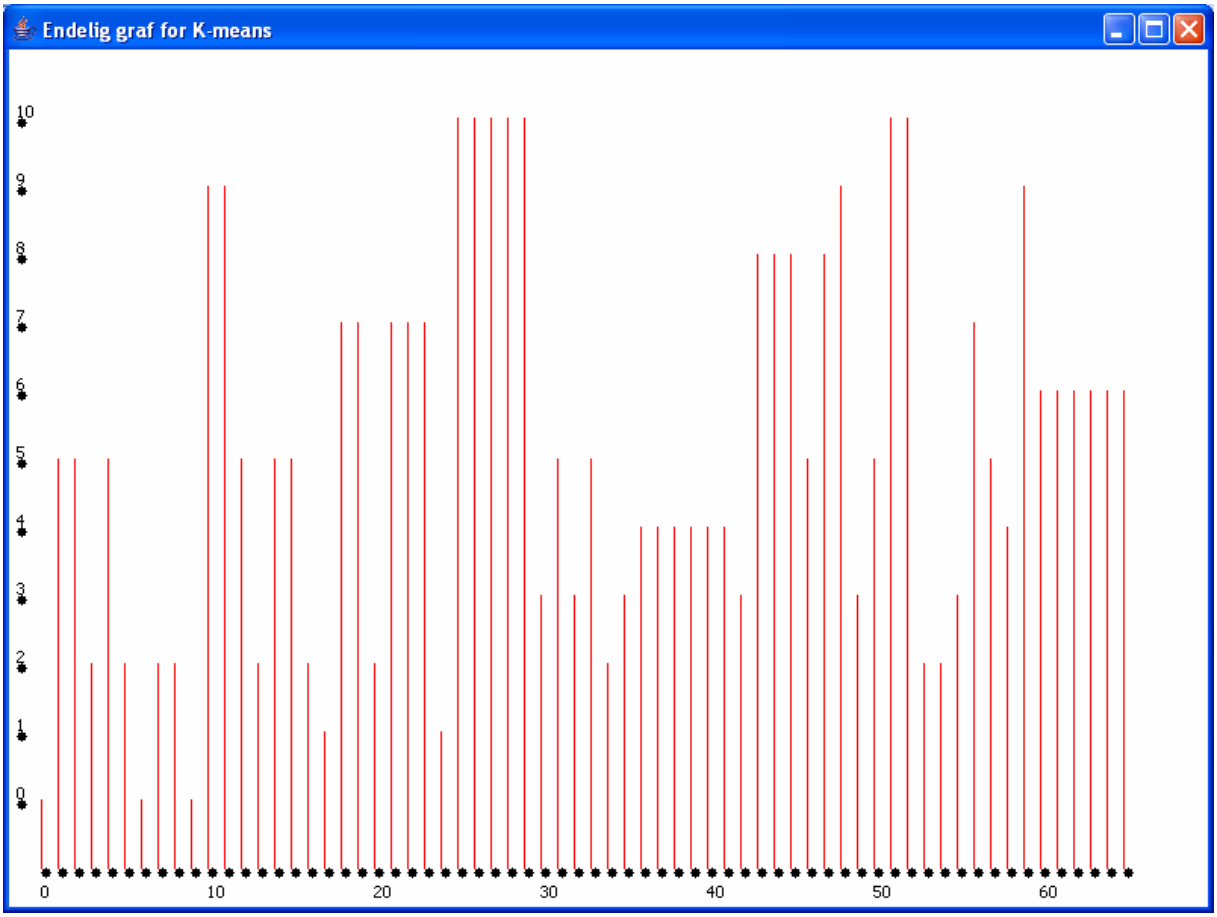
Tabell 6: Dokumentfordeling ved K-means med 6 dokumenter fra hvert emne og brukervalgte seeds.

Klynge: 0 bil1, bil12, fiske4, politikk13,
Klynge: 1 data1, data2, data3, data4, psyk4,
Klynge: 2 fiske, fiske3, fiske5, fiske6, fiske7,
Klynge: 3 data10, fotball1, fotball10, fotball11, fotball12, fotball13, fotball14,
Klynge: 4 foto1, foto10, foto11, foto12, foto2, foto3,
Klynge: 5 bil10, fugl1, fugl10, fugl11, fugl12, fugl2, fugl3,
Klynge: 6 håndball1, håndball2, håndball3, håndball4, håndball5, håndball6,
Klynge: 7 bil11, okonomi01, okonomi02, okonomi03, okonomi04, okonomi05, okonomi06,
Klynge: 8 bil13, bil14, politikk1, politikk10, politikk11, politikk12,
Klynge: 9 data11, psyk1, psyk10, psyk11, psyk2, psyk3,
Klynge: 10 politikk14, religion1, religion2, religion3, religion4, religion5, religion6,

Av Tabell 6 som viser dokumentfordelingen av klyngeanalysen, ser en at bildokumentene blir fordelt i flere ulike klynger, mens de fleste andre dokumentene grupperes etter emne.

Tilfeldig valgte seeds

Figur 26 viser resultat fra testkjøringen med 6 dokumenter fra hvert emne i dokumentsamlingen med tilfeldig valgte seeds. Her er det på samme måte som i testkjøringen med brukervalgte seeds, 11 klynger.



Figur 26: Resultat av K-means med 6 dokumenter fra hvert emne og tilfeldige seeds.

Dokumentfordelingen av denne testkjøringen er vist i Tabell 7. Igjen ser en at bildokumentene havner i flere forskjellige grupper. Her er det også en del andre dokumenter som blir splittet, deriblant fiske-, politikk- og psykiatridokumentene.

Tabell 7: Dokumentfordeling ved K-means med 6 dokumenter fra hvert emne og tilfeldige seeds.

Klynge: 0 bil1, data1, data2,
Klynge: 1 fiske7, foto1,
Klynge: 2 bil12, bil14, data10, data11, fiske3, fiske6, fotball11, fugl2 politikk14, psyk1,
Klynge: 3 fugl1, fugl11, fugl3, okonomi01, politikk10, psyk10,
Klynge: 4 håndball1, håndball2, håndball3, håndball4, håndball5, håndball6, psyk3,
Klynge: 5 bil10, bil11, bil13, fiske, fiske4, fiske5, fugl10, fugl12 okonomi05, politikk11, psyk2,
Klynge: 6 religion1, religion2, religion3, religion4, religion5, religion6,
Klynge: 7 fotball1, fotball10, fotball12, fotball13, fotball14, psyk11,
Klynge: 8 okonomi02, okonomi03, okonomi04, okonomi06,
Klynge: 9 data3, data4, politikk1, psyk4,
Klynge: 10 foto10, foto11, foto12, foto2, foto3, politikk12, politikk13,

Som nevnt algoritmen for K-means med brukervalgte seeds mye bedre enn med tilfeldige seeds, noe Tabell 6 og Tabell 7 også viser. Dokumentene i de ulike emnene er mer spredt ved tilfeldige seeds enn ved brukervalgte seeds. Det som derimot er felles for begge disse to testkjøringene er at bildokumentene blir forholdsvis spredt. Ser man nærmere på disse bildokumentene ser man at selv om de er klassifisert under samme emne, bil, har de ganske forskjellig innhold. ”bil1” handler om turtabber og ruteplanlegging, ”bil10” handler om hvilken bil en bør velge, ”bil11” handler om småbiler, ”bil12” handler om kutt i kjøregodtgjørelsen, ”bil13” handler om hissige bilister og ”bil14” handler om stjernetegn i forbindelse med bilulykker. Det er da ikke så rart at disse havner i ulike klynger. Selv om en person kan forbinde disse emnene med bil, inneholder dokumentene ikke veldig mange felles termer. Ettersom termene er det eneste algoritmen benytter for å måle dokumenter opp mot hverandre, vil altså disse dokumentene ofte ha større likhet med andre dokumenter fra andre klynger enn innad i samme emne, nettopp på grunn av at de er så forskjellige med tanke på innhold.

Ellers ser en at med likt antall dokumenter fra hvert emne gir gode resultater, men dette har sannsynligvis lite å si for algoritmen. Som testresultatene i seksjon 9.2.1 viser, ga kjøring med brukervalgte seeds relativt gode resultater. Hovedparten av dokumentene ble gruppert riktig ut

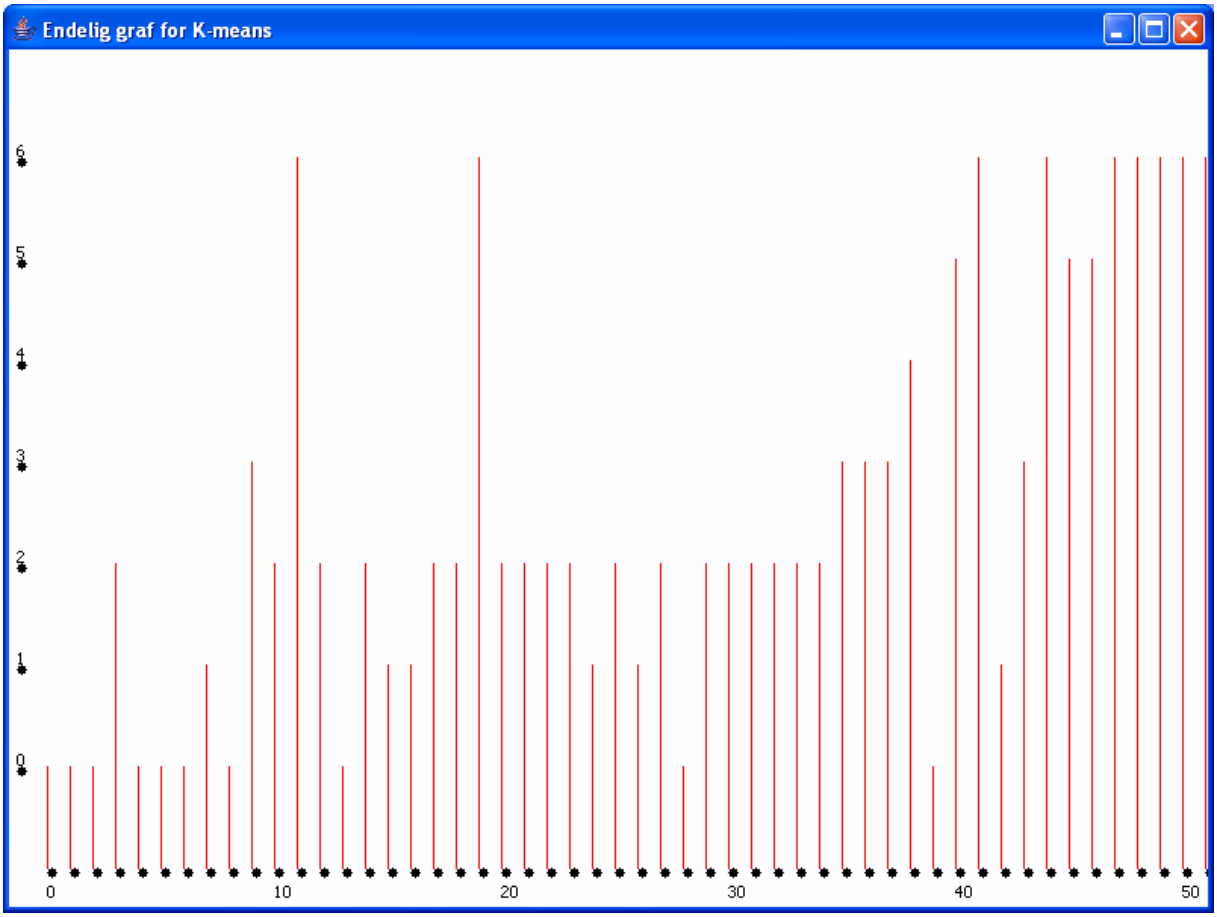
fra hva som var forhåndsdefinert emne, og testen ga bare 67 feilklassifiseringer. Også denne testen viste at bildokumentene var litt problematiske, med stor andel av feilklassifiseringer. Hovedgrunnen til dette er sannsynligvis som nevnt at selv om de er gruppert i samme emne, består de av svært ulikt innhold.

9.2.4 Variasjon av dokumentstørrelse

For å se om dokumentstørrelsen har noen innvirkning på resultatene for klyngeanalysen, ble det utført tester med få termer og mange termer. Disse er beskrevet nærmere i følgende seksjoner. Begge testene her ble gjort med K-means med brukervalgte seeds, grenseverdi på 0,0 og boolsk vekting.

Dokumenter med få termer

Denne testen ble utført på de dokumentene med færrest termer. Dokumentene som ble valgt ut er de fra de to første blokkene i Tabell 2. Som tabellen viser er størrelsen på disse dokumentene fra 0-99 termer. Ettersom det ikke inngår dokumenter fra alle emnene i denne samlingen, men bare dokumenter fra 7 ulike emner, ble det også brukt 7 valgte seeds. Emnene til denne testingen var bil, data, fotball, foto, økonomi, psykiatri og religion, og seedsene gitt inn i samme alfabetisk rekkefølge. Resultatene er vist i Figur 27, og tilhørende Tabell 8.



Figur 27: K-means på dokumenter med få termer, brukervalgt seeds.

Tabell 8: Dokumentfordeling ved K-means på dokumenter med få termer, brukervalgte seeds.

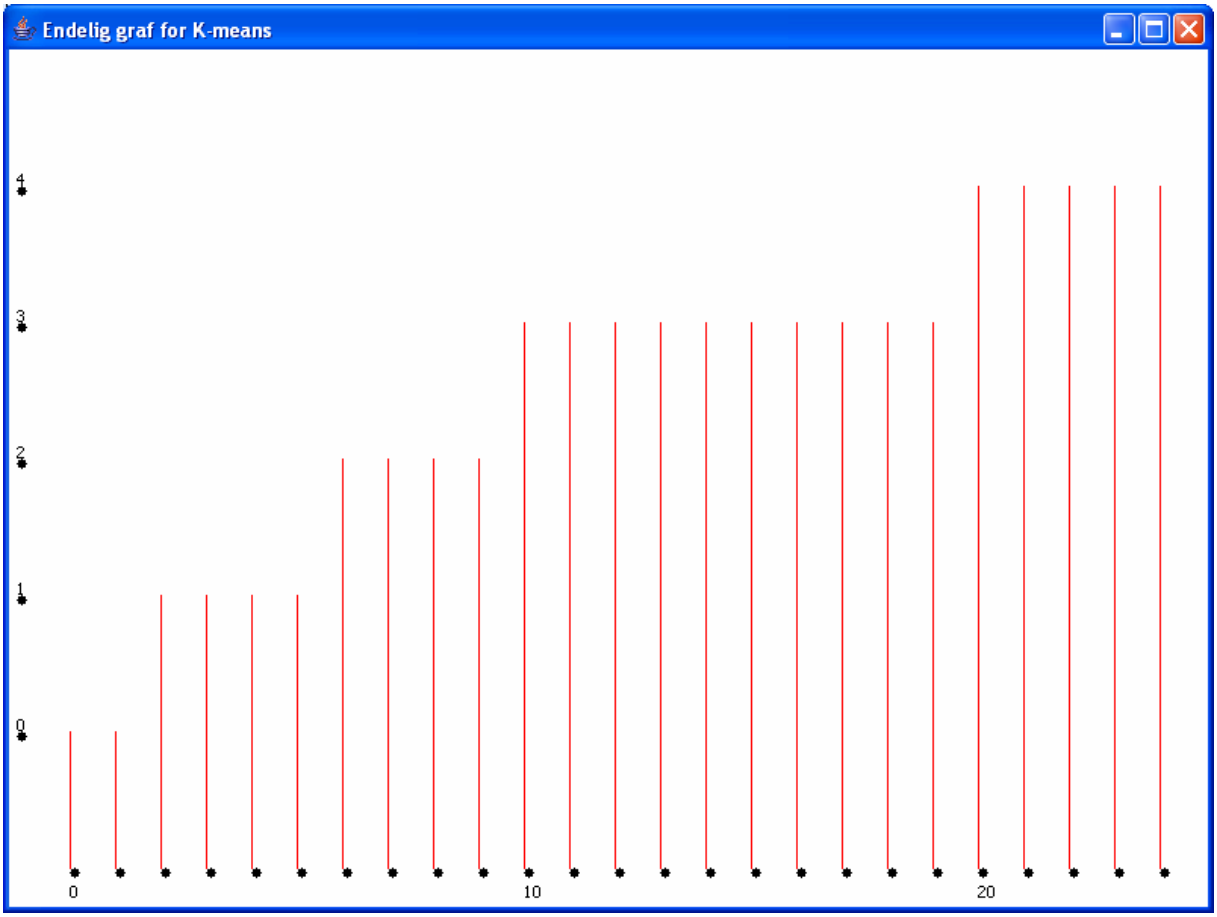
Klynge: 0 bil13, bil16, bil18, bil23, bil26, bil27, bil33, bil49 fotball61, okonomi34,
Klynge: 1 bil31, data1, data5, fotball5, fotball55, psyk4,
Klynge: 2 bil19, bil41, bil48, bil50, fotball11, fotball16, fotball18, fotball25 fotball30, fotball41, fotball52, fotball56, fotball64, fotball65, fotball69, fotball70 fotball74, fotball80,
Klynge: 3 bil34, foto1, foto10, foto12, psyk5,
Klynge: 4 okonomi29,
Klynge: 5 psyk2, psyk8, psyk9,
Klynge: 6 bil46, fotball17, psyk3, psyk7, religion1, religion2, religion4, religion5 religion8,

Tabellen viser her at de dokumentene som algoritmen hovedsakelig sliter med å tilordne til samme klynge er bildokumentene. De tidligere testene viser også at disse dokumentene var

noe problematiske, derfor er det ikke nødvendigvis dokumentlengden som spiller noen rolle, men heller innholdet i disse dokumentene. Ettersom de aller fleste andre dokumentene havnet i klynger i henhold til seedsene som ble brukt, kan man slutte at i denne samlingen får man relativt gode resultater fra en klyngeanalyse, selv på dokumenter med få termer.

Dokumenter med mange termer

Til denne testen ble det brukt de dokumentene med flest termer. Da det er få dokumenter med mange termer varierer dokumentstørrelsen noe mer enn i forrige test med små dokumenter. Nå er det brukt dokumenter fra de ti siste blokkene. Dokumentstørrelsen på disse dokumentene varierer mellom 350-849. Det er dokumenter fra 5 ulike emner, bil, fiske, fugl, økonomi og politikk, igjen er seedsene valgt ett for hvert emne i alfabetisk rekkefølge. Resultatet er vist i Figur 28 og Tabell 9. Figuren viser en perfekt trappeformfordeling. I og med at både dokumentene og klyngene var sortert alfabetisk, tilsier dette at dokumentene er gruppert helt i henhold til det som ble gitt inn som seeds.



Figur 28: K-means på dokumenter med mange termer, brukervalgte seeds.

Tabellen viser også at denne klyngefordelingen var helt i henhold til hvordan dokumentene på forhånd var sortert etter emner. Alle dokumentene innenfor hvert emne havnet i samme klynge, med andre ord ingen feilklassifiseringer.

Tabell 9: Dokumentfordeling ved K-means på dokumenter med mange termer, brukervalgte seeds.

Klynge: 0 bil40, bil47,
Klynge: 1 fiske3, fiske4, fiske5, fiske6,
Klynge: 2 fugl1, fugl4, fugl6, fugl7,
Klynge: 3 okonomi01, okonomi07, okonomi09, okonomi14, okonomi28, okonomi32, okonomi42, okonomi48 okonomi50, okonomi74,
Klynge: 4 politikk10, politikk35, politikk41, politikk49, politikk9,

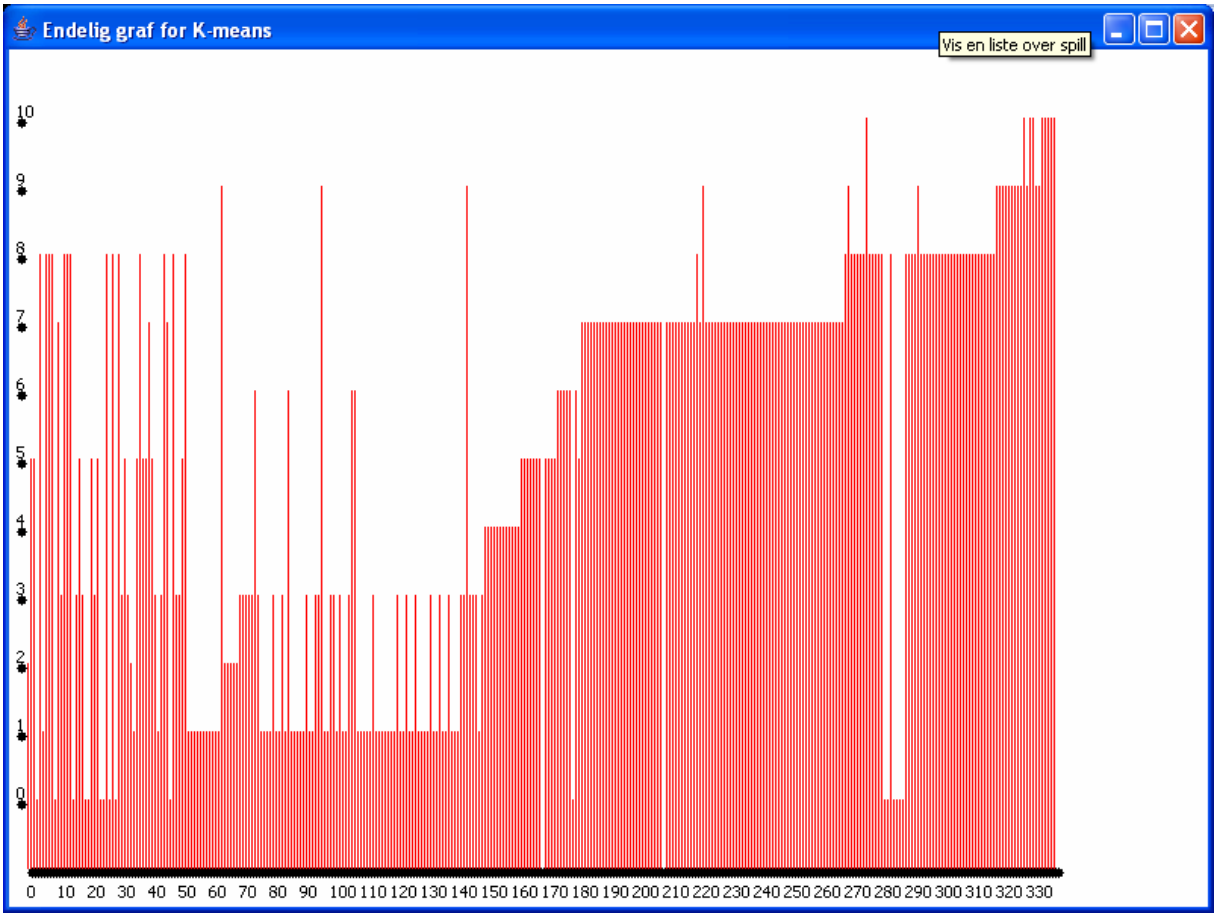
Om en sammenligner resultatet fra testing av dokumenter med få termer og dokumenter med mange termer ser en at dokumenter med mange termer helt klart er gruppert bedre enn de med få termer. Det kan være flere grunner til dette. For det første er det ikke dokumenter innenfor samme emner som er brukt. Det kan være at visse emner, som for eksempel politikk inneholder mange flere termer enn hva emner som fotball gjør. Om en ser på indeksstørrelsen hadde samlingen med dokumenter med få termer en indeksstørrelse på 1941 antall unike termer, mens samlingen med dokumenter med mange termer hadde 4356 unike termer. Grunnen til dette kan som sagt være at det er brukt forskjellige emner, eller mest sannsynlig vil store dokumenter generelt også inneholde flere ulike termer enn små dokumenter gjør.

Dersom det er slik at store dokumenter inneholder flere ulike termer enn små dokumenter, betyr dette at det er lettere å klassifisere dokumenter med stor størrelse. Store dokumenter vil sannsynligvis inneholde flere emnespesifikke termer, med andre ord får en et mye større sammenligningsgrunnlag. Det vil da være lettere å skille dokumenter fra ulike emner fra hverandre.

9.2.5 Testing med ulik grenseverdi (threshold)

For å teste hvilken betydning grenseverdien har for algoritmen, ble det utført tre tester med grenseverdi på 0,1, 0,2 og 0,3. Dette betyr at dokumenter som har similaritet under disse

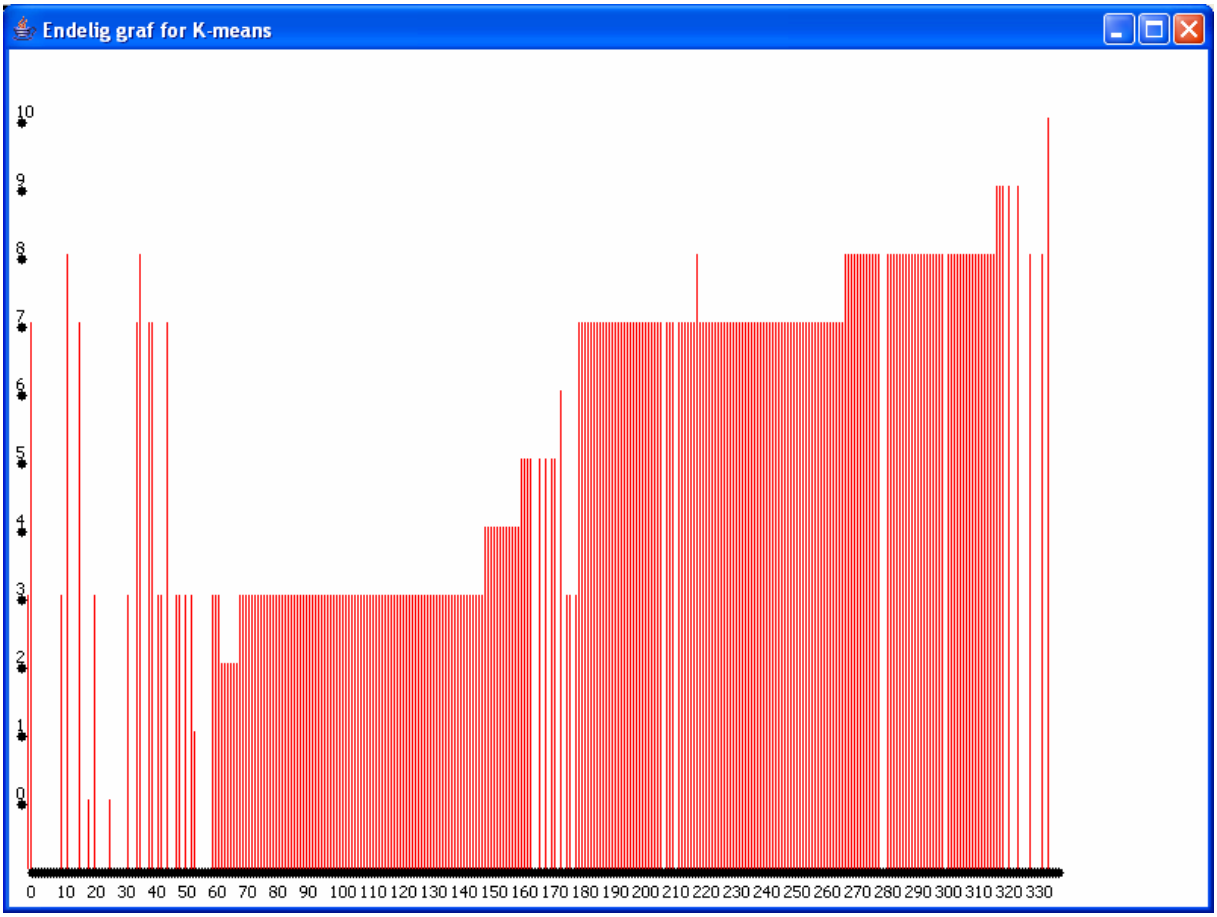
grenseverdiene til alle klyngerepresentantene, blir ikke tilordnet noen klynge. Igjen ble testene utført med boolsk vekting, på hele dokumentsamlingen og med 11 brukervalgte seeds.



Figur 29: K-means med grenseverdi 0,1 og brukervalgte seeds.

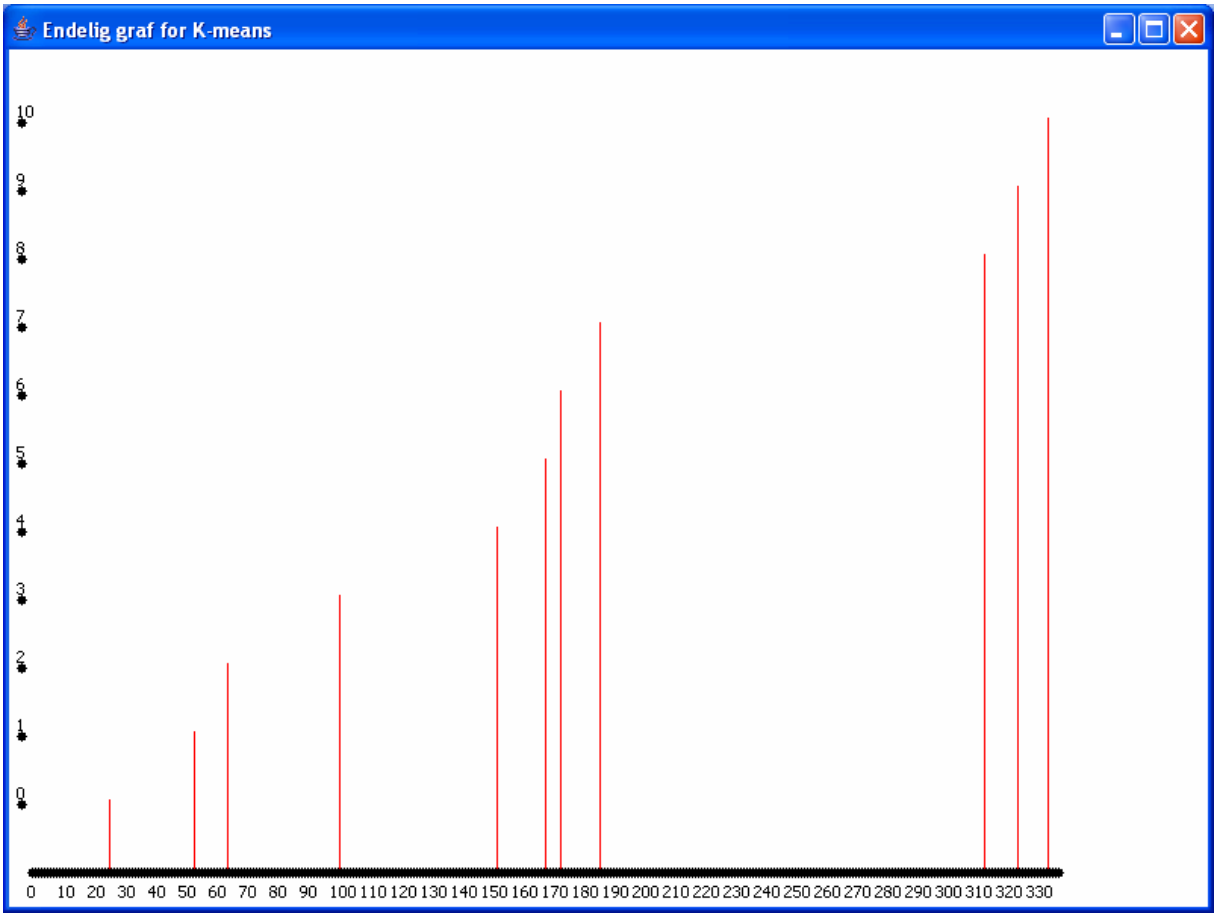
Som Figur 29 viser, er resultatet veldig likt den første testen som beskrevet i seksjon 9.2.1, som var en tilsvarende test, men med grenseverdi på 0,0. Det er her helt lik klyngefordeling bortsett fra at nå er to dokumenter forkastet fordi de var for ulike alle klyngene.

Økes grenseverdien til 0,2 ser vi av Figur 30 at det igjen er samme klyngefordeling, men nå har enda flere dokumenter blitt forkastet.



Figur 30: K-means med grenseverdi 0,2 og brukervalgte seeds.

Som en siste test ble grenseverdien økt til 0,3. Her ble, som vist i Figur 31, alle dokumenter bortsett fra de som ble satt til klyngerepresentanter forkastet.



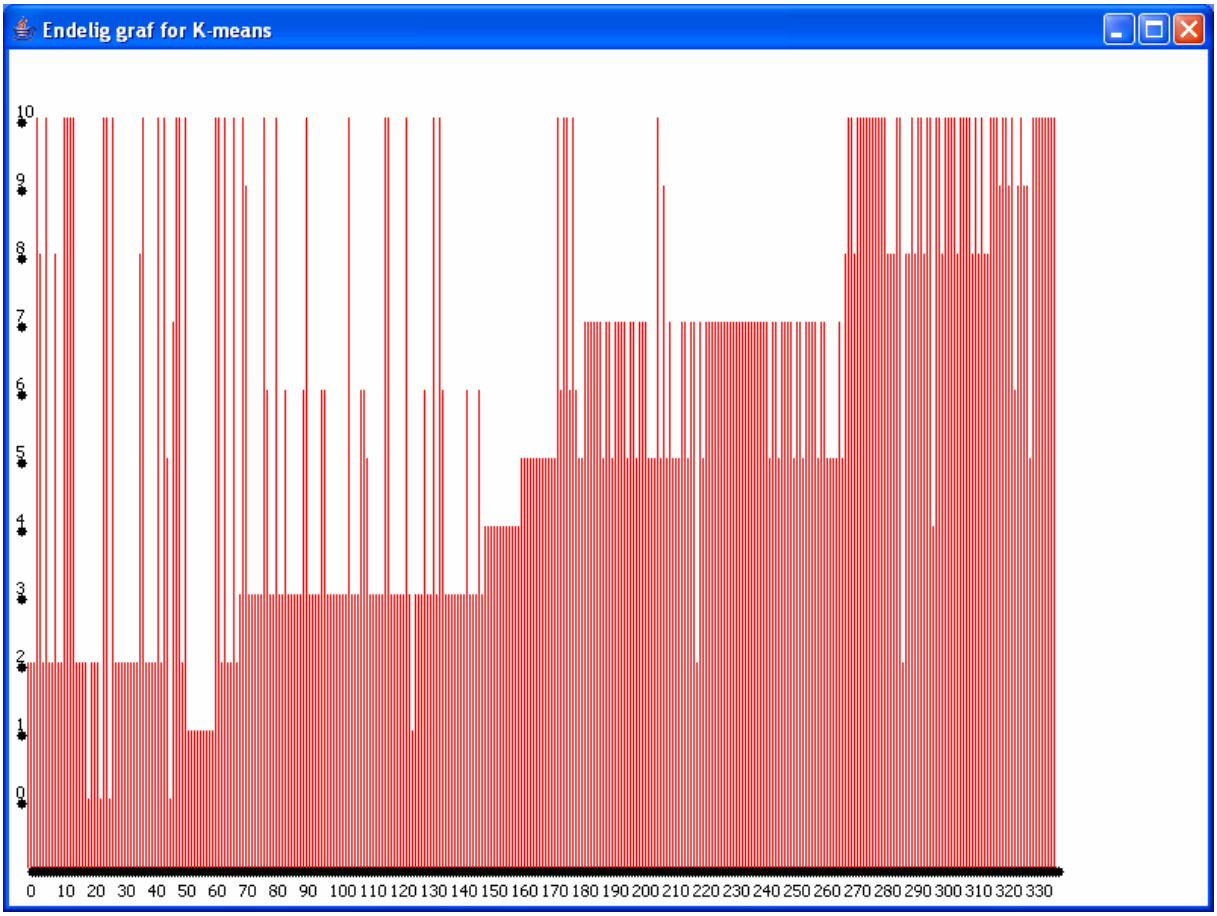
Figur 31: K-means med grenseverdi 0,3 og brukervalgte seeds.

En ser altså her at i denne samlingen ligger similariteten mellom dokumentene og klyngerepresentantene mellom 0 og opp mot 0,3. Det er ingen dokumenter som har likhet over 0,3 med sin respektive klyngerepresentant. Dette betyr at om man skal bruke grenseverdier bør man ved bruk av boolsk vekting variere dette på lave verdier, helst under 0,3. Noe av grunnen til at det er så lite variasjon på disse likhetene er at ved boolsk vekting er vektene enten 0 eller 1, noe som fører til mindre varierte likhetsmål.

9.2.6 Bruk av termfrekvensvekting

I de foregående testene ble det brukt boolsk vekting. Dette innebærer at alle vektene i vektorene som representerer dokumentene enten er 0 eller 1. For å se hva slags betydning disse vektene har, er det implementert en annen type vekting, termfrekvensvekting (se seksjon 5.3). Denne lar alle vektene for hver term i et dokument angi antall termen forekommer i det aktuellet dokumentet.

Resultat fra denne kjøringen er vist i Figur 32. Om en sammenligner grafen fra denne kjøringen med grafen for kjøring tilsvarende test, klyngeanalyse på alle dokumentene i samlingen med brukervalgte seeds og bed bruk av boolsk vekting, ser en at denne kjøringen med termfrekvensvekting ikke gir like god klyngefordeling. Det er nå dårligere ”trappetrinn”-framstilling, og flere feilklassifiseringer.



Figur 32: K-means med brukervalgte seeds og termfrekvensvekting.

Ser en på den tilhørende tabellen (Tabell 10), ser en at i likhet med den tidligere tilsvarende kjøringen blir bildokumentene også nå splittet opp i flere klynger. Denne gangen er det enda færre bildokumenter som havnet i ”riktig” klynge. I tillegg er nå økonomidokumentene oppdelt i to klynger, mens de var i en klynge ved den tidligere tilsvarende testen, og fotballdokumentene mer fordelt.

Tabell 10: Dokumentfordeling ved K-means med brukervalgte seeds og termfrekvensvekting.

Klynge: 0 bil28, bil31, bil34, bil52,
Klynge: 1 data1, data10, data11, data2, data3, data4, data5, data6 data7, fotball61,
Klynge: 2 bil1, bil10, bil11, bil14, bil16, bil17, bil19, bil2 bil24, bil25, bil26, bil27, bil29, bil3, bil30, bil36 bil37, bil38, bil39, bil4, bil40, bil41, bil42, bil45 bil46, bil47, bil48, bil5, bil8, fiske, fiske4, fiske5 fiske7, okonomi40, politikk27,
Klynge: 3 fotball1, fotball12, fotball13, fotball14, fotball15, fotball16, fotball19, fotball2 fotball21, fotball22, fotball24, fotball25, fotball26, fotball27, fotball28, fotball30 fotball31, fotball32, fotball33, fotball36, fotball37, fotball38, fotball39, fotball4 fotball40, fotball41, fotball43, fotball44, fotball45, fotball49, fotball5, fotball50 fotball51, fotball52, fotball55, fotball56, fotball57, fotball58, fotball59, fotball60 fotball62, fotball63, fotball64, fotball66, fotball67, fotball69, fotball71, fotball72 fotball73, fotball74, fotball75, fotball76, fotball77, fotball79, fotball8, fotball80 fotball9,
Klynge: 4 foto1, foto10, foto11, foto12, foto2, foto3, foto4, foto5 foto6, foto7, foto8, foto9, politikk36,
Klynge: 5 bil51, fotball48, fugl1, fugl10, fugl11, fugl12, fugl2, fugl3 fugl4, fugl5, fugl6, fugl7, fugl8, fugl9, okonomi01, okonomi02 okonomi09, okonomi12, okonomi17, okonomi20, okonomi24, okonomi25, okonomi26, okonomi28 okonomi30, okonomi32, okonomi33, okonomi34, okonomi37, okonomi42, okonomi64, okonomi67 okonomi72, okonomi75, okonomi80, okonomi83, okonomi84, okonomi85, okonomi86, okonomi88 religion1,
Klynge: 6 fotball18, fotball23, fotball29, fotball34, fotball35, fotball46, fotball47, fotball65 fotball70, fotball78, fotball81, håndball2, håndball5, håndball7, psyk5,
Klynge: 7 bil53, okonomi03, okonomi04, okonomi05, okonomi06, okonomi07, okonomi08, okonomi10 okonomi11, okonomi13, okonomi14, okonomi15, okonomi16, okonomi18, okonomi19, okonomi21 okonomi22, okonomi23, okonomi31, okonomi35, okonomi36, okonomi38, okonomi39, okonomi41 okonomi43, okonomi44, okonomi45, okonomi46, okonomi47, okonomi48, okonomi49, okonomi50 okonomi51, okonomi52, okonomi53, okonomi54, okonomi55, okonomi56, okonomi57, okonomi58 okonomi59, okonomi60, okonomi61, okonomi62, okonomi63, okonomi65, okonomi66, okonomi68 okonomi69, okonomi70, okonomi71, okonomi73, okonomi74, okonomi76, okonomi77, okonomi78 okonomi79, okonomi81, okonomi82, okonomi87,
Klynge: 8 bil13, bil18, bil43, politikk1, politikk12, politikk22, politikk23, politikk24 politikk28, politikk29, politikk30, politikk33, politikk39, politikk43, politikk48, politikk5 politikk6, politikk7,
Klynge: 9 fotball11, okonomi29, psyk10, psyk3, psyk6, psyk8, psyk9,
Klynge: 10 bil12, bil15, bil20, bil21, bil22, bil23, bil32, bil33 bil35, bil44, bil49, bil50, bil6, bil7, bil9, data8 data9, fiske3, fiske6, fotball10, fotball17, fotball20, fotball3, fotball42 fotball53, fotball54, fotball6, fotball68, fotball7, håndball1, håndball3, håndball4 håndball6, okonomi27, politikk10, politikk11, politikk13, politikk14, politikk15, politikk16 politikk17, politikk18, politikk19, politikk2, politikk20, politikk21, politikk25, politikk26 politikk3, politikk31, politikk32, politikk34, politikk35, politikk37, politikk38, politikk4 politikk40, politikk41, politikk42, politikk44, politikk45, politikk46, politikk47, politikk49 politikk50, politikk8, politikk9, psyk1, psyk11, psyk2, psyk4, psyk7 religion2, religion3, religion4, religion5, religion6, religion7, religion8, religion9

Det er med andre ord en generell dårligere klyngefordeling enn ved tilsvarende test med boolsk vekting. Noe av grunnen til dette kan være at selv om termfrekvensvekting tar hensyn til at termer forekommer flere ganger i samme dokument, blir dette for mye vektlagt. Om en skal ta med termfrekvens i sammenligningen av dokumentene, burde man sannsynligvis bruke en normalisert termfrekvensvekting som beskrevet tidligere i seksjon 5.3. Men som igjen testene med boolsk vekting viste, ser ikke dette ut til å være nødvendig for å få gode resultater. Så hvorvidt hvilken testing en burde bruke gjenstår å teste i framtidige applikasjoner.

9.3 Testing av agglomerativ hierarkisk algoritme

Denne algoritmen er meget ressurskrevende og er derfor testet med bare deler av dokumentsamlingen, slik at en kan se hvordan den organiserer klyngene underveis. Det er her testet med 6 dokumenter innenfor hvert emne med boolsk vekting.

Et utvalg av dokumentfordeling i forhold til klynger i hver iterasjon er vist i Tabell 11, Tabell 12 og Tabell 13. Som disse tabellene viser er det to religionsdokument som ligger nærest hverandre i forhold til resten av dokumentene, dvs. har størst similaritet etter første iterasjon. Disse dokumentene havner derfor i samme klynge. Etter iterasjon 21 kan en se at både noen fiskedokumenter, fotballdokumenter, fotodokumenter og økonomidokumenter har havnet i samme klynge. Etter iterasjon 51 ser en mer tydelige klyngefordelinger. Nå er klyngeantallet redusert fra 66 som var antallet ved start, til 15 klynger. En del bildokumenter, datadokumenter og fotballdokumenter har nå havnet i samme klynge. Dette er ikke helt i henhold til hvordan dokumentene var klassifisert på forhånd, men algoritmen baserer seg som kjent kun på sammenligning av de tilhørende vektorene. Dette betyr at disse dokumentene hadde større similaritet enn for eksempel noen fugledokumenter hadde med hverandre. Iterasjon 64 og 65 viser den aller siste sammenslåingen av klynger. Som en ser av tabellen er det et fugledokument, fugl5 som var mest ulik alle andre dokumenter og som dermed ble slått sammen med en annen klynge helt sist.

Hele denne analysen, dvs. klyngefordelingen etter hver iterasjon er lagd ved elektronisk.

Tabell 11: Dokumentfordeling ved hierarkisk klyngeanalyse (iterasjon 1 og 21).

ITERASJON 1

Klynge 0	**Klynge 10**	**Klynge 20**	**Klynge 30**	**Klynge 40**	**Klynge 50**	**Klynge 61**
bil1	data5	fotball3	fugl1	håndball5	politikk3	religion2
Klynge 1	**Klynge 11**	**Klynge 21**	**Klynge 31**	**Klynge 41**	**Klynge 51**	**Klynge 62**
bil2	data6	fotball4	fugl2	håndball6	politikk4	religion3
Klynge 2	**Klynge 12**	**Klynge 22**	**Klynge 32**	**Klynge 42**	**Klynge 52**	**Klynge 63**
bil3	fiske	fotball5	fugl3	okonomi01	politikk5	religion4
Klynge 3	**Klynge 13**	**Klynge 23**	**Klynge 33**	**Klynge 43**	**Klynge 53**	**Klynge 64**
bil4	fiske3	fotball6	fugl4	okonomi02	politikk6	religion5, religion1
Klynge 4	**Klynge 14**	**Klynge 24**	**Klynge 34**	**Klynge 44**	**Klynge 54**	**Klynge 65**
bil5	fiske4	foto1	fugl5	okonomi03	psyk1	religion6
Klynge 5	**Klynge 15**	**Klynge 25**	**Klynge 35**	**Klynge 45**	**Klynge 55**	
bil6	fiske5	foto2	fugl6	okonomi04	psyk2	
Klynge 6	**Klynge 16**	**Klynge 26**	**Klynge 36**	**Klynge 46**	**Klynge 56**	
data1	fiske6	foto3	håndball1	okonomi05	psyk3	
Klynge 7	**Klynge 17**	**Klynge 27**	**Klynge 37**	**Klynge 47**	**Klynge 57**	
data2	fiske7	foto4	håndball2	okonomi06	psyk4	
Klynge 8	**Klynge 18**	**Klynge 28**	**Klynge 38**	**Klynge 48**	**Klynge 58**	
data3	fotball1	foto5	håndball3	politikk1	psyk5	
Klynge 9	**Klynge 19**	**Klynge 29**	**Klynge 39**	**Klynge 49**	**Klynge 59**	
data4	fotball2	foto6	håndball4	politikk2	psyk6	

ITERASJON 21

Klynge 0	**Klynge 32**	**Klynge 61**
bil1	fugl3	religion2
Klynge 1	**Klynge 34**	**Klynge 62**
bil2	fugl5	religion3
Klynge 2	**Klynge 35**	**Klynge 63**
bil3	fugl6, fugl4, fugl1	religion4
Klynge 3	**Klynge 36**	**Klynge 64**
bil4	håndball1	religion5, religion1
Klynge 4	**Klynge 37**	**Klynge 65**
bil5	håndball2	religion6
Klynge 5	**Klynge 39**	
bil6	håndball4, håndball3	
Klynge 6	**Klynge 40**	
data1	håndball5	
Klynge 7	**Klynge 41**	
data2	håndball6	
Klynge 8	**Klynge 42**	
data3	okonomi01	
Klynge 9	**Klynge 47**	
data4	okonomi06, okonomi05, okonomi04, okonomi03, okonomi02	
Klynge 11	**Klynge 48**	
data6, data5	politikk1	
Klynge 12	**Klynge 49**	
fiske	politikk2	
Klynge 13	**Klynge 50**	
fiske3	politikk3	
Klynge 16	**Klynge 51**	
fiske6	politikk4	
Klynge 17	**Klynge 53**	
fiske7, fiske5, fiske4	politikk6, politikk5	
Klynge 19	**Klynge 54**	
fotball2	psyk1	
Klynge 23	**Klynge 55**	
fotball6, fotball5, fotball4, fotball3, fotball1	psyk2	
Klynge 24	**Klynge 57**	
foto1	psyk4	
Klynge 29	**Klynge 58**	
foto6, foto5, foto4, foto3, foto2	psyk5	
Klynge 31	**Klynge 59**	
fugl2	psyk6, psyk3	

Tabell 12: Dokumentfordeling ved hierarkisk analyse (iterasjon 51 og 61).

ITERASJON 51

Klynge 12
fiske
Klynge 17
fiske7, fiske5, fiske4, fiske6, fiske3
Klynge 23
fotball6, fotball5, fotball4, fotball3, fotball1, bil1, fotball2, data6, data5, data4
data3, data2, data1, bil6, bil5, bil4, bil2, bil3
Klynge 31
fugl2
Klynge 32
fugl3
Klynge 34
fugl5
Klynge 35
fugl6, fugl4, fugl1
Klynge 41
håndball6, håndball5, håndball4, håndball3, håndball2, håndball1
Klynge 47
okonomi06, okonomi05, okonomi04, okonomi03, okonomi02, okonomi01
Klynge 48
politikk1
Klynge 49
politikk2
Klynge 51
politikk4
Klynge 53
politikk6, politikk5, politikk3, foto6, foto5, foto4, foto3, foto2, foto1
Klynge 57
psyk4
Klynge 65
religion6, religion4, religion5, religion1, religion3, religion2, psyk6, psyk3, psyk2, psyk1
psyk5

ITERASJON 61

Klynge 23
fotball6, fotball5, fotball4, fotball3, fotball1, bil1, fotball2, data6, data5, data4
data3, data2, data1, bil6, bil5, bil4, bil2, bil3, fiske7, fiske5
fiske4, fiske6, fiske3, fiske
Klynge 31
fugl2
Klynge 32
fugl3
Klynge 34
fugl5
Klynge 65
religion6, religion4, religion5, religion1, religion3, religion2, psyk6, psyk3, psyk2, psyk1
psyk5, okonomi06, okonomi05, okonomi04, okonomi03, okonomi02, okonomi01, psyk4, politikk6, politikk5
politikk3, foto6, foto5, foto4, foto3, foto2, foto1, politikk4, politikk2, politikk1
håndball6, håndball5, håndball4, håndball3, håndball2, håndball1, fugl6, fugl4, fugl1

Tabell 13: Dokumentfordeling ved hierarkisk analyse (iterasjon 64 og 65).

ITERASJON 64

Klynge 34
fugl5
Klynge 65
religion6, religion4, religion5, religion1, religion3, religion2, psyk6, psyk3, psyk2, psyk1
psyk5, okonomi06, okonomi05, okonomi04, okonomi03, okonomi02, okonomi01, psyk4, politikk6, politikk5
politikk3, foto6, foto5, foto4, foto3, foto2, foto1, politikk4, politikk2, politikk1
håndball6, håndball5, håndball4, håndball3, håndball2, håndball1, fugl6, fugl4, fugl1, fugl3
fugl2, fotball6, fotball5, fotball4, fotball3, fotball1, bil1, fotball2, data6, data5
data4, data3, data2, data1, bil6, bil5, bil4, bil2, bil3, fiske7
fiske5, fiske4, fiske6, fiske3, fiske

ITERASJON 65

Klynge 65
religion6, religion4, religion5, religion1, religion3, religion2, psyk6, psyk3, psyk2, psyk1
psyk5, okonomi06, okonomi05, okonomi04, okonomi03, okonomi02, okonomi01, psyk4, politikk6, politikk5
politikk3, foto6, foto5, foto4, foto3, foto2, foto1, politikk4, politikk2, politikk1
håndball6, håndball5, håndball4, håndball3, håndball2, håndball1, fugl6, fugl4, fugl1, fugl3
fugl2, fotball6, fotball5, fotball4, fotball3, fotball1, bil1, fotball2, data6, data5
data4, data3, data2, data1, bil6, bil5, bil4, bil2, bil3, fiske7
fiske5, fiske4, fiske6, fiske3, fiske, fugl5

10 Konklusjon

Med vår definisjon av lærings- og kunnskapsobjekt, viser det seg at det er mulig å automatisk kunne organisere kunnskapsobjektene ved hjelp av klyngeanalyse. Dette forutsetter at til et hvert kunnskapsobjekt tilhører det en tekstlig beskrivelse. Denne beskrivelsen kan enkelt la seg vektorisere og dermed indekseres. På denne måten slipper man å benytte metadata som viser seg å være meget subjektivt og situasjonsavhengig. Selv om den tekstlige beskrivelsen av kunnskapsobjektene også er subjektive, er dette en mye mer utfyllende representasjon av objektene, enn hva bruk av metadata vil være. Vektoriseringen er helt statistisk og gir heller ikke rom for semantisk tolkning. Dette betyr at kunnskapsobjektene blir organisert uavhengig av hva slags læringskonsept (eller læringsobjekt) de er satt i.

Kunnskapsobjektene vil i framtiden mest sannsynlig bli organisert i samlinger, dvs. repositories. En ser her for seg profesjonelle produsenter som utvikler og har ansvaret for objekter, mens faglige veiledere, lærende og andre brukere er abonnenter til repositoriene.

Ettersom dokumenter ofte blir klassifisert ut fra en mental modell, altså ut fra hvordan vi mennesker tolker innholdet, er det ikke alltid at en klyngeanalyse basert på ren statistikk vil gi samme resultat som de forhåndsdefinerte klassifiseringene. I en mental modell kan et dokument passe inn i flere emner, for eksempel kan et dokument som handler om kjøregodtgjørelse klassifiseres som bildokument eller skattedokument, avhengig av hvilken sammenheng dokumentet forekommer i. Med klyngeanalyse benyttes likhetsmål som kun ser termsyntaks, og ikke den semantiske betydningen. Dette kan altså føre til en uoverensstemmelse mellom forhåndsklassifisert emne og gruppefordeling etter en klyngeanalyse. Bakgrunnen til forfatteren har også en del å si. En forfatter med ekspertise om fugler ville sannsynligvis beskrevet et villmarksområde sett fra et annet perspektiv og dermed på en annen måte enn hva en forfatter med bakgrunn i skogsbruk ville gjort.

I en læringssammenheng derimot, vil det sannsynligvis være domenebaserte emner og dermed mer bruk av fagterminologi, altså en mindre variert terminologi. Dette vil føre til at emnefordelingen etter en mental modell også blir mer entydig, og dermed sannsynligvis mer lik klyngeanalyseringen.

Som testene av K-means viser, fungerer klyngeanalyse best med brukervalgte initielle klyngerepresentanter. Dette betyr at en bruker må ha en viss kjennskap til kunnskapsobjektsamlingen. Dette anses ikke som et problem i læringssammenheng da det vil være domenebaserte repositories hvor brukerne kjenner de konseptene som kunnskapsobjektene beskriver. Videre viser testene at de dokumentene med flest termer hadde best gruppering i forhold til hva som var forhåndsdefinerte emner. Med andre ord bør ikke de tekstlige beskrivelsene for et kunnskapsobjekt være for kort. En ser for seg at gunstig størrelse er på cirka en A4-side.

Antallet dokumenter fra hvert emne spiller relativ liten rolle for resultatet av klyngeanalyseringen, da de nye klyngerepresentantene hele tiden blir satt til å være gjennomsnittet av alle tilhørende dokument for klyngen. Videre viser testingen at ulike grenseverdier har lite å si for klyngeanalyse med boolsk vekting. Hovedgrunnen til dette er at det ikke tas hensyn til termfrekvens. To dokument hvor en term blir gjentatt veldig ofte vil ikke nødvendigvis få veldig stor likhet, fordi begge dokumentene har et sett av andre termer som ikke er like.

Klyngeanalysen gir til slutt et godt grunnlag for en visuell presentasjon. En kan benytte likhetsmålet mellom ulike klynger for å tegne opp avstander, og dermed presentere kunnskapsobjektene i en tredimensjonal graf, eller et navigerbart konseptkart.

11 Veien videre

Det som bør arbeides med vider i forbindelse med klyngeanalyse og organisering av læringsmateriell generelt, er å knytte sammen arbeidsbok, klyngeanalyse og en tredimensjonal presentasjon til et rammeverk hvor både den lærende og faglig veileder kan organisere og selv utvikle kunnskapsobjekter og læringsobjekter. I tillegg bør en jobbe med å få knyttet latent semantisk indeksering til objektsamlingene slik at uerfarne også skal kunne kommunisere med systemet og kunne jobbe med kunnskapsobjektene uten å måtte lære seg alle begreper og uttrykk først. En ser for seg at dette blir et læringsmiljø sett fra den lærendes ståsted, i motsetning til kursets og faglig veileders ståsted som de fleste systemer i dag tar utgangspunkt i.

Som en ser av testingen med klyngeanalysering, kan det føre til uoverensstemmelse mellom hva som er forhåndsdefinerte emner og klassifiseringer, og klassifisering etter analyse. Hovedgrunnen til dette er at de forhåndsdefinerte emnene er klassifisert ved bruk av en mental modell, mens klyngeanalyseringen benytter statistikk. Det en kan jobbe videre med i denne sammenheng, er å se på om latent semantisk indeksering kan minke disse forskjellene ved at man foretar en termreduksjon ved indekseringen.

Et annet viktig moment for framtidig arbeid, er hvordan kunnskapsobjektene skal utvikles og distribueres. Det er på dette området muligheter for profesjonelle aktører, da særlig med tanke på forlag, og få i gang en produksjonslinje for dette. For nettbaserte læringsmiljø vil det være gunstig om kunnskapen blir produsert i andre former enn vanlige sekvensielle lærebøker.

12 Referanser

ART, “What is ART”, *Internet FAQ Archives.* Lokalisert 29.05.06 på Verdensveven: http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-19.html.

Baeza-Yates, R. & Ribeiro-Neto, B. (1999) “Modern Information Retrieval”, Addison Wesley, New York: ACM Press.

BCPL (Basic Combined Programming Language). Lokalisert 04.04.06 på Verdensveven: http://www.cl.cam.ac.uk/users/mr/BCPL.html.

Clatworthy, J., Buick, D., Hankins, M., Weinman, J., & Horne, R. (2005). “The use and reporting of cluster analysis in health psychology: A review”. *British Journal of Health Psychology* 10: 329-358.

Enclyclopedia of Learning Technology. Lokalisert 09.05.06 på Verdensveven: http://coe.sdsu.edu/eet/Admin/TOC/.

Faloutsos & Oard, “A survey of information retrieval and filtering methods”. Lokalisert 04.04.06 på Verdensveven: http://www.enee.umd.edu/medlab/filter/papers/survey.ps.

Hamel, C. J., & Ryan-Jones, D. (2002). “Designing instruction with learning objects”. *International Journal of Educational Technology (IJET), 3(1).* Lokalisert 08.05.06 på Verdensveven: http://www.ao.uiuc.edu/ijet/v3n1/hamel/index.html

Heyer, L.J., Kruglyak, S. and Yooseph, S., “Exploring Expression Data: Identification and Analysis of Coexpressed Genes”, *Genome Research 9*: 1106-1115.

Holme, Arvid (2006) ”Bruk av kunnskaps-/læringsobjekter I nettbasert læring og organisering av kunnskap”. Kunnskaps- og læringsobjekter i nettbasert læring.

Holme, Arvid (2000) Informasjonsgjenfinning, kompendium.

Husby, Ole (1997) "Foredrag ved Kunnskapsorganisasjonsdagene”, Høgskolen i Oslo, BIBSYS. Lokalisert 03.03.06 på Verdensveven: http://www.bibsys.no/meta/korg97.html.

IBM Glossary. Lokalisert 10.05.06 på Verdensveven: http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.ii.of.doc/common/iiysgloss.htm.

Ilstad, Steinar (2002) Generell Psykologi, Tapir Akademisk Forlag, Trondheim: 278-280.

K-means Clustering, “A Tutorial on Clustering Algorithms”. Lokalisert 30.03.06 på Verdensveven: http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/kmeans.html.

Larsen, Jarle (2006) Læringsobjekt.

Leksikon.org: Konstruktivisme. Lokalisert 06.04.06 på Verdensveven: http://www.leksikon.org/art.php?n=5014.

Porterstemming. Lokalisert 29.03.06 på Verdensveven: http://www.tartarus.org/martin/PorterStemmer/.

Problembasert læring, ”Generelt om problembasert læring”, Fagområdet for universitetspedagogikk, UiO. Lokalisert 09.05.06 på Verdensveven: http://www.pfi.uio.no/uniped/gpm/pbl.html.

Rekkedal, Torstein (1999). ”Pedagogiske utfordringer knyttet til læring via nett”, *foredrag på NIPAs konferanse "Fra kunnskap til handling"*, personalforum 99, Lillehammer 9.-11-november 1999. Lokalisert 12.05.06 på Verdensveven: http://www.nettskolen.com/pub/artikkel.xsql?artid=106.

REN (Research and educational network) (2002) “Lær av Lego”, Om teknologiske standarder i e-læring, Fase 1 i RENs prosjekt Pedagogiske kvalitetskriterier for nettbasert læring. Lokalisert 08.05.06 på Verdensveven: http://www2.invanor.no/upload/extranet/ren/LaeravLego.pdf.

UKeU, London. Lokalisert 09.05.06 på Verdensveven: http://www.univ-montp3.fr/praxiling/~rachel/spip/IMG/pdf/7_Darby.pdf.

Wikipedia: BCPL. Lokalisert 30.03.06 på Verdensveven: http://en.wikipedia.org/wiki/BCPL.

Wikipedia: Data clustering. Lokalisert 01.04.06 på Verdensveven: http://en.wikipedia.org/wiki/Data_clustering.

Wikipedia: Metadata. Lokalisert 30.03.06 på Verdensveven: http://en.wikipedia.org/wiki/Metadata.

Wikipedia: Logic. Lokalisert 10.05.06 på Verdensveven: http://en.wikipedia.org/wiki/Logic.

Wikipedia: Proximity Search. Lokalisert 10.05.06 på Verdensveven: http://en.wikipedia.org/wiki/Proximity_search.

Wiley, David A. (2000) “Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy” i *The Instructional Use of Learning Objects: Online Version*. Lokalisert 08.05.06 på Verdensveven: http://reusability.org/read/chapters/wiley.doc.

Øiestad, G. (1993) ”Motivasjon og dataspill” i *Data i skolen*, Aschehoug/Apple.