NTNU
Det skapende universitet

# Ontologikonstruksjon for Statoil

Evaluering av metoder for konstruksjon av søkeontologier

**Geir Øyvin Grimnes**

Master i datateknikk

Oppgaven levert: Juni 2006

Hovedveileder: Jon Atle Gulla, IDI

Biveileder(e): Terje Brasethvik, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

# Oppgavetekst

KUDOS er et samarbeidsprosjekt mellom Statoil og NTNU innenfor området semantisk informasjonsforvaltning. Statoil håndterer i sitt daglige arbeide store dokumentsamlinger og har store ambisjoner for  forretningsstøttende informasjonsforvaltning og IT-arkitektur. KUDOS prosjektet fokuserer på bruken av ontologier for å representere domenespesifikk kunnskap med sikte på å oppnå en mer semantisk tilnærming til dokumentgjenfinning og forvaltning. Prosjektet studerer også bruken av avanserte tekstbaserte datagruvedrifts metoder for å ekstrahere kunnskap fra dokumentsamlinger og relatere disse til prosedyrebeskrivelser og styrende dokumenter i Statoil.

Målet for oppgaven er å utvikle en ontologi for et bestemt domene og en tilhørende dokumentsamling i Statoil. Ontologien skal modelleres i OWL ved hjelp av Protégé, en ontologi-editor fra Stanford. Tanken er at ontologien skal reflektere innholdet i den gitte dokumentsamlingen og skal kunne være til nytte for brukere som søker etter dokumenter og informasjon fra denne. Den utviklede ontologien skal evalueres med hensyn på tekstsøk i dokumentsamlingen.

Oppgaven gitt: 20. januar 2006
Hovedveileder: Jon Atle Gulla, IDI

# Sammendrag

I dagens industriorganisasjoner er store mengder kunnskap og informasjon spredd utover organisasjonen gjennom ulike dokumenter som er lagret digitalt. I store organisasjoner vil mengden dokumenter øke svært raskt, og kravet om effektiv tilgang og gjennfinning av informasjon er derfor sterkt økende. Dette gjelder spesielt for kunnskapsintensive bedrifter hvor god informasjonsforvaltning ofte er en av nøkkelfaktorene for å oppnå suksess. På bakgrunn av dette har Statoil ASA og gruppe for informasjonssystemer ved IDI/NTNU inngått et samarbeidsprosjekt, KUDOS (Kunnskap i Dokumenter i Statoil), som skal bygge systemer for informasjonsforvaltning, understøttet av semantisk søk og uttrekking av informasjon fra dokumenter. Bruk av ontologier i søkesammenheng er sentralt i prosjektet.

I arbeidet er det blitt konstruert to ontologier; en rent manuelt, og en semi-automatisk. Begge ontolgiene har bakgrunn i domenet *project management* og er realisert i Web Ontology Language (OWL) ved hjelp av ontologieditoren Protégé. Jobben med å konstruere den manuelle ontologien er gjort ved å inspisere domenespesifikke dokumenter, samt å holde møter med en domeneekspert for å komme frem til viktige begreper og relasjoner som beskriver domenet. I den semi-automatiske tilnærmingen er det brukt et system for uttrekning av nøkkelfraser fra tekst, utviklet ved IDI, til å komme med forslag til relevante begreper og fraser fra dokumentsamlingen. Forslagene er så blitt kontrollert av en domeneekspert før selve ontologibyggingen.

For å evaluere ontologienes kvalitet oppimot hverandre har man valgt en tredelt evaluering:

- Sammenligning og evaluering av elementene i ontologiene.
- Evaluering av ontologiene utifra et sett kvalitetskriterier.
- Evaluering av ontologiene utifra bruk i praktisk søk.

På grunn av få evaluatorer og liten dokumentsamling er variasjonene i resultatene fra de to tilnærmingene liten. Det er dermed vanskelig å dra noen faste konklusjoner, men man ser likevel en klar tendens til at den manuelle ontologien jevnt over har et høyere kvalitetsnivå enn den semi-automatiske ontologien i alle de tre evalueringsfasene.

# Forord

Denne rapporten er, sammen med to ontologier, resultatet av diplomoppgaven i faget TDT4900 Masteroppgave ved Institutt for Datateknikk og Informasjonsvitenskap (IDI) ved Norges teknisk-naturvitenskaplige universitet (NTNU) våren 2006. Oppgaven markerer slutten på det 5-årige sivilingeniørstudiet i datateknikk ved NTNU.

Rapporten er skrevet som en del av KUDOS-prosjektet, et samarbeidsprosjekt mellom Statoil ASA og gruppe for informasjonssystemer ved IDI. Målet for KUDOS er å bygge systemer for informasjonsforvaltning som understøtter semantisk søk og uttrekning av informasjon fra dokumenter.

Jeg vil takke faglærer professor Dr. Jon Atle Gulla og veileder Dr. Terje Brasethvik for god hjelp og veiledning gjennom hele arbeidet. Jeg vil også rette takk til medstudent Hans Olaf Borch og stipendiatene Stein L. Tommasen, Jon Espen Ingvaldsen og Darijus Strasunskas for verdifull hjelp til evaluering av resultater, samt til Dr. Harald Rønneberg og Jan Frode Saasen for å ha bidratt med verdifull domenekunnskap i forbindelse med bygging av de to ontologiene. En spesiell takk går til medstudentene Tarjei Lægreid og Paul Christian Sandal for evaluering av resultater, samt hjelp til oppsett og implementasjon av Apache Lucene.

Geir Øyvin Grimnes

Trondheim, 23.06.2006

# Innhold


**1 Innledning** **1**
- 1.1 Bakgrunn . . . 1
- 1.2 Problem og mulig løsning . . . 1
- 1.3 Oppgaven . . . 2
- 1.4 Mål for oppgaven . . . 3
- 1.5 Rammer for oppgaven . . . 3
- 1.6 Arbeidsmedtode . . . 4
  - 1.6.1 Litteraturstudie . . . 4
  - 1.6.2 "Feltarbeid" . . . 4
  - 1.6.3 Ontologibygging . . . 4
  - 1.6.4 Testing og evaluering . . . 4
- 1.7 Rapportens oppbygning . . . 5

**2 Teoretisk bakgrunn** **7**
- 2.1 Semantisk web . . . 7
  - 2.1.1 Struktur . . . 8
- 2.2 Ontologier . . . 10
  - 2.2.1 OWL - Web Ontology Language . . . 12
  - 2.2.2 OWL Full . . . 13
  - 2.2.3 OWL DL (Description Logic) . . . 13
  - 2.2.4 OWL Lite . . . 14
  - 2.2.5 Sammenlikning av OWL-artene . . . 14
- 2.3 Informasjonsgjenfinning . . . 15
  - 2.3.1 Indeksering og invertert fil . . . 16
  - 2.3.2 Preprosessering . . . 17
  - 2.3.3 IR-modeller . . . 19
  - 2.3.4 Ontologi i søkeapplikasjoner . . . 20

**3 Manuell ontologikonstruksjon** **23**
- 3.1 Manuell begrepsekstraksjon . . . 23
  - 3.1.1 Definer ontologiskop . . . 23
  - 3.1.2 Definering av begreper . . . 24

3.1.3 Gjennomgang av definisjoner . . . 25
3.1.4 Konstruksjon av meta-ontologi . . . 25
3.2 Praktisk gjennomføring . . . 26
3.2.1 PMBOKs innhold og oppbygning . . . 26
3.2.2 Gjennomgang av dokumentet . . . 28
3.2.3 Møte med domeneekspert . . . 28
3.2.4 Lærdom . . . 30

**4 Semi-automatisk ontologikonstruksjon 33**
4.1 Basismetoder for begrepsekstraksjon . . . 33
4.1.1 Begrepsuttrekning . . . 33
4.1.2 Ordklassifisering . . . 34
4.1.3 Assosiasjonsregler . . . 34
4.2 Fraseekstraksjonssystem . . . 34
4.3 Praktisk gjennomføring . . . 35
4.3.1 Lærdom . . . 37

**5 Evaluering 39**
5.1 Evalueringsbakgrunn . . . 39
5.2 Del 1 - Sammenligning av ontologiene . . . 41
5.2.1 Klasser og attributter . . . 41
5.2.2 Begreper og definisjoner . . . 43
5.3 Del 2 - Kvalitetsevaluering . . . 44
5.3.1 Kvalitetskrav . . . 44
5.3.2 Konsistenssjekk . . . 48
5.4 Del 3 - IR-evaluering . . . 49
5.4.1 Aspekter ved IR-evaluering . . . 49
5.4.2 Evalueringstilnærming . . . 50
5.4.3 Resultater og diskusjon . . . 55
5.4.4 Ontologibasert søk vs. ordinært fritekstsøk . . . 60
5.4.5 Oppsummering . . . 60

**6 Konklusjon og videre arbeid 63**
6.1 Konklusjon . . . 63
6.2 Videre arbeid . . . 64

**A Utvidelser av spørring 65**

**B Resultater IR-test 69**

**C Resultat brukerevaluering 75**

**D Ekstraherte ord/fraser 85**

**E Kildekode** **91**

**F Digitalt vedlegg** **97**

# Figurer


2.1 Den lagdelte arkitekturen til semantisk web [McGuineness, 2001] . 8
2.2 Ontologispektrum [McGuineness, 2001] . . . . . 11
2.3 De tre artene av OWL . . . . . 13
2.4 Eksempel på tradisjonelt IR-system . . . . . 16
2.5 Eksempel på invertert fil . . . . . 17
2.6 Eksempel på ontologistøttet IR-system [Thommasen et al., 2005] . 22

3.1 Eksempel på UMMO . . . . . 26
3.2 Struktur PMBOK [(PMI), 2000] . . . . . 27
3.3 Utsnitt av begrepshierarki . . . . . 29
3.4 Utsnitt av attributt-hierarki . . . . . 29
3.5 Protégé OWLViz . . . . . 30

4.1 Arkitekturen til fraseekstraksjonssystemet [Borch, 2005] . . . . . 35
4.2 Brukergrensesnittet i Protégé . . . . . 36

5.1 Strengt IS-A hierarki . . . . . 45
5.2 Dårlig oppbygd hierarki . . . . . 45
5.3 Konsistenssjekk med RacerPro . . . . . 48
5.4 Textpipe Pro brukergrensesnitt . . . . . 51
5.5 Evalueringsskjema . . . . . 54
5.6 Graf over snittet til de normaliserte resultatene . . . . . 55

C.1 Graf over snittet til resultatene . . . . . 83

# Tabeller


2.1 Beskrivelse av OWL . . . . . 12
2.2 Sammenligning av OWL-artene . . . . . 15
2.3 Tre anvendelsesområde for ontologisk informasjon i IR . . . . . 21

4.1 Poengskjema brukt av domeneekspert . . . . . 36

5.1 Antall klasser i manuell ontologi . . . . . 41
5.2 Antall klasser i semi-automatisk ontologi . . . . . 41
5.3 Synonymer i ontologiene . . . . . 42
5.4 Poengskjema brukt av domeneekspert . . . . . 43
5.5 Poengfordeling manuell ontologi . . . . . 43
5.6 Poengfordeling semi-automatisk ontologi . . . . . 43
5.7 Spørringene brukt til evalueringen . . . . . 53
5.8 Poenggivning i forhold til relevans . . . . . 53
5.9 Poenggivning i forhold til posisjon . . . . . 54
5.10 Samlede normaliserte resultater . . . . . 56
5.11 Manuell ontologi . . . . . 59
5.12 Semi-automatisk ontologi . . . . . 59

A.1 Utvidelser fra den manuelle ontologien . . . . . 66
A.2 Utvidelser fra den semi-automatiske ontologien . . . . . 67

C.1 Resultater evaluator 1 . . . . . 76
C.2 Resultater evaluator 2 . . . . . 77
C.3 Resultater evaluator 3 . . . . . 78
C.4 Resultater evaluator 4 . . . . . 79
C.5 Resultater evaluator 5 . . . . . 80
C.6 Resultater evaluator 6 . . . . . 81
C.7 Samlede resultater . . . . . 82

# Kapittel 1

# Innledning

Innledningskapitlet gir et overblikk over problemet som ligger til grunn for oppgaven, samt en beskrivelse av oppgaven i seg selv. Videre presenteres mål og rammer for prosjektet, arbeidmetode, samt rapportens oppbygning.

## 1.1 Bakgrunn

I dagens industriorganisasjoner er store mengder kunnskap og informasjon spredd utover organisasjonen gjennom ulike dokumenter som er lagret digitalt. I store organisasjoner vil mengden dokumenter øke svært raskt, og tall fra Statoil viser en økning på 300 000 MS Office dokumenter hver måned (2004). Som en konsekvens av dette ønsker Statoil å se på muligheten til å ta i bruk state-of-the-art teknikker for mer effektiv gjenfinning av informasjon.

På bakgrunn av dette har Statoil og gruppe for informasjonssystemer ved IDI/NTNU inngått et samarbeidsprosjekt, KUDOS (KUnnskap i DOkumenter i Statoil), som skal ha hovedfokus på bruk av avanserte teknikker for informasjons- og kunnskapsforvaltning i store virksomheter. Målet for prosjektet er å bygge systemer for informasjonsforvaltning som understøtter semantisk søk og uttrekning av informasjon fra dokumenter. Arbeid rundt konstruksjon og bruk av ontologier er sentralt i prosjektet.

## 1.2 Problem og mulig løsning

Store mengder dokumenter gjør jobben med å finne de mest relevante utfra en gitt spørring utfordrende, og i mange sammenhenger er muligheten til effektiv gjennfinning av relevante dokumentene, og dermed kunnskapen i dokumentene,

begrenset. Som en konsekvens av dette kan mye av kunnskapen forbli uutnyttet ved at informasjon ikke blir delt og gjort tilgjengelig på en god nok måte. Dette kan virke svært negativt i kunnskapsintensive bedrifter hvor god informasjonsforvaltning ofte kan være en av nøkkelfaktorene for å oppnå suksess.

For å gjøre søkeapplikasjoner bedre kan man utvikle en ontologi til bruk som støtte for søkemotoren. Ontologier er hierarkiske begrepsstrukturer som beskriver viktige begreper og relasjonene mellom disse begrepene innenfor et gitt domene. For at ontologien skal ha effekt i søkesammenheng må begrepene den inneholder gjenspeile dokumentsamlingen det søkes i, og den skaper således en form for felles forståelse av strukturen til domenet.

## 1.3 Oppgaven

Oppgaven er gitt som en deloppgave i KUDOS prosjektet, og er en videreføring av fordypningsprosjektet "Metoder for ontologikonstruksjon i Statoil" [Grimnes, 2005] fra høsten 2005.

> **Ontologikonstruksjon for Statoil**
> KUDOS er et samarbeidsprosjekt mellom Statoil og NTNU innenfor området semantisk informasjonsforvaltning. Statoil håndterer i sitt daglige arbeide store dokumentsamlinger og har store ambisjoner for forretningsstøttende informasjonsforvaltning og IT-arkitektur. KUDOS prosjektet fokuserer på bruken av ontologier for å representere domenespesifikk kunnskap med sikte på å oppnå en mer semantisk tilnærming til dokumentgjenfinning og forvaltning. Prosjektet studerer også bruken av avanserte tekstbaserte datagruvedrifts metoder for å ekstrahere kunnskap fra dokumentsamlinger og relatere disse til prosedyrebeskrivelser og styrende dokumenter i Statoil.
>
> Målet for oppgaven er å utvikle en ontologi for et bestemt domene og en tilhørende dokumentsamling i Statoil. Ontologien skal modelleres i OWL ved hjelp av Protégé, en ontologi-editor fra Stanford. Tanken er at ontologien skal reflektere innholdet i den gitte dokumentsamlingen, og skal kunne være til nytte for brukere som søker etter dokumenter og informasjon fra denne. Den utviklede ontologien skal evalueres med hensyn på tekstsøk i dokumentsamlingen.

I fordypningsprosjektet "Metoder for ontologikonstruksjon i Statoil" ble teknikker og metoder for å bygge ontologier studert. Det ble også utviklet en metode for manuell konstruksjon av ontologier utfra en gitt dokumenetsamling, med sikte på anvendelser i dokumentsøk.

Fokus for dette arbeide blir derfor å benytte kunnskapen fra høstprosjektet til å manuelt utvikle en ontologi til bruk for dokumentsøk basert en dokumentsamling gitt av Statoil. For å kunne evaluere ontologien vil det også bli bygget en referanseontologi basert på automatisk uttrekning av ord og fraser fra den samme dokumentsamlingen. Begge ontologiene vil deretter bli implementert i et søkemaskineri og evaluert og sammnlignet gjennom et søkeeksperiment.

## 1.4 Mål for oppgaven

Målsetningene for oppgaven kan deles i forskningsmål og resultatmål:

*Forskningsmål:*
Studere mulige effekter av en manuelt utviklet ontoliogi på søk i en gitt dokumentsamling.

*Resultatmål:*
Utvikle to ontologier; en utfra en rent manuell tilnærming, og en semi-automatisk, basert på ekstraherte ord og fraser fra en dokumentsamling.

- Gjøre en sammenligning av de to ontologiene utfra deres oppbygning og struktur.
- Evaluere ontologiene utfra et sett kvalitetskrav for søkeontologier.
- Gjøre en evaluering av ontologiene med bakgrunn i resultater fra tester i praktisk søk.

## 1.5 Rammer for oppgaven

- Verktøy for ontologimodellering (Protégé) og ontologispråk (OWL) er gitt fra KUDOS.
- Konstruksjonen av ontologiene skal ta utgangspunkt i dokumentet "A Guide to the Project Management Body of Knowledge" (PMBOK) [(PMI), 2000] utgitt av Project Management Institute Inc. [(PMI), 2006]
- Konstruksjonen av den semi-automatiske ontologien skal ta utgangspunkt ekstraherte fraser fra et ekstraksjonssystem utviklet av Hans Olaf Borch høsten 2005. [Borch, 2005]

## 1.6 Arbeidsmedtode

Arbeidet har i stor grad vært utført som et praktisk arbeid hvor ontologibygging, testing og evaluering har blitt mest vektlagt. Noe litteraturstudium har naturlig nok også vært nødvendig. Under følger en kort beskrivelse arbeidet.

### 1.6.1 Litteraturstudie

Hovedvekten i litteraturstudiet har vært studier rundt ontologibasert informasjonsgjenfinning, evaluering av ontologier, samt studier av ontologier generelt og semantisk web teknologi. Spesifikasjonen til OWL har også blitt grundig studert.

### 1.6.2 "Feltarbeid"

En del av arbeidet rundt bygging av ontologier ligger rundt det å identifisere de viktige begrepene og relasjonene som beskriver domenet. For å få til dette har møter med domeneeksperter fra Statoil blitt gjennomført.

### 1.6.3 Ontologibygging

Det er i løpet av arbeidet utviklet to ontologier på bakgrunn av den samme dokumentsamlingen. Begge ontologiene er bygd i Protégé (OWL Plugin 2.1 Build 284) en ontologieditor utviklet ved Stanford Medical Informatics.

### 1.6.4 Testing og evaluering

I forbindelse med arbeidet har det blitt implementert et enkelt Lucene-søkesystem. [Apache, 2006]. Dette er brukt til å teste ontologiene i praktisk søk oppimot en dokumentsamling satt sammen som en del av arbeidet. Resultatet fra disse testene danner grunnlaget for evalueringen av arbeidet.

## 1.7 Rapportens oppbygning

Kapittel 2 beskriver relevant teori for oppgaven.

Kapittel 3 beskriver prosessen rundt arbeidet med den manuelt bygde ontologien.

Kapittel 4 beskriver prosessen rundt arbeidet med den semi-automatisk bygde ontologien.

Kapittel 5 presenterer evalueringen av arbeidet.

Kapittel 6 gir en konklusjon til rapporten.

# Kapittel 2

# Teoretisk bakgrunn

Dette kapitlet tar for seg nødvendig bakgrunnsteori for denne rapporten. Først presenteres semantisk web, dets struktur og litt generelt om ontologier. Videre presenteres Web Ontology Language (OWL) som er ontologispråket benyttet i dette arbeidet. Til slutt presenteres ulike metoder og teknikker brukt i informasjonsgjenfinning.

## 2.1 Semantisk web

> "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation."
>
> *Tim Berners-Lee et.al.*
> *The Semantic Web, Scientific American, May 2001* [Berners-Lee et al., 2001]

Semantisk web er neste generasjons World Wide Web (WWW). Dagens web er i hovedsak utviklet for at mennesker skal forstå innholdet, og ikke for at maskiner skal kunne resonere over, og "forstå" innholdet. Med overgangen til semantisk web vil dette problemet minske ved at informasjonsressursene på WWW får mer signifikant data knyttet til seg som informasjonsagenter vil kunne forstå og bruke. Dette vil være svært nyttig ettersom informasjonsmengden som finnes på nettet stadig øker, og det ofte kan være vanskelig å finne frem til nøyaktig det man er ute etter.

Selv om sitatet og avsnittet over gir inntrykk av at semantisk web er en teknologi som hovedsakelig kan brukes på Internett, finnes det et uttall andre bruksområder. Som nevnt i innledningen til rapporten vokser også informasjonsmengden innad

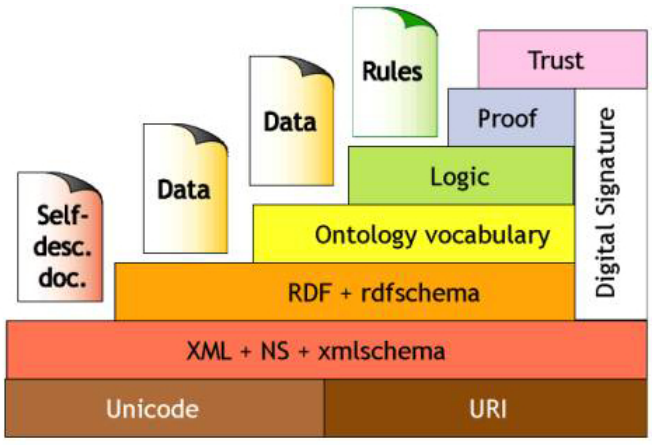
i organisasjoner svært raskt. Derfor vil det også her være bruk for semantisk web teknologi. Et eksempel på dette kan være å knytte ny teknologi oppimot intranett og ERP-systemer med søkefunksjoner for således å hjelpe brukerne av systemet til mer effektivt å finne informasjonen de er ute etter.

For at informasjonsagenter skal kunne bruke den kunnskapen som ligger lagret på nettet og innad i organsasjoner er man avhengig av at informasjonen er strukturert på en optimal måte. Organiseringen må ikke være for kompleks (paradokser kan oppstå), men samtidig være tilstrekkelig strukturert slik at informasjonsagentene kan gjøre seg nytte av dataen. På bakgrunn av dette er en av hovedutfordringene rundt semantisk web å kunne tilby et språk som kan uttrykke både data og regler (logikk) for å resonere i dataen.

De to neste avsnittene tar for seg strukturen i semantisk web som helhet, og hvordan semantiske webtjenester kan brukes for å oppnå bedre interaksjonen mellom mennesker og maskiner.

### 2.1.1 Struktur

Figur 2.1 viser strukturen i semantisk web slik Tim Berners-Lee presenterte den i 2001. Pyramiden viser hvordan de ulike elementene i arkitekturen er delt opp i forskjellige lag. Nederste lag tar for seg koding av tekst og ressursidentifikatorer. I laget over blir strukturen på dataen representert, mens i de to neste lagene blir kjernen til semantisk web presentert; RDF, RDF Schema og ontologispråk (for eksempel OWL). Det logiske laget gir ontologilaget et utvidelse ved å tilføre mulighet for å koble logiske egenskaper til data. De øvrige lagene er ikke relevante for denne rapporten.




Figur 2.1: Den lagdelte arkitekturen til semantisk web [McGuineness, 2001]

### URI + UNICODE

Uniform Resource Identifier (URI) er nøkkelen for å navngi ressurser. Slike ressurser er tilgjengelig via protokoller og adresser som for eksempel http://www.statoil.com. En slik adresse er kalt Uniform Resource Locator (URL), og er et eksempel på en URI. URI tilbyr en felles syntaks for å navngi ressurser uavhengig av protokollen brukt til å aksessere ressursen. Det er viktig å fremheve at URI er en identifikator til noe som kan identifiseres unikt, og ikke bare en adresse til en webside.

Unicode er en standard som lar tekst og symboler fra alle språk bli endret og håndtert på en konsistent.

### XML + XML Schema

XML er språket brukes for å strukturere informasjon. Dette gjøres ved å gi informasjonen en identifikator og identifikatorens verdi. Dette løser imidlertidig ikke problemene rundt semantisk web. Selv med identifikatorene er det ikke mulig for andre enn mennesker å forstå meningen med informasjonen. XML sier med andre ord ikke noe om semantikken til informasjonen. Funksjonen til XML blir derfor å danne en basis som andre språk med mer semantisk uttrykskraft kan bruke.

XML Schema er en utvidelse av XML som setter restriksjoner til strukturen på XML dokumenter.

### RDF + RDF Schema

Dette laget inneholder to viktige elementer i semantisk web som gir støtte til ontologilaget; RDF og dets utvidelse RDF Schema (RDFS). RDF er en datamodell som bruker XML-syntaks til å representerer informasjon. RDF strukturen danner basis for mulighetene til å lage regler som ontologilaget kan bruke i prosessen rundt resonering i data.

RDFS er en utvidelse av RDF som muliggjør overgangen fra RDF-laget til ontologilaget. RDFS tilbyr elementer brukt til å koble informasjon om data til data. RDF + RDFS blir således en form for metadatalag.

### Ontologilaget

Ontologilaget har mer utrykkskraft enn de nedre lagene fordi det tilbyr ontologispråk som for eksempel OWL. OWL er utviklet med RDFS som basis og er et språk for å beskrive data. OWL tilbyr elementer som gir informasjonsagenter

muligheten til å resonere i data, nettopp det som er hovedtanken bak semantisk web. OWL blir nærmere beskrevet i avsnitt 2.2.1.

#### Øvrige lag

Over ontologilaget finnes det tre lag:

- *Det logiske laget* utvider funksjonaliteten til ontologilaget ved at man kan koble logiske egenskaper til data.
- *Bevislaget* tilbyr språk som lar oss bevis at et utsagn er sant eller usant, og om det derfor skal gis tillit eller ikke.
- *Tillitslaget* forbedrer sikkerheten ved å tilby verktøy for å oppdage forandringer i dokumenter.

## 2.2 Ontologier

Ontologi er en gren av filosofi som omhandler læren rundt "hva er". Med dette menes de objekter, strukturer, egenskaper, hendelser og relasjoner som eksisterer i den virkelige verden. "Hva er jeg?", "Hva eksisterer?" og "Hva beskriver dette for meg?" er alle spørsmål rundt det å være. De beskriver det mest grunnleggende for ontologier; finne et emne, finne en relasjon og et objekt å snakke til.

Innen fagretningen informasjonssystemer har man et mer pragmatisk syn på ontologier. Her anser man en ontologi for være en form for delt enighet om et domene,

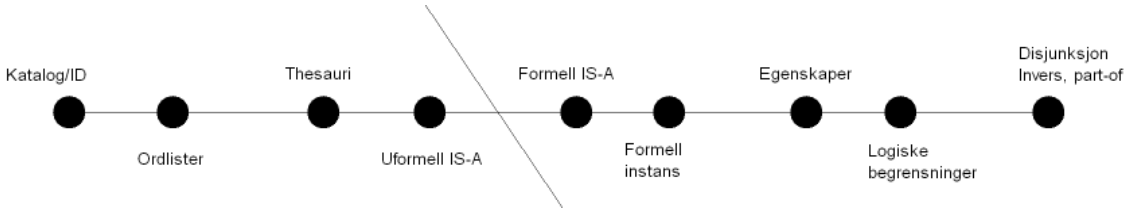
> "Ontologies (...) are often able to provide an objective specification of domain information by representing a consensual agreement on the concepts and relations characterizing the way knowledge in that domain is expressed. This specification can be the first step in building semantically-aware information systems to support diverse enterprise, government, and personal activities."
>
> *Michael Denny* [Denny, 2002]

Med tiden har fokus rundt metoder og prosesser for utvikling ontologier blitt mer og mer fremtredende. Dette gjenspeiles i ontologidefinisjonen til Guarino [Guarino, 1998], som er en tung aktør innenfor ontologier for informasjonssytemer.

> "[An ontology is] an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words (...) In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation."
>
> *Nicola Guarino* [Guarino, 1998]

Sitatet til Guarino synliggjør behovet for gjennomtenkte konstruksjons- og designprosesser i utviklingen av ontologier. Designprosessen til en ontologi møter ofte de samme problemene som all annen design, ulike interessenter har ulike meninger og ulike synspunkter. Som en konsekvens av dette finnes det et uttall ulike ontologier tilgjengelige på web i dag. Dette varierer fra uformelle taksonomier som finnes i søkemotorer, for eksempel Yahoo! Web Directory [McGuineness, 2001], til spesifikke og strengt formelle ontologier med resoneringsstøtte, som for eksempel The Open Source Business Management Ontology (BMO) [Jenz and Partner, 2006].

Med bakgrunn i de mange forskjellige typene ontologier har McGuinness i [McGuineness, 2001] valgt å sette et klart skille for hva hun mener kan, og ikke kan, betegnes som en ontologi. Oppdelingen er vist i figur 2.2. Figuren viser et klart skille mellom thesauri og formaliserte IS-A hierarkier.




Figur 2.2: Ontologispektrum [McGuineness, 2001]

Grunnene til at man konstruerer ontologier for bruk i informasjonssystemer er mange, men to grunner er spesielt fremtredende [Brasethvik, 2004]:

- Skape en felles forståelse av et domene, og dermed legge til rette for deling av informasjon om domenet.
- Muliggjøre konstruksjonen av intelligente, semantiske applikasjoner.

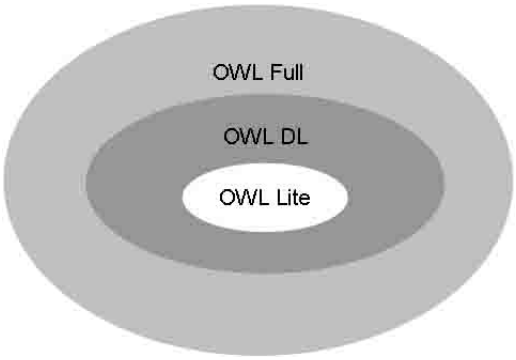
Begge disse punktene er viktig sett i sammenheng med semantisk web da semantisk web omhandler både deling og gjenfinning av informasjon, samt konstruksjon av intelligente applikasjoner og tjenester.

### 2.2.1 OWL - Web Ontology Language

Web Ontology Language (OWL) er et ontologispråk som bygger på DAML+OIL, og har rollen som ontologispråk for semantisk web. OWL ligger over RDFS i semantisk web pyramiden, og anser RDFS til å være "databasen" i semantisk web, mens OWL har fokus på representasjon av semantikken til klasser og relasjoner brukt som webressurser, og med mål om å innføre større logiske muligheter og resoneringsstøtte for semantisk web applikasjoner. OWL er utviklet av W3Cs [W3C, 2006] Web Ontology Working Group, og består av tre inkompatible sett av subspråk/arter [W3C, 2004] som vist i figur 2.3. En beskrivelse av hovedelementene i språket er gitt i tabell 2.1.

| | |
|---|---|
| **Konsepter** | OWL bygger på RDFS og dens basiskomponenter er bygget opp av RDF komponenter. **owl:Class** er således en spesialisering av **rdfs:Class**. |
| **Attributter** | Basert på RDF. OWL har to spesialiseringer av **rdf:Properties**; **owl:ObjectProperty** og **owl:DatatypeProperty**. |
| **Relasjoner** | Relasjoner er basert på RDF elementer som **subClassOf** og **subPropertyOf**. Arbitrære relasjoner er definert som egenskaper med **rdfs:Domain** og **rdf:Range** som begrensninger, og som kardinalitetsrestriksjoner. Egenskapskarakteristika gir man ved å bruke elementer som **inverseOf**, **Transitiv**, **Symmetric** og **Functional**. |
| **Instanser** | Instanser er definert gjennom "individual"-elementet. |
| **Aksiomer** | Aksiomer er brukt til å gi logisk informasjon til klasser og attributter. |
| **Verktøystøtte** | Protégé [Stanford, 2006] er blitt de-facto standard, men editorer som SWOOP [Mindswap, 2006] og Topbraid Composer [TopQuadrant, 2006] blir stadig mer populære. |

Tabell 2.1: Beskrivelse av OWL




Figur 2.3: De tre artene av OWL

### 2.2.2 OWL Full

Det komplette språket kalles OWL Full, og bruker alle OWL språkprimitivene. Det tillater også vilkårlige kombinasjoner av disse primitivene med RDF og RDF Schema. Dette inkluderer muligheten til å forandre meningen av forhåndsdefinerte primitiver ved å slå sammen språkprimitiver. [W3C, 2004]

Fordelen med OWL Full er at den er fullt kompatibel med RDF, både syntaktisk og semantisk; det vil si at ethvert gyldig RDF dokument også er et gyldig OWL Full dokument, og enhver gyldig RDF/RDF Schema konklusjon er også en gyldig OWL Full konklusjon. [W3C, 2004]

Ulempen med OWL Full er at språket har blitt så komplekst at det er vanskelig å bygge resoneringsverktøy for det. Det er også forholdsvis tregt å resonnere i, og gir ikke alltid brukeren et komplett svar. [W3C, 2004]

### 2.2.3 OWL DL (Description Logic)

OWL DL er designet for å gi støtte til beskrivende logikk, og å tilby et språksubsett som har fordelaktige utregningsegenskaper for resonnementssystemer. For å utnytte den formelle understøtten for beskrivende logikk, er følgende restriksjoner fulgt i en OWL DL ontologi: [W3C, 2004]

**Vokabulær oppdeling:** Enhver ressurs kan bare være enten en klasse, en datatype, en datatype-egenskap, en objektegenskap, et individ, en dataverdi eller en del av det innebygde vokabularet, og ikke mer enn en av disse.

**Eksplisitte spesifiseringer:** Alle ressurser må være partisjonert, og partisjoneringen må være spesifisert eksplisitt.

**Separerte egenskaper:** Inverse egenskaper, invers funksjonalitet og symmetrisk karakteristikk kan aldri spesifiseres for datatype egenskaper.

**Ingen begrensninger på transitiv kardinalitet:** Det kan ikke legges noen begrensninger på kardinaliteten til transitive egenskaper.

**Begrensede anonyme klasser:** Anonyme klasser er kun tillatt i domenet: **owl:equivalentClass** og **owl:disjointWith**.

### 2.2.4 OWL Lite

OWL Lite er laget for å kunne implementeres enkelt og tilbyr brukerne et funksjonelt subsett av OWL. Den er spesielt rettet mot verktøybyggere som ønsker støtte for OWL, men som vil starte med et relativt lite og enkelt sett av språkegenskapene. En OWL Lite ontologi må være en OWL DL ontologi, og i tillegg tilfredsstille følgende krav: [W3C, 2004]

- Konstruktorene **owl:oneOf, owl:disjointWith, owl:unionOf, owl:complementOf** og **owl:hasValue** er ikke tillatt.
- Kardinalitetsutsagn (både minimal, maksimal og eksakt kardinalitet) kan bare lages ut fra verdiene 0 og 1.
- **owl:equivalentClass** utsagn kan ikke lenger lages mellom anonyme klasser, bare mellom klasseidentifikatorer.

### 2.2.5 Sammenlikning av OWL-artene

Følgende regler gjelder for kompabilitet mellom de tre subspråkene [Schneider and Dean, 2004]:

- Enhver lovlig OWL Lite ontologi er en lovlig OWL DL ontologi.
- Enhver lovlig OWL DL ontologi er en lovlig OWL Full ontologi.
- Enhver gyldig OWL Lite konklusjon er en gyldig OWL DL konklusjon.
- Enhver gyldig OWL DL konklusjon er en gyldig OWL Full konklusjon

En kort sammenlikning av OWL-artene er sammenfattet i tabell 2.2.

| **OWL Full** | **OWL DL** | **OWL Lite** |
|---|---|---|
| - Alle språkprimitiver er tillatt.<br>- RDF Schema definisjoner kan blandes med OWL definisjoner. | - Kan ikke bruke **owl:cardinality** med **TransitiveProperty**.<br>- En DL-ontologi kan ikke bruke hele OWL Full ontologien.<br>- Kan ikke bruke en klasse som medlem i en annen klasse, dvs kan ikke ha metaklasser.<br>- **FunctionalProperty** og **InverseFunctionalProperty** kan ikke brukes med datatyper (kun med **ObjectProperty**). | - Samme restriksjoner som i DL, i tillegg til:<br>- Kan ikke bruke **owl:minCardinality** eller **owl:maxCardinality**.<br>- Eneste tillatte verdier for **owl:cardinality** er 0 og 1.<br>- Kan ikke bruke **owl:hasValue**.<br>- Kan ikke bruke **owl:disjointWith**.<br>- Kan ikke bruke **owl:oneOf**.<br>- Kan ikke bruke **owl:complementOf**.<br>- Kan ikke bruke **owl:unionOf**. |

Tabell 2.2: Sammenligning av OWL-artene

## 2.3 Informasjonsgjenfinning

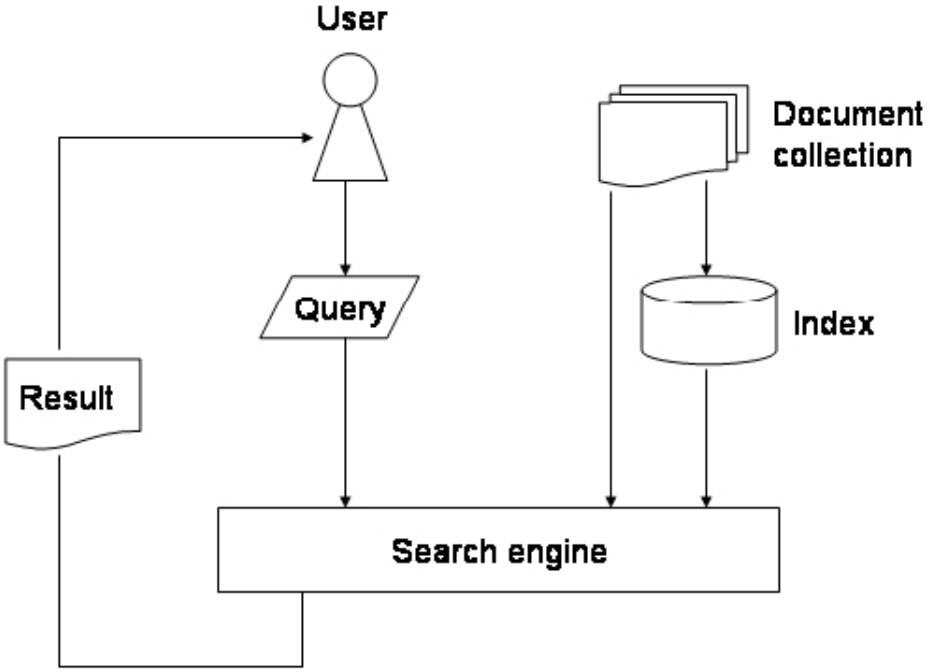
Klassiske teknikker for informasjonsgjenfinning (IR) er kjernen i de fleste av dagens søke- og gjenfinningssystemer på Internett og i ulike intranett og dokumentstyringssystemer. IR ble definert så tidlig som i 1950-årene som:

> "Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into a actual list of documents in storage containing information useful to him."
>
> [Mooers, 1950], resitert i [Savino and Sebastiani, 1998]

En nyere definisjon er gitt av Baeza-Yates og Ribeiro-Neto:

> "Information retrieval deals with the representation, storage, organisation of, and access to information items. (...) [IR] is more concerned with retrieving information about a subject then with retrieving data which satisfies a given query. (...) the retrieved objects might be inaccurate and small errors are likely to go unnoticed."
>
> [Baeza-Yates and Ribeiro-Neto, 1999]

I sin klassiske, og fortsatt mest vanlige form, omhandler IR tekstbasert gjennfinning på dokumentnivå. Figur 2.4 illustrerer den generelle arkitekturen til et IR-system på dette nivået, og man vil i de neste avsnittene se på metoder og teknikker brukt i et slikt system.



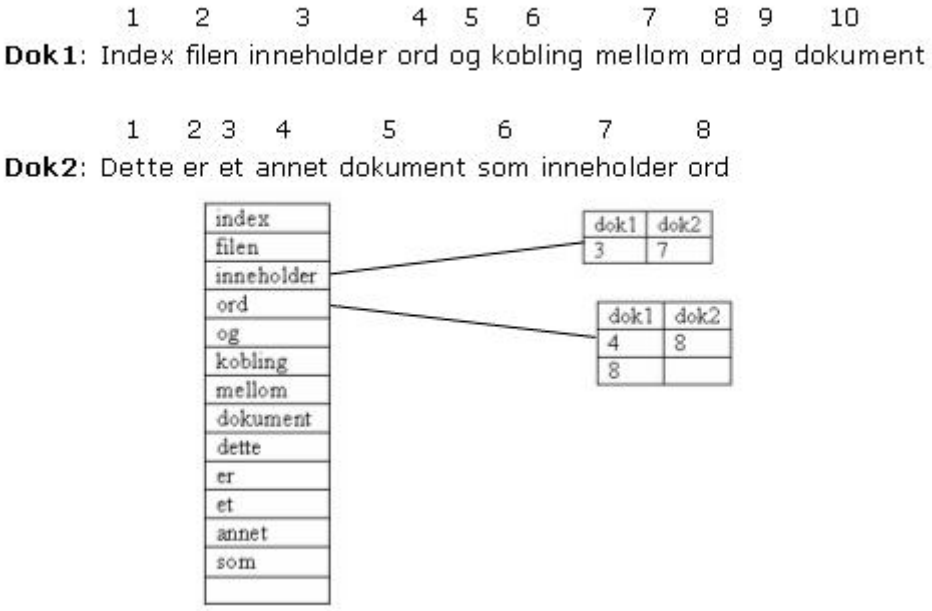

Figur 2.4: Eksempel på tradisjonelt IR-system

### 2.3.1 Indeksering og invertert fil

Dersom man velger å søke etter tekst i dokumenter ved å søke sekvensielt gjennom hvert dokument, vil kostnaden ved å søke øke dramatisk dersom dokumentsamlingen øker. Med bakgrunn i dette er det vanlig å indeksere dokumentsamlingen for å øke ytelsen på søket, noe som oppnås ved å samle nødvendig informasjon fra dokumentene i en indeks som søket utføres mot. [Baeza-Yates and Ribeiro-Neto, 1999]

En vanlig indekseringsteknikk er å opprette en såkalt invertert fil som er en ordorientert mekanisme for å indeksere en dokumentsamling. Strukturen i inverterte filer er to- eller tredelt; en ordliste, en liste over hvilke dokumenter ordet forekommer og eventuellt en liste over hvor i dokumentet ordet forekommer. Konseptet er illustrert i figur 2.5. Hver rad i ordlisten representerer et begrep (ord eller frase) som forekommer i dokumentene, mens hver kolonne i forekomstlisten representerer et dokument hvor ordet forekommer. Radene i forekomstlisten viser hvor

i dokumentet ordet forekommer. En invertert fil gjør det dermed enkelt å finne treff i en dokumentsamling utfra en spørring. Man slår opp i ordlisten og finner ordet som korresponderer til spørringen, og returnerer dokumentene som er listet opp. [Baeza-Yates and Ribeiro-Neto, 1999]




Figur 2.5: Eksempel på invertert fil

### 2.3.2 Preprosessering

Ettersom dokumenter vil inneholde ord, og former av ord, som ikke utgjør noen forskjell for dokumentets semantikk, er det derfor vanlig å kjøre enkelte teknikker på dokumentene før de ulike ordene legges inn i den inverterte filen. I dette avsnittet vil en presentere to slike teknikker; stemming og fjerning av stoppord. [Baeza-Yates and Ribeiro-Neto, 1999]

#### Stemming og lemmatisering

I det fleste tilfeller har morfologiske[1] varianter av ord lik semantisk tolkning, og kan derfor sees på som ekvivalente i IR-sammenheng. Med bakgrunn i dette er

[1]Morfologi: Hvordan ord er oppbygd og blir dannet av mindre enheter, det vil si ordbøyning og orddanning.

det blitt utviklet flere såkalte stemming-algoritmer for å redusere et ord til sin ordstamme, også kalt stem. Dette gjøres normalt med å fjerne endingen til ordet utifra regler gitt i algoritmen. På denne måten vil et IR-system også kunne finne andre bøyningsformer av ord ved et søk ettersom det er ordstammen som blir lagret i indeksen, samt at spørringen også reduseres til ordstammen før søket utføres. [Baeza-Yates and Ribeiro-Neto, 1999]

```
Eksempel: projects -> project
```

Som nevnt over utføres stemming av regler gitt i stemming-algoritmen og følger dermed en fast definert framgangsmåte. På grunn av dette vil det oppstå flere tilfeller hvor en del av et ord blir fjernet slik at ordstammen ikke blir riktig i forhold ordets oppbygning i språket. Effekten av dette er likevel begrenset ettersom de alle fleste ord vil få en unik stamme.

```
Eksempel: computation -> comput
```

Et alternativ til stemming er bruk av en metode kalt lemmatisering. Forskjellen mellom de to metodene er at lemmatisering bruker ordlister med ulike bøyningsformer for å redusere ordet tilbake til grunnformen av ordet.

```
Eksempel: computation -> compute
```

Et problem med lemmatisering er at ordet kun blir gjort om til grunnformen dersom ordet finnes i ordlisten, samt at arbeidet med å utarbeide og vedlikeholde slike lister vil kunne bli komplekst og kostbart.

### Fjerning av stoppord

Ord som opptrer i over 80% av dokumentene har ingen differensierende effekt for dokumentene i samlingen, og vil være ubrukelig i en søkesituasjon. Slike ord kaller man stoppord, og ordene hører normalt til visse ordklasser som adverb, preposisjoner, konjunksjoner og artikler.

Det finnes både styrker og svakheter ved en eventuell fjerning av stoppord. Siden stoppord ikke har noen differensiserende verdi, har fjerning av disse en stor fordel ved at størrelsen på indeksstrukturen reduseres betraktelig, og det er vanlig med 40% minking av indeksen. Fjerning av stoppord har også ulemper siden dette kan redusere recall[2]. For eksempel kan søket etter frasen "to be or not to be" resultere i at så og si hele strengen fjernes, og det vil gjøre det nesten umulig å få returnert relevante treff ved søk på denne typen fraser. Dette er grunnen til at flere søkemaskiner plasserer alle ord i dokumentsamlingen inn i den inverterte filen og ikke fjerner stoppord. [Baeza-Yates and Ribeiro-Neto, 1999]

[2]Treffrate: Andel dokumenter i resultatet som er relevante i forhold til spørringen

### 2.3.3 IR-modeller

En IR-modell er ifølge [Baeza-Yates and Ribeiro-Neto, 1999] definert som et kvadruppel $[\mathbf{D}, \mathbf{Q}, \mathcal{F}, R(q_i, d_j)]$ hvor

1. $\mathbf{D}$ er en mengde av logiske representasjoner av dokumentene i samlingen.
2. $\mathbf{Q}$ er en mengde logiske representasjoner av brukerens informasjonsbehov.
3. $\mathcal{F}$ er et rammeverk for å modellere dokumentrepresentasjoner, spørringer og deres forbindelser.
4. $R(q_i, d_j)$ er en rangeringsfunksjon som assosierer et reelt tall med en spørring $q_i \in \mathbf{Q}$ og en dokumentrepresentasjon $d_j \in \mathbf{D}$. En slik rangering definerer en orden i dokumentene i henhold til spørringen $q_i$.

Innenfor tekstgjennfinning er de logiske representasjonene av både spørringer og dokumenter ekstrahert eller opprettet med bakgrunn i ordene som finnes i teksen, det vil med andre ord si at de logiske representasjonene er opprettet med bakgrunn i indekstermer. [Brasethvik, 2004]

Det finnes tre klassiske modeller for IR; boolsk modell, vektormodell og sannsynlighetsmodellen. Vektormodellen er den mest brukte av de tre, og vil bli presentert nærmere i denne rapporten. De to resterende presenteres detaljert i [Baeza-Yates and Ribeiro-Neto, 1999].

#### VSM - Vector Space Modell

I vektormodellen blir dokumenter og spørringer representert som vektorer definert utifra indekstermer. Modellen tilbyr et rammeverk der det er mulig å finne delvis match i et søk utifra en gitt spørring. Dette gjøres ved å bruke ikke-binær vekting av indekstermene i både spørringer og dokumenter. På grunnlag av dette er det mulig å regne ut graden av likhet mellom dokumentene i samlingen og spørringen. Ved å sortere dokumentene som returneres i synkende rekkefølge etter graden av likhet vil modellen returnere dokument som også bare matcher spørringen delvis.

Likheten mellom et dokument og en spørring kan måles ved se på størrelsen til vinkelen mellom vektorene deres. Dette gjøres normalt ved å beregne cosinuslikheten som vist i ligning 2.1.

$$sim(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}, \; hvor \tag{2.1}$$

| | | |
|---|---|---|
| $sim(d_j, q)$ | = | korrelsjonen mellom $d_j$ og $q$ |
| $\vec{d_j}$ | = | vektoren til dokument j |
| $\vec{q}$ | = | vektoren til spørringen |
| $t$ | = | totalt antall indekstermer i dokumentsamlingen |
| $w_{i,j}$ | = | vekten til termen i dokument j |
| $w_{i,q}$ | = | vekten til termen i spørringen |

Som nevnt over bruker vektormodellen ikke-binær vekting av spørring og dokumenter. Et eksempel på en vektingsalgoritme er TF-IDF (Term Frequency - Inverse Document Frequency). TF sier noe om ordet sin relative frekvens i dokumentet. IDF sier noe om viktigheten av ordet i hele dokumentsamlingen. Målet med vektingen er at ord som forekommer i mange dokumenter får en lavere vekting, da dette ordet i liten grad vil kunne brukes til å skille dokumenter fra hverandre. [Baeza-Yates and Ribeiro-Neto, 1999] Algoritmen beskrives slik:

1. $N$ = antall dokumenter i systemet.
2. $n_i$ = antall dokumenter hvor ordet $k_i$ forekommer totalt.
3. $freq_{i,j}$ = antall ganger $k_i$ forekommer i dokumentet $d_j$.
4. $max_l freq_{l,j}$ = antallet ordet som forekommer mest i dokumentet $d_j$ forekommer.
5. Den normaliserte frekvensen $f_{i,j}$ for ordet $k_i$ i dokumentet $d_j$ er da $TF_{i,j} = \frac{freq_{i,j}}{max_l freq_{l,j}}$.
6. Den inverse dokumentfrekvensen $IDF_i = log\frac{N}{n_i}$.
7. Returnerer vekten til ordet $k_i$ i dokumentet $d_j$: $w_{i,j} = TF_{i,j} * IDF_i$.

Siden spørringer av natur er mye kortere enn dokumenter bruker en ofte varianter av TF-IDF for å gi vekt til spørringer. Salton og Buckley har utarbeidet følgende forslag for vekting av spørringer [Baeza-Yates and Ribeiro-Neto, 1999]:

$$w_{i,q} = (0.5 + \frac{0.5 freq_{i,q}}{max_l freq_{l,q}} * log\frac{N}{n_i}) \tag{2.2}$$

Bokstavene har her samme betydning som for TF-IDF over. $w_{i,q}$ er vekten av ordet $k_i$ i spørringen $q$.

### 2.3.4 Ontologi i søkeapplikasjoner

Ontologidrevet IR inkorporerer en rekke elementer som ikke er vanlige i tradisjonelle IR-systmer. De vanligste søkemotorene på Web bruker i dag enkle linguistiske teknikker i indekseringsfasen, men prøver ikke å "forstå" innholdet til

dokumentene og er primært opptatt av å gjøre dokumentene tilgjengelig for gjennfinning. [Gulla et al., 2002]

Ifølge [Gulla et al., 2006] opererer et ontologibasert IR-system på tre forskjellige ambisjonsnivå som vist i tabell 2.3. På det laveste nivået bruker en begrepshierarkier i ontologien til å gjennfinne og presentere dokumenter for brukeren. Her blir ontologien enten brukt til å reformulere spørringen eller til å bygge semantiske indekser. På det neste nivået blir ontologien gjort synlig for brukeren. Ideen er at brukeren selv kan bla seg gjennom de ulike nivåene i ontologien for å se de ulike begrepene og relasjonene som finnes. På denne måten får brukeren oversikt over domenet og kan derfor lettere finne relatert informasjon. På det høyeste nivået bruker en resonering for å kunne gi brukeren svar med bakgrunn i dokumenter og logikk inkorporert i ontologien.

| **Funksjon** | **Fokus** | **Ontologielement** |
|---|---|---|
| Gjennfinne dokument | Begreper | Begreper, hierarkier |
| Bla i kunnskap | Ontolgiske strukturer | + Attributter, relasjoner |
| Utarbeide svar | Resonering | + Logikk, restriksjoner |

Tabell 2.3: Tre anvendelsesområde for ontologisk informasjon i IR

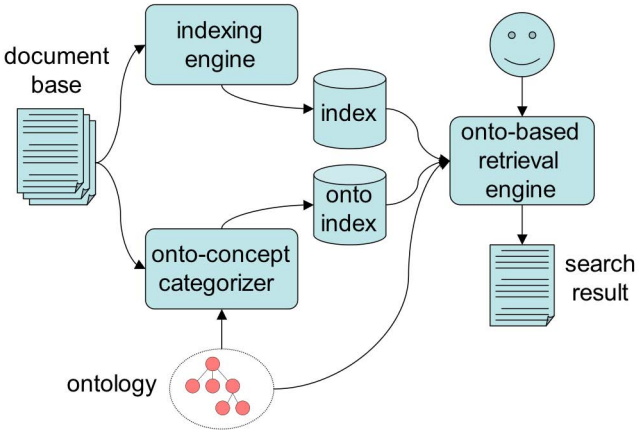
Fokuset rundt bruk av ontologi i dette arbeidet ligger på det laveste ambisjonsnivået, det vil si ontologistøttet søk etter dokumenter.

### Eksempel på ontologistøttet IR-system

For å få et innblikk i hvordan et state-of-the-art ontologibasert IR-system kan realiseres vil en i dette avsnittet kort presentere et ontologibasert system som er under utvikling ved IDI. [Thommasen et al., 2005] I tilnærmingen prøver man å berike dokumentene med ontologiske beskrivelser for å forsøke å plassere dokumentene i riktig domene, med det mål å forbedre precision (presisjon) og recall (treffrate) i søkesammenheng. Den overordnede arkitekturen til systemet er vist i figur 2.6.

Hovedkomponentene i systemet beskrives slik:

**Onto-concept categorizer:** Her gjennomføres kategorisering av begreper som er relevant for et spesielt ontologibegrep. Ontologibegrepene er klasser som er definert i ontologien. Ontologibegrepene blir utvidet med et sett relevante termer fra dokumentene ved hjelp av ulike tekstkategoriseringsteknikker. Videre blir ontologibegrepet, sammen med de relaterte begrepene (synonymer, bøyningsformer o.l) lagt inn i *onto index* som begrepsvektorer, og brukt av *onto-based retrieval engine* i gjennfinningsprosessen.




Figur 2.6: Eksempel på ontologistøttet IR-system [Thommasen et al., 2005]

**Indexing engine:** Hovedoppgaven til denne komponenten er indeksering av dokumentsamlingen. Indekseringsmotoren er Lucene-basert, som er en fritt tilgjengelig søkemotor fra Apache. [Apache, 2006]

**Onto-based retrieval engine:** Her håndteres spørringer fra brukeren. Systemet tar sikte på å støtte både vanlig fritekstsøk, og samtidig ha en funksjon hvor brukeren kan bla i ontologihierarkiet via et brukergrensesnitt. I tillegg vil brukeren få muligheten til å vekte termene som er koblet til ontologibegrepet utifra personlig mening om relevans.

Søk etter begreper som finnes i ontologien håndteres annerledes enn andre begreper, og søket blir her delt opp i to faser. I den første fasen returneres dokumenter som står til ontologibegrepet. Disse taes så med til fase to hvor bare de dokumentene som inneholder ontologibegrepet og de relaterte termene blir returnert.

# Kapittel 3

# Manuell ontologikonstruksjon

I dette kapitlet presenteres det arbeidet som er gjort for å komme frem til den manuelt bygde ontologien. Først presenteres noen teknikker og metoder fra litteraturen som er brukt under arbeidet, før man presenterer det faktiske arbeidet som er gjort.

## 3.1 Manuell begrepsekstraksjon

I dette avsnittet vil en beskrive en tilnærming for ren manuell uthenting av begreper fra en dokumentsamling. Selv om dette lett kan sees på som en enkel og "rett på sak"-oppgave vil det være en klar fordel å ha en klar plan for arbeidet. Dette for å sikre god dekning av domenet, og for å unngå problemer som tvetydige begreper. Fremgangsmåten som beskrives i avsnittet er presentert i [Uschold and Gruninger, 1996], og ble brukt i utviklingen av Enterprise-ontologien beskrevet i [Uschold and King, 1995]. Fremgangsmåten har fire steg; definere ontologiskop, lage definisjoner, gjennomgang av definisjoner og konstruksjon av meta-ontologi.

### 3.1.1 Definer ontologiskop

**Brainstorming** - Denne teknikken kan brukes til å identifisere alle potensielle relevante begreper og fraser. Etter endt brainstorming sitter man igjen med en mengde begreper som alene representerer et konsept innenfor domenet. Eventuelle tvetydigheter og uenigheter rundt meningen med begrepene blir ikke avklart i denne sekvensen. For å oppnå god dekning av relevante begreper om domenet er det viktig å ha med domeneeksperter dersom personene involvert i prosessen ikke innehar denne kunnskapen selv.

**Gruppering** - Her strukturerer man begrepene "løst" ved å plassere dem i naturlige sub-grupper, det vil si grupper av ord som har relasjoner til hverandre. For hvert enkelt begrep bør følgende gjøres:

- Kategoriser begrepene: Bør begrepet være med, bør det ikke være med, eller ligger det i grenseland. Dokumenter avgjørelsene.
- Grupper like begreper og potensielle synonymer sammen for videre vurdering.
- Identifiser semantiske kryssreferanser. Det vil si begreper som kan bli referert til av begreper i andre sub-grupper.

### 3.1.2 Definering av begreper

I prosessen med å definere ontologiskopet har man kommet frem til de viktigste begrepene i domenet. Ifølge [Uschold and Gruninger, 1996] er den viktiste delen av en ontologiutviklingsprosess det å komme til enighet om begrepene i ontologien, det vil si å skape en felles forståelse om hva begrepene omfatter, og hvilke relasjoner de har til andre begreper. Ved å gjøre dette for hvert begrep vil en til slutt skape en felles forståelse for domenet ontologien dekker. Dette avsnittet tar for seg oppgaver og valg som må taes på veien mot felles forståelse.

**Overlapping** - Ved å ta utgangspunkt i gruppene fra 3.1.1 er det viktig å starte arbeidet med begreper i grupper som har stor grad av semantisk tilknytning. Disse begrepene er viktige å få klarhet i først da feil vil kunne føre til merarbeid senere.

**Strategivalg** - Ifølge [Uschold and Gruninger, 1996] bør starte å definere begreper utifra en middle-out strategi. Dette innebærer å starte med de grunnleggende begrepene for hver gruppe før en definerer mer generelle eller spesialiserte begrepet. For eksempel vil "hund" være et grunnleggende begrep, "pattedyr" vil være en generalisering, mens "Labrador" vil være en spesialisering.

Grunnen til at en bør velge en middle-out strategi i motsetning til top-down (starte med de mest generelle begrepene, for så å ta de mer spesialiserte begrepene) eller bottom-up (starte med de mest spesifiserte begrepene, for så å ta de mer generelle begrepene) er at denne vil skape en "balanse" i detaljeringsnivået til ontologien. Ved bruk av bottom-up eller top-down er det fare for at detaljeringsgraden blir henholdvis for høy eller for lav. [Uschold and Gruninger, 1996]

**Håndtering av tvetydige begreper** - Tvetydige begreper er en et problem i ontologier ettersom det er en klar trussel mot målet om å oppnå felles forståelse rundt et domene. Sett oppimot en søkesituasjon vil det kunne føre til at brukere

får returnert svar på en spørring som ikke er det brukeren var ute etter. En stegvis fremgangsmåte for å bli kvitt tvetydige termer er:

1. Ta ut det tvetydige begrepet.
2. Bruk ordbøker eller andre oppslagsverk for å finne de ulike konseptene det tvetydige begrepet kan bety.
3. Bli enig om hvilket konsept som passer inn i ontologien.
4. Finn et passende og utvetydig begrep som erstatter for det opprinnelige tvetyde begrepet.

**Generelle retningslinjer** - [Uschold and Gruninger, 1996] foreslår et sett retningslinjer som bør gjennomsyre hele begrepsprosessen:

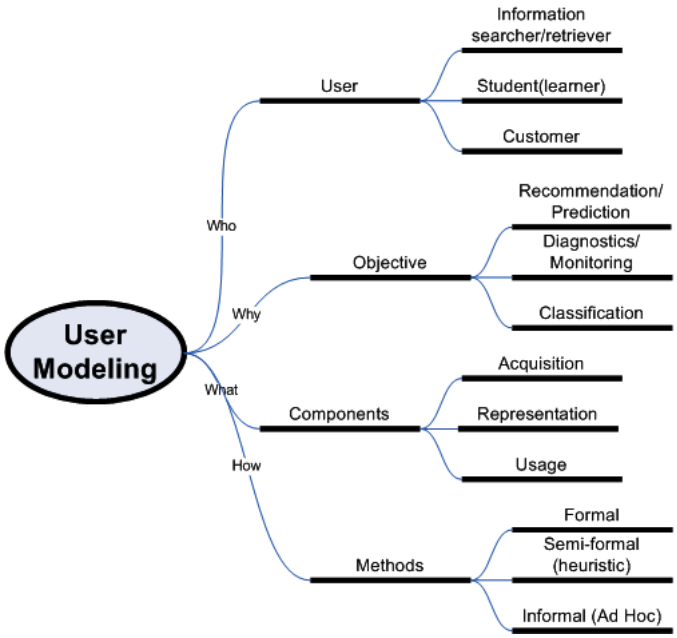
- Lag en presis beskrivelse av hvert begrep ved hjelp av naturlig språk, unngå "runde" beskrivelser.
- Sikre konsistens i forhold til begrepene som allerede finnes i ontologien.
- Vis relasjoner til andre brukte begreper, synonymer o.l.
- Bruk eksempler der det passer.

### 3.1.3 Gjennomgang av definisjoner

I dette steget skal det gjøres en kritisk gjennomgang av begrepene med tilhørende definisjon som man sitter igjen med etter foregående steg, og endre der det er nødvendig. Endringer skal dokumenteres.

### 3.1.4 Konstruksjon av meta-ontologi

Det siste går ut på å konstruere en meta-ontologi ved å bruke naturlig språk definisjonene som en kravspesifikasjon. En meta-ontologi er en abstrakt ontologi som viser hvilke deler ontologien skal bestå av. Meta-ontologien danner grunnlaget for videre implementasjon av ontologien. Et eksempel på en user-modelling meta-ontologi (UMMO) er vist i figur 3.1.




Figur 3.1: Eksempel på UMMO

## 3.2 Praktisk gjennomføring

Dette avsnittet presenterer kort hva som faktisk ble gjort i prosessen med å bygge den manuelle ontologien. Først ser en kort på hvordan dokumentet (PMBOK [(PMI), 2000]) som ligger til grunn for ontologiene er oppbygd, før en ser på måten PMBOK ble gjennomgått på for å komme frem til et førsteutkast av ontologien som kunne presenteres for en domeneekspert. Videre ser man på gjennomføringen av møtet med domeneeksperten, før en til slutt ser på hvilken lærdom man sitter igjen med i etterkant av prosessen.

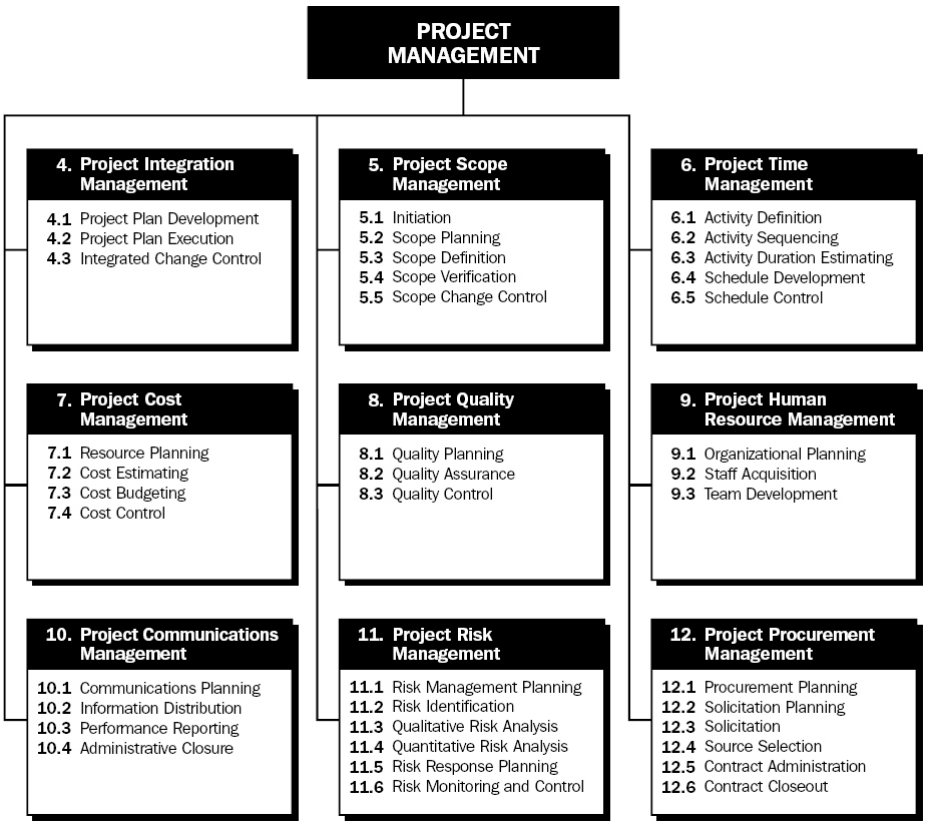
### 3.2.1 PMBOKs innhold og oppbygning

For å forstå bedre hvordan man har gått fram i utvelgelsen av begreper og fraser til ontologien gis det i dette avsnittet en kort innføring i PMBOKs innhold og struktur.

PMBOK er delt inn i fire hoveddeler hvorav de to første er interessante for denne oppgaven. De to resterende består av appendix og en ordliste. Den første delen består av tre kapitler som gir en grundig introduksjon til prosjektledelse og prosjektprosesser generelt, samt en innføring i hvordan de ulike kunnskapsområdene i del 2 henger sammen. Del 2 utgjør hovedtyngden av dokumentet og består av kunnskapsområder:

- Project Integration Management
- Project Scope Management
- Project Time Management
- Project Cost Management
- Project Quality Managemet
- Project Human Resource Management
- Project Communications Management
- Project Risk Management
- Project Procurement Management

Hvert kunnskapsområde har sitt eget kapittel, som igjen er delt inn i en rekke delkapitler. En fullstendig oversikt over innholdet er vist i figur 3.2.




Figur 3.2: Struktur PMBOK [(PMI), 2000]

### 3.2.2 Gjennomgang av dokumentet

Prosessen startet med å lese gjennom hele dokumentet for å skaffe seg en grov oversikt over domenet. PMBOK er rikt illustrert med figurer og modeller som hjelper leseren til å få oversikt over de viktigste elementene, og studier av disse hadde sterk fokus i denne fasen. Det ble hverken gjort bemerkninger i teksten, eller tatt notater iløpet av den første gjennomlesningen.

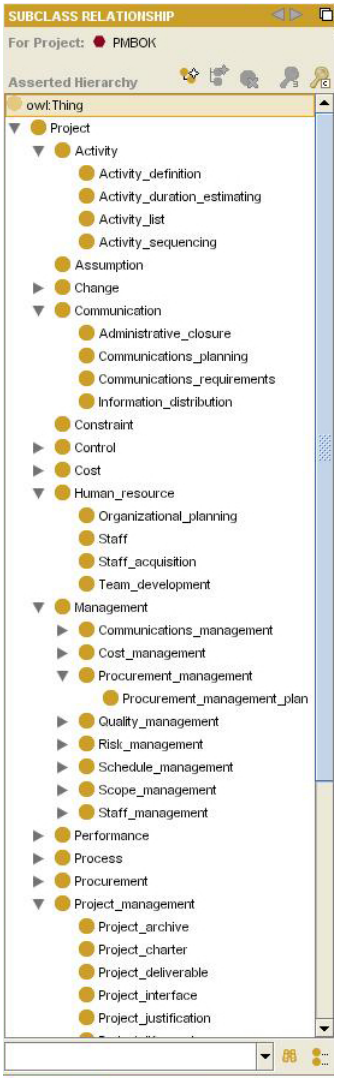
I den neste fasen startet arbeidet med å samle inn begreper. For å gjøre dette så strukturert som mulig tok en utgangspunkt i strukturen til PMBOK og analyserte ett og ett kapittel om gangen. Ettersom alle delkapitlene i hvert kapittel har svært meningsbærende navn ble ofte disse markert ut som viktige da man antok at disse representete vesentlig elementer i domenet. Ved å gjøre dette fulgte man den middle-out strategien som anbefales av [Uschold and Gruninger, 1996] for bedre å kunne kontrollere detaljnivået i ontologien.

Etter innsamlingen grovsorterte man begrepene i hierarkier på papir, og dobbeltsjekket deretter med dokumentet om alle begreper som burde være med faktisk var det. Deretter brukte man Protégé [Stanford, 2006], en ontologieditor fra Stanford Medical Informatics, til å overføre begrepene fra papir til det formelle ontologispråket OWL. Et utsnitt av begreps- og attributthierarkiet i den ferdige ontologien er vist i figurene 3.3 og 3.4. Når det gjelder attributter ble det allerede på et tidlig tidspunkt klart at dette OWL-elementet ville bli nedprioritert i dette arbeidet, og at man heller skulle ha sterkt fokus på gode begreper og begrepshierarki. Dette fordi attributtene ikke er tiltenkt noen rolle i IR-evalueringen som presenteres i kapittel 5. Det ble likevel lagt til 34 attributter for øvingens skyld.

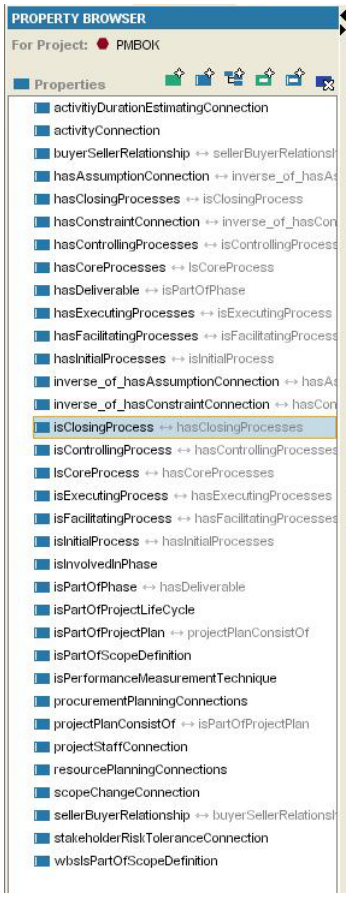
### 3.2.3 Møte med domeneekspert

For å sikre skikkelig kvalitet i ontologien valgte man å gjennomføre et møte med en domeneeksperter (prosjektleder) fra Statoil ASA. I forkant av møte brukte man en av funksjonene i Protégé til å visualisere hierarkiet i ontologien og så eksportere det til et bilde. Dette bilde ble så sendt til domeneeksperten sammen med en tekstlig forklaring av hva som var målet med hele diplomoppgaven, samt en spesifikk beskrivelse av hva man ønsket å gå gjennom på møtet. Grunnen til at man sendte bildet er at enkle grafiske fremstillinger gjør jobben med å sette seg inn i innholdet i ontologien mye enklere og mer effektivt. [de Almeida Falbo et al., 1998] Figur 3.5 viser visualiseringsverktøyet i Protégé som ble brukt til å lage bildet. Bildet av ontologien ligger vedlagt i tillegg F.

Selve møtet ble gjennomført i Statoils lokaler på Rotvoll i Trondheim hvor man disponerte et møterom med videoprosjektør. Under møtet gikk man gjennom hvert enkelt begrep og avgjorde om det var relevant eller ikke, og om det var



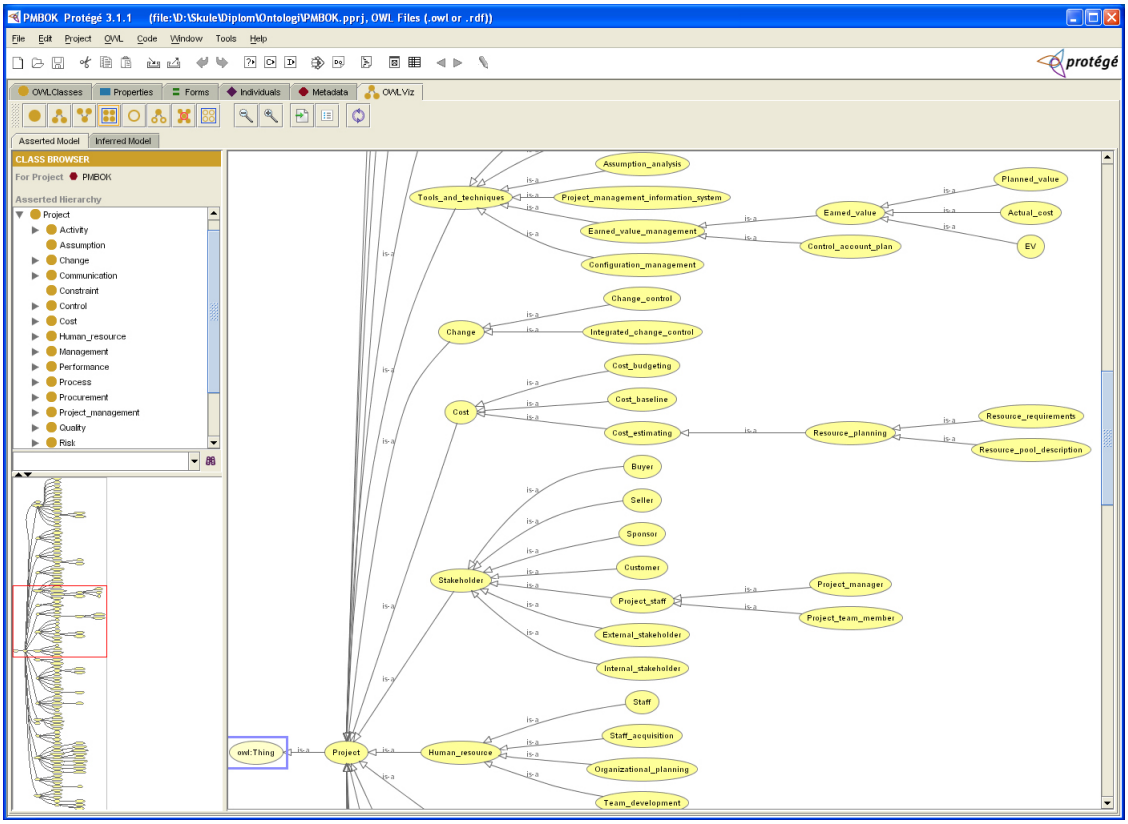

Figur 3.3: Utsnitt av begrepshierarki




Figur 3.4: Utsnitt av attributt-hierarki

riktig plassert i hierarkiet. For å utnytte den begrensede tiden som var satt av til møtet ble det ikke utført noen endringer på stedet, men ble istede notert ned for senere endring.

I etterkant av møtet ble de noterte endringene ordnet i Protégé og et nytt hierarkibilde ble laget. Dette ble så sendt tilbake til domeneeksperten for kontroll, og ontologien ble deretter ferdigstilt etter å ha lagt til skriftlige definisjoner for hvert enkelt begrep/frase.




Figur 3.5: Protégé OWLViz

### 3.2.4 Lærdom

I etterkant av arbeidet med den manuelle ontologien sitter man igjen med endel viktige erfaringer:

**Tidsbruk:** Å konstruere en ontologi manuelt er svært arbeidskrevende og tar lang tid. Selv med en så liten dokumentsamling som PMBOK (160 sider), og dermed også en liten ontologi, brukte man betydelig med tid på å sette seg inn i, og få kontroll over innholdet. Arbeidet med å trekke ut de viktige begrepene tar også mye tid og det er derfor viktig å sette seg nøye inn i stoffet på forhånd slik at man utnytter tiden effektivt.

**Gjennomføring av møte med domeneekspert:** Erfaringen fra dette arbeidet er at møtene med domeneekspertene ikke bør ha en fastsatt tidsramme på forhånd. I dette tilfellet hadde en satt av 2 timer til møtet, mens man endte opp med å bruke 3 timer og ble da avbrutt fordi domeneeksperten hadde andre avtaler han måtte rekke. Man klarte likevel å få gjennomgått alle begrepene selv om det ble noe hastearbeid mot slutten av møte, noe som kan ha påvirket evalueringen av enkelte begrep.

En annen viktig erfaring man har gjort seg at en bør frigjøre seg mest mulig fra selve dokumentsamlingen under møtet, og heller bruke den kunnskapen domeneeksperten faktisk sitter inne med. Under møtet opplevde man at domeneeksperten ble sittende å dobbeltsjekke alt han sa, selv om det var veldig tydelig at han faktisk hadde kontroll på det han snakket om. Dette gjorde at fremdriften i møtet gikk svært sakte i begynnelsen, og man bestemte seg etter hvert for å legge bort dokumentet, og kun bruke det til dobbeltsjekking i tvilstilfeller. Konklusjonen blir at en heller bør utnytte den tiden man har med domeneeksperten effektivt for så eventuelt å dobbelsjekke ting man er usikker på i etterkant.

**Økende domenekunnskap:** Personer som utvikler en ontologi manuelt vil gjennom utviklingsprosessen selv opparbeider seg domenekunnskap, noe som kan være svært nyttig ved senere videreutvikling og forbedring av ontologien.

# Kapittel 4

# Semi-automatisk ontologikonstruksjon

I dette kapitlet presenteres det arbeidet som er gjort for å komme frem til den semi-automatisk konstruerte ontologien. Først presenterer man noen av basismetodene for å trekke ut begreper fra ustrukturert tekst, før man til slutt presenterer den praktiske gjennomføringen av arbeidet.

## 4.1 Basismetoder for begrepsekstraksjon

Mange av teknikkene brukt i en semi-automatisk tilnærming kommer fra fagområder som tekst- og datamining. De neste avsnittene presenterer kort noen vanlig brukte teknikker.

### 4.1.1 Begrepsuttrekning

De fleste av de eksisterende tilnærmingene bruker naturlig-språk tekst som input, og starter med å trekke ut begreper fra teksten. Teksten blir parset, og begreper blir trukket ut av teksten på bakgrunn av hyppighetsfrekvensen i dokumentsamlingen. Ord som har høy hyppighet i all form for tekst, og som dermed ikke er domenespesifikk, såkalte stoppord, blir fjernet sammen med begreper som har for lav frekvens til å være viktige. [Baeza-Yates and Ribeiro-Neto, 1999] En annen tilnærming for begrepsuttrekning er å bruke eksisterende lingvistiske mønster i språket for å trekke ut begreper eller fraser som er viktige. [Blomqvist, 2005]

### 4.1.2 Ordklassifisering

En vanlig oppgave i en semi-automatisk tilnærming er å klassifisere ordene i dokumentene. En måte er å klassifisere begrepene etter hvilken ordklasse de tilhører. Dette kan gjøres ved å bruke såkalt "part-of-speech"-tagging som er en svært vanlig lingvistisk analysemetode. Taggingen kan gjøres ved å bruke et større sett håndskrevne regler (regelbasert tagging) [Brill, 1992], eller ved å tagge på bakgrunn av statistiske beregninger (stokastisk tagging) [Manning and Schütze, 1999].

### 4.1.3 Assosiasjonsregler

Bruk av assosiasjonsregler er en vanlig datamining teknikk. Teknikken går ut på å oppdage relasjoner mellom begreper i dokumentsamlinger. I et klassisk eksempel hentet fra [Holt and Chung, 1999] ser en på en handlekurv som en transaksjon, og hver handlekurv inneholder en mengde varer. Ved å sammenligne en stor mengde kurver kan man komme frem til en assosiasjonsregel ala "hvis en handlekurv inneholder A, så vil den også inneholde B".

## 4.2 Fraseekstraksjonssystem

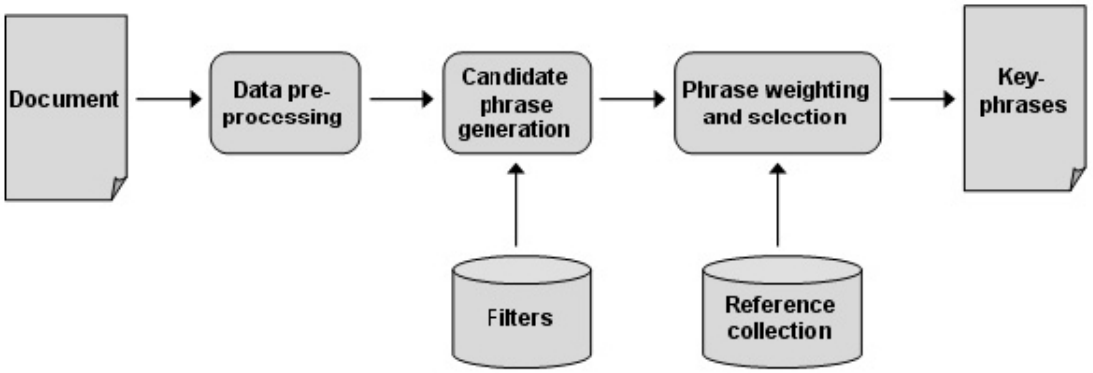
I dette avsnittet ser man kort på fraseekstraksjonssystemet generelt, og gir en kort beskrivelse av systemet som er brukt i forbindelse med dette arbeidet.

Nøkkelfraseekstraksjon er prosessen med å trekke ut et optimalt sett av nøkkelfraser eller nøkkelbegreper for å beskrive et dokument. Dette kan gjøres på to måter:

- Overvåket nøkkelfraseekstraksjon
- Automatisk nøkkelfraseekstraksjon

Ved overvåket nøkkelfraseekstraksjon bruker en dokumenter som inneholder forhåndsdefinerte nøkkelfraser til å trene opp ekstraksjonsalgoritmen, såkalt machine learning. Ved automatisk nøkkelfraseekstraksjon bruker man derimot kun statistiske data fra en referansesamling for å beregne signifikansen til et begrep eller frase i et dokument i forhold til resten av dokumentene, og rangerer så begrepene/frasene utifra oppnådd score. [Gulla et al., 2006]

I dette arbeidet har man benyttet et automatisk fraseekstraksjonssystem utviklet ved IDI høsten 2005. [Borch, 2005] Arkitekturen til systemet er vist i 4.1, og kjernekomponentene i systemet er *data preprocessing, candidate phrase generation* and *phrase weighting and selection.*




Figur 4.1: Arkitekturen til fraseekstraksjonssystemet [Borch, 2005]

**Data pre-processing:** Denne komponenten tar inn tekst fra dokumentfilene, fjerner diverse uønskede spesialtegn, fjerner stoppord dersom ønskelig, normaliserer (stemming) teksten, og utfører part-of-speech tagging på dokumentene.

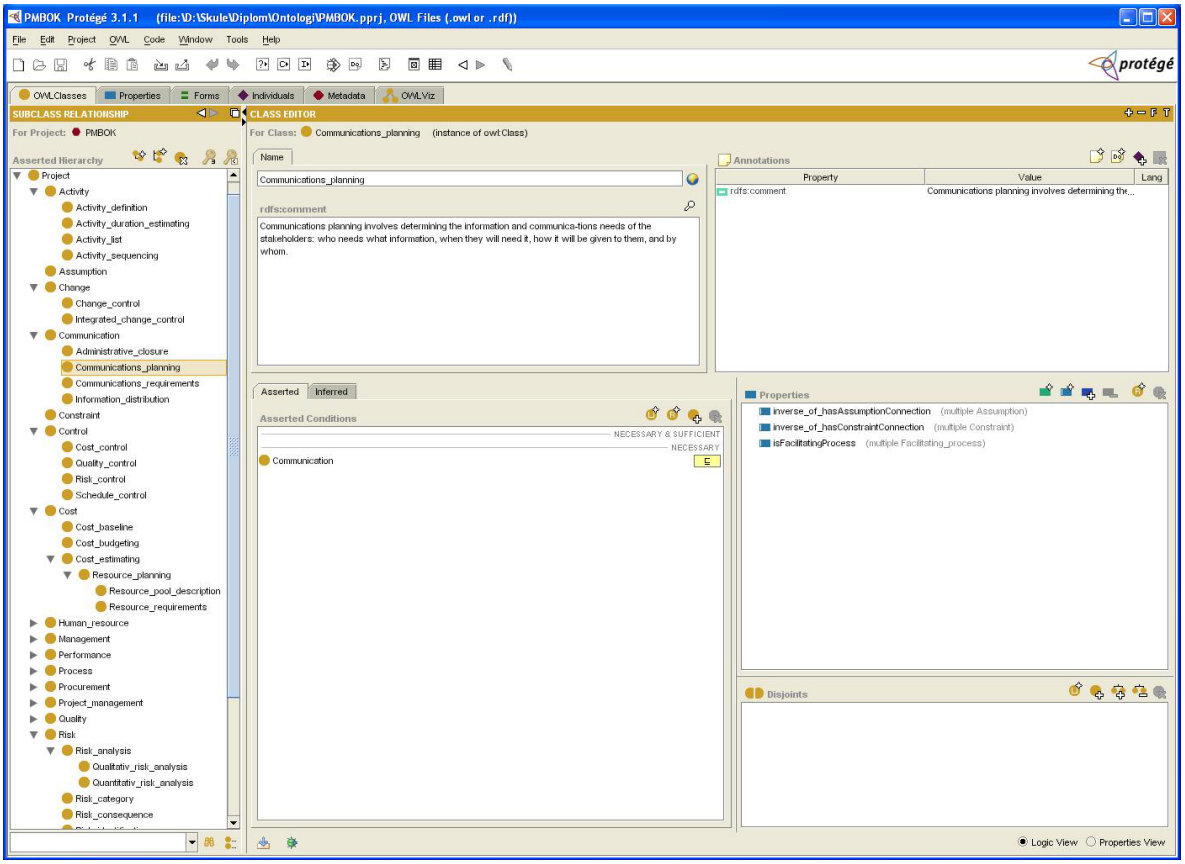
**Candidate phrase generation:** Denne komponenten bruker lingvistiske filter for å velge ut settet av begreper/fraser som med høy sannsynlighet er nøkkelfraser. I systemet blir disse kalt *candidate phrases.*

**Phrase weighting and selection:** Denne komponenten gir vekt til nøkkelfrasene, rangerer dem etter vekt, og returnerer det endelige settet med nøkkelfraser.

## 4.3 Praktisk gjennomføring

Jobben med å utvikle den semi-automatiske ontologien var langt mindre kompleks enn med den manuelle tilnærmingen. Prosessen startet med å konvertere PMBOK fra PDF-format til vanlig TXT-format, for så å kjøre disse gjennom fraseekstraksjonssystemet. Resultatene ble deretter plottet inn i et regneark, og sendt til domeneekspert for evaluering. Domeneeksperten gav deretter poeng til hver enkelt frase utifra skjemaet vist i tabell 4.1.

Etter å ha fått returnert resultatene fra domeneeksperten forkastet man alle resultatene med verdi 0, og startet så jobben med å sette sammen hierarkiet. Ettersom man utviklet den semi-automatiske ontologien etter den manuelle, og dermed hadde fått opparbeidet seg domenekunnskap, slapp man å sette seg dypt inn i dokumentene for å skjønne hvilke begreper og synonymer som var relatert til hverandre. For i det hele tatt å få til et hierarki som gav mening la man også til




Figur 4.2: Brukergrensesnittet i Protégé

| Poeng | Beskrivelse |
|---|---|
| 0 | Irrelevant |
| 1 | OK |
| 2 | Bra |

Tabell 4.1: Poengskjema brukt av domeneekspert

enkelte generelle begreper i ontologien. De begrepene som ble lagt til er generelle begreper i domenet, og befinner seg derfor på det høyeste nivået i ontologien. På lik linje med den manuelle ontologien ble også denne modellert i Protégé, hvis brukergrensesnitt er vist i figur 4.2.

Som nevnt i kapittel 3 har ikke bruken av attributter vært prioritert i dette arbeidet, og den semi-automatiske ontologien inneholder derfor ingen attributter. Det finnes heller ingen skriftlige definisjoner av hvert enkelt begrep i denne ontologien.

Domeneekspertens evaluering av de foreslåtte nøkkelfrasene ligger vedlagt i tillegg F, mens de faktisk returnerte frasene fra fraseekstraksjonssystemet er vedlagt i tillegg D.

### 4.3.1 Lærdom

I etterkant av arbeidet med den semi-automatiske ontologien sitter man igjen med endel erfaringer:

**Tidsbruk:** I forhold til byggingen av den manuelle ontologien tok denne tilnærmingen langt kortere tid. Hovedgrunnen til dette var selvfølgelig at fraseekstraksjonssystemet hentet ut begrepene fra dokumentet automatisk. En annen viktig årsak er det faktum at man hadde opparbeidet seg domenekunnskap under byggingen av den manuelle ontologien, noe som gjorde at man raskere klarte å se sammenhengen mellom begreper, som igjen førte til raskere oppbygning av hierarkiet. Dette viser at en viss kunnskap om domenet er en fordel, også ved semi-automatisk ontologikonstruksjon.

**Detaljnivå:** Et annet observert problem var at det ble vanskeligere å tilpasse detaljnivået i ontologien med den semi-automatiske tilnærmingen. Faren er at man kan komme opp i situasjoner hvor man har to begreper eller fraser som er relatert, men hvor de ene er veldig generell mens det andre er veldig spesifikk. Dette kan få negative konsekvenser for søkeontologier hvor det er en fordel å ha en gradvis stigning i detaljnivået på de ulike nivåene i hierarkiet. Dette for å unngå at en svært generell spørring kan risikere å bli utvidet med et svært spesifikt begrep, og dermed få returnert et resultatsett som inneholder svært spesifikk informasjon, mens man egentlig var ute etter mer generell informasjon.

# Kapittel 5

# Evaluering

I dette kapitlet presenteres resultatene, og evalueringen av resultatene, fra tester med de to utviklede ontologiene som er presentert i kapittel 3 og 4. Evalueringen er tredelt, og består av følgende deler:

- Del 1 - Sammenligning av ontologiene.
- Del 2 - Vurdering av ontologienes kvalitet utfra et sett kvalitetskrav.
- Del 3 - IR-evaluering.

## 5.1 Evalueringsbakgrunn

I litteraturen presenteres det en rekke ulike tilnærminger for evaluering av ontologier, avhengig av hvilke typer ontologier som evalueres og hva som er hensikten med dem. [Brank et al., 2005] Generelt sett kan kan si at de fleste evalueringsteknikkene faller inn under disse fire kategoriene:

- Evaluering av ontologien oppimot en "gullstandard" (som i seg selv kan være en ontologi). [Maedche and Staab, 2002]
- Bruk av ontologien i en applikasjon og evaluere resultatene.
  [Porzel and Malaka, 2004] [Paralic and Kostial, 2005]
- Evaluering av ontologi oppimot en kilde av data, for eksempel en dokumentsamling, som beskriver domenet ontologien skal beskrive. [Brewster et al., 2004]
- Evalueringer hvor mennesker prøver å fastsette hvor bra en ontologi møter et forhåndsbestemt sett kriterier, standarder, krav osv.
  [Lozano-Tello and Gomez-Perez, 2004]

I tillegg til de ovenfornevnte kategoriene velger man ofte å bryte ned evalueringen i flere nivåer ettersom ontologier kan bli svært komplekse, og det derfor er mer praktisk å holde fokus på ett nivå om gangen. Ifølge [Brank et al., 2005] er de ulike nivåene noe noe forskjellig definert hos ulike forfattere, men innholdet er i store trekk de samme og inneholder normalt disse nivåene:

1. *Leksikal-, vokabular-, eller datanivå:* På dette nivået fokuserer man på hvilke begreper, instanser og egenskaper som har blitt lagt til i ontologien, samt vokabularet som ligger til grunn for disse elementene. En vanlig møte å gjennomføre evaluering på dette nivået er ved inspeksjon av data som omhandler domenet ontologien representerer, for eksempel en dokumentsamling.

2. *Hierarki og semantiske relasjoner:* En ontologi består vanligvis av et hierarki av begreper med relasjonen mellom begrepene. Relasjonene er normalt IS-A, part-of, disjunksjon eller invers som vist i ontologispektrumet i figur 2.2. Evaluering av dette punktet innebærer inspeksjon og kontroll av at hierarkiet og relasjonene mellom begrepene er korrekt i henhold til domenet.

3. *Kontekst- eller applikasjonsnivå:* En ontologi kan i enkelte tilfeller være del av en større samling ontologier hvor det kan finnes referanser til definisjoner i andre ontologier. I slike tilfeller er det viktig å ta konteksten inn i betraktningen i evalueringen. I de tilfellene hvor ontologien opererer sammen med en applikasjon vil evalueringen bestå i å se på hvordan resultatene til applikasjonen blir påvirket av ontologien.

4. *Syntaktisk nivå:* Dette nivået er spesielt interessant for ontologier som er konstruert rent manuelt. Ontologier er normalt uttrykt i et formelt språk som for eksempel OWL eller RDF, og må derfor følge de syntaktiske regler som gjelder for dette språket. Videre bør en kontrollere og evaluere andre aspekter slik som naturlig-språk dokumentasjon i ontologien. [Gomez-Perez, 1994] For å sjekke at innholdet i ontologien er konsistent og korrekt i henhold til språkets syntaks er det normalt og bruke et program som RacerPro [Racer-Systems, 2006] som automatisk sjekker ontologien for feil.

5. *Struktur, arkitektur, design:* Som i punktet over er dette nivået interessant for manuelt konstruerte ontologier. Her evaluerer en om hvorvidt eventuelle forhåndsstilte krav til slik som oppbygning, struktur og mulighet til senere utvidelse er korrekt. [Gomez-Perez, 1994] [Gomez-Perez, 1996]

For evalueringen i dette arbeidet vil alle de fem nivåene over være interessante i forhold den evalueringsplanen som er presentert i innledningen til kapitlet.

## 5.2 Del 1 - Sammenligning av ontologiene

Dette avsnittet ser på likheter og forskjeller i de to ontologienes oppbygning og struktur. Ser en denne delen av evalueringen oppimot nivåene som er presentert i avsnitt 5.1 vil denne delen passe inn under nivå 1.

### 5.2.1 Klasser og attributter

Klasser er kanskje det viktigste elementet i en ontologier ettersom det er nettopp de som i stor grad beskriver domenet. Tabell 5.1 og 5.2 viser antallet klasser på de ulike nivåene i ontologien. Med nivå menes i dette tilfellet hvor dypt klassen ligger i hierarkiet.

| **Nivå** | **Antall klasser** |
|---|---|
| 1 | 1 |
| 2 | 19 |
| 3 | 91 |
| 4 | 26 |
| 5 | 5 |
| **Totalt** | 142 |

Tabell 5.1: Antall klasser i manuell ontologi

| **Nivå** | **Antall klasser** |
|---|---|
| 1 | 13 |
| 2 | 80 |
| 3 | 13 |
| **Totalt** | 106 |

Tabell 5.2: Antall klasser i semi-automatisk ontologi

Som en ser av tabellene har den manuelle ontologien flere klasser enn den semi-automatiske ontologien. Det kan være både fordeler og ulemper med mange klasser i en ontologi for søk ettersom for mange klasser gjør at rollen til ontologien utspiller seg, mens for få klasser gjør at viktige begreper i domenet ikke får behandling som fortjent i søkeprosessen, noe som kan gå ut over presisjonen til søket. Når det gjelder antallet klasser for de to ontologiene har man hatt to klare begrensninger:

- Selv om PMBOK er en form for gullstandard for den kunnskapen som finnes innenfor domenet ”Project management” vil det være naivt å tro at det ikke

finnes annen dokumentasjon med flere begreper som også med rette hadde fortjent en plass i ontologien.

- I byggingen av den semi-automatiske ontologien har man tatt utgangspunkt i de 180 frasene ekstrahert av systemet presentert i kapittel 4. Et slikt system vil aldri kunne gi et 100% riktig svar på hva som er de mest relevante frasene i domenet.

Forskjellen i antall klasser er likevel ikke spesielt stor, og når en intern domeneekspert (IDI) i begynnelsen av arbeidet gjorde et anslag på at den manuelle ontologien ville bestå av rundt 100-120 klasser vil en kunne anta at klasseantallet i de to ontologiene er naturlig i forhold til dokumentsamlingen de representerer.

Som nevnt i kapittel 3 og 4 har begreper og hierarki hatt fokus i dette arbeidet ettersom det er de elementene som brukes i IR-testen senere i kapitlet. Attributter (properties) er derfor blitt nedprioritert i dette arbeidet, men det er likevel blitt laget 34 attributter i den manuelle ontologien

### Synonymer

For ontologier brukt i søk vil synonymer spille en viktig rolle. Dette fordi synonymer til en klasse vil kunne bli lagt til brukerens spørring automatisk, eller at brukeren kan opplyses om at synonymer til begrepet/frasen han søker etter eksisterer. På denne måten kan brukeren også få treff i dokumenter som bare inneholder synonymet til spørringen.

Tabell 5.3 viser antallet synonymer i de to ontologiene, og hvor mange klasser disse synonymene er fordelt på.

| **Ontologi** | **Antall** | **Antall klasser** |
|---|---|---|
| Manuell | 26 | 13 |
| Semi-automatisk | 6 | 6 |

Tabell 5.3: Synonymer i ontologiene

Også her har den manuelle ontologien flere elementer enn den semi-automatisk ontologien. Dette har en naturlig årsak ettersom en, som nevnt tidligere, kun har tatt utgangspunkt i resultatene fra fraseekstraksjonssystemet i byggingen av den semi-automatiske ontologien. Siden bruken av begreper i PMBOK er veldig konsistent opptrer ofte synonymene bare en gang i hele dokumentet, og blir derfor ikke ansett som viktig av systemet. Når det gjelder synonymene i den manuelle ontologien har man tatt med de som står i klartekst i PMBOK, og ikke gjort studier av hvilke andre synonymer til de ulike begrepene som finnes i annen dokumentasjon om domenet.

### 5.2.2 Begreper og definisjoner

Som nevnt i 5.2.1 er klasser et svært viktig element i en ontologi. Det som er enda viktigere er hva klassen inneholder, nemlig begreper eller fraser. Av begrepene/frasene som er valgt til de to ontologiene er 62 stykker like. For å sikre kvaliteten til begrepene i ontologien har alle begrepene blitt evaluert av en domeneekspert utifra poengskjemaet vist i tabell 5.4.

| **Poeng** | **Beskrivelse** |
|---|---|
| 0 | Irrelevant |
| 1 | OK |
| 2 | Bra |

Tabell 5.4: Poengskjema brukt av domeneekspert

Tabell 5.5 og 5.6 viser poengfordelingen for begrepene som er brukt i de to ontologiene. Bare begrepene med poengverdi 1 og 2 ble brukt i ontologibyggingen, mens begrepene med poengverdi 0 ble forkastet.

| **Poeng** | **Antall** | **% av total** |
|---|---|---|
| 1 | 20 | 14% |
| 2 | 122 | 86% |

Tabell 5.5: Poengfordeling manuell ontologi

| **Poeng** | **Antall** | **% av total** |
|---|---|---|
| 1 | 33 | 21% |
| 2 | 73 | 79% |

Tabell 5.6: Poengfordeling semi-automatisk ontologi

Tabellene viser at også her har den manuelle ontologien et kvalitetsforsprang i forhold til den semi-automatiske ettersom prosentandelen begreper ved verdi 2 er høyere. Dette er imidlertid forventet ettersom en manuell gjennomgang av dokumentsamlingen må kunne ansees å være betraktelig grundigere enn den gjort av fraseekstraksjonssystemet. Forskjellen i prosentandel 2'ere er likevel ikke skremmende stor, og en må derfor kunne si at fraseekstraksjonssystemet har fungert bra på dokumentsamlingen som er brukt. Det må likevel nevnes at 74 av de totalt 180 returnerte begrepene/frasene fra systemet fikk 0 poeng, det vil si irrelevant, fra domeneeksperten. Dette utgjør 41% av de 180 frasene.

I sin begrunnelse for valg av poeng til hvert begrep tar domeneeksperten utgangspunkt i at brukeren har en viss kjennskap til domenet:

> ”Merk at jeg forutsetter at den som bruker dette vet at konteksten er prosjektledelse. Min score går derfor på hvor lett det er for en som kan litt om prosjekledelse å skjønne av begrepene at det dreier seg om det temaet som de forskjellige kapitlene fokuserer på. Hvis jeg skal gi score basert på at brukeren ikke vet hvilket domene det er, men kan litt om prosjektledelse, blir scoren en del lavere. Hvis jeg skal gi score basert på at brukeren ikke vet hvilket domene det er og ikke kan noe særlig om prosjektledelse, blir scoren mye lavere.”
>
> *Domeneekspert Statoil*

Utifra det scenariet man ser for seg at de to utviklede ontologiene vil kunne bli brukt, det vil si søk etter dokumenter relatert til prosjetledelse innad i en organisasjon, vil det være naturlig at brukeren har en viss kjennskap til domenet. Med bakgrunn i dette og kommentaren til domeneeksperten vil en dermed kunne si at begrepene ontologiene består av er gode i forhold til tenkt bruk.

En annen forskjell ved de to ontologiene er at den manuelle har en definisjon knyttet opp mot hvert enkelt begrep i ontologien. Fordelen med dette er først og fremst at det blir enklere å vedlikeholde ontologien senere ved at man har full oversikt over hva som ligger i begrepet, eller at definisjonen kan vises til brukeren for at han skal få en rask oversikt over hva som ligger i begrepet. For at definisjonene skal stå til det som faktisk menes med begrepet har man i utviklingen av ontologien kopiert definisjonene direkte fra PMBOK. På denne måten får man en tettere tilknytning mellom ontologien og domenet den representerer.

## 5.3 Del 2 - Kvalitetsevaluering

I denne delen av evalueringen tar man utgangspunkt i kvalitetskrav som ble identifisert i prosjektoppgaven fra høsten 2005 [Grimnes, 2005], og evaluere om hvorvidt de to ontologiene oppfyller disse kravene. Ser en denne delen av evalueringen oppimot nivåene som er presentert i avsnitt 5.1 vil denne delen passe inn under nivåene 2, 4 og 5.

### 5.3.1 Kvalitetskrav

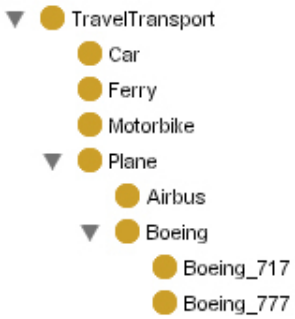
Selv om alle ontologier må oppfylle visse felles krav som for eksempel korrekt syntaks i forhold til språket den er utviklet i, vil den tiltenkte bruken av ontologien også spille inn i avgjørelsen om hvilke krav som stilles til den. For ontologier brukt i søk ble det i forbindelse med høstprosjektet identifisert følgende seks krav:

- Gode begreper som beskriver domenet.
- Relasjoner (attributter) mellom begreper.
- Ingen tvetydigheter.
- Kobling mellom begrep i ontologi og begreper i dokumenter.
- Tilstrekkelig detaljnivå.
- Strenge hierarkier.

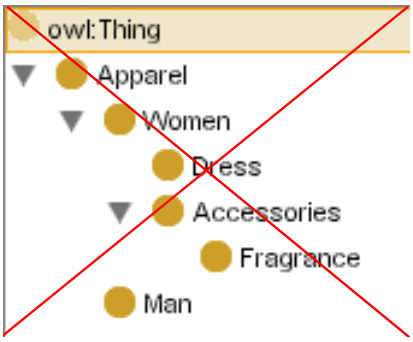
De tre sistnevnte evalueringskriteriene trenger litt utdypning. Med kobling mellom begreper i ontologi og begreper i dokumenter menes det at begrepene i ontologien også må være representert i tekst i dokumentene. For å forsterke denne koblingen bør man koble synonymer til begrep i ontologien.

Det trengs også et "passende" detaljnivå i ontologien. For mange begreper vil begrense den ønskede effekten av ontologien, samtidig som det vil være bortkastet å modellere informasjon det aldri blir søkt etter. Dersom ontologien er for grovkornet vil man kunne gå glipp informasjon, noe som også vil begrense effekten av ontologien.

Med strenge hierarkier menes hierarkier hvor subklassene alltid har en naturlig semantisk relasjon til klassene over i hierarkiet. Et eksempel på et slikt hierarki er vist i figur 5.1. Klassen "Boeing_717" er av typen "Boeing", som er en type "Plane", som igjen er en form for "TravelTransport". Et eksempel på en dårlig oppbygd hierarkisk struktur er vist i figur 5.2. Klassen "Fragrance" er riktignok en type "Accessory", men ikke en type "Women" som igjen ikke er en type "Apparel". Effekten av å bruke et slikt hierarki vil få alvorlige konsekvenser i tilfeller hvor man bruker hierarkiet til å spesialisere eller generalisere søkestrengen.




Figur 5.1: Strengt IS-A hierarki




Figur 5.2: Dårlig oppbygd hierarki

På de neste sidene følger en kort beskrivelse av hva som er gjort for å oppfylle kravene.

### Gode begreper som beskriver domenet

Som nevnt i avsnitt 2.2 er ontologi definert som "en delt enighet om et domene." Det er derfor viktig at ontologien inneholder de begrepene som er vesentlige og nødvendige for å kunne beskrive domenet. For å sikre at dette kravet er oppfylt har man benyttet domeneeksperter til evalueringen av begrepene. Bruk av personer med domeneekspertise for å sikre at de riktige nøkkelbegrepene blir identifisert er også omtalt i litteraturen, blant annet i [Uschold and Gruninger, 1996] og [de Almeida Falbo et al., 1998].

Med bakgrunn ovenfornevnte punkter, samt at alle begreper har vært gjennom en seleksjonsprosess vil en kunne konkludere med at begrepene som finnes i de to ontologiene oppfyller dette kravet til kvalitet.

### Relasjoner (attributter) mellom begreper

Som nevnt tidligere i rapporten har attributter blitt nedprioritert i dette arbeidet da man har valgt å ikke bruke disse elementene i forbindelse med den praktiske IR-testen som beskrives i avsnitt 5.4. Selv om den manuelle ontologien inneholder enkelte relasjoner er det for få til at en kan si at kravet er oppfylt. På den annen side er ikke oppfyllelse av dette kravet viktig i seg selv ettersom det helt fra starten av arbeidet har vært klart at attributter ikke skal brukes i IR-testen.

### Ingen tvetydigheter

Tvetydige begreper er et problem i ontologier ettersom det kan skape forvirring rundt hva som faktisk ligger i begrepet, og dermed også usikkerhet rundt enigheten om domenet. Ettersom ontologiene er utviklet med bakgrunn i PMBOK blir faren for tvetydige begrep mindre fordi dokumentet inneholder svært klare definisjoner av alle viktige begreper. I tillegg til dette er prosjektledelse-dokumentasjon fra Project Management Institute [(PMI), 2006] svært mye brukt i industrien, deriblant i Statoil, og terminologien brukt i utgivelsene vil derfor i stor grad være kjent for aktuelle brukere av et eventuelt søkesystem hvor ontologiene er inkorporert. Det ble heller ikke funnet noen tvetydigheter ved gjennomgangen med domeneekspertene.

### Kobling mellom begrep i ontologi og begreper i dokumenter

For at ontologier skal ha noen form for effekt i en søkesammenheng er det viktig at begrepene i ontologien "matcher" de begrepene som finnes i dokumentsamlingen. Naturlig nok har man derfor valgt å kopiert begrepene direkte fra PMBOK i utviklingen av den manuelle ontologien, mens bruken av fraseekstraksjonsystemet gjør dette skjer automatisk i forhold til den semi-automatiske ontologien. Ettersom man aktivt har brukt synonymer i utviklingen av ontologiene blir koblingen mellom ontologi og dokument blir enda sterkere.

### Tilstrekkelig detaljnivå

Dette kravet er vanskelig å måle om hvorvidt er oppfylt eller ikke. For den manuelle ontologien kan man til en viss grad uttale seg ettersom man der har benyttet en middle-out strategi til å identifisere nøkkelbegrepene i PMBOK. Ved middle-out starter man med de grunnleggende begrepene for hver gruppe før man ser på mer generelle og spesialiserte begrep. Dette gjør at man får en naturlig balanse i detaljnivået, men om denne balansen er tilstrekkelig er det vanskelig å måle med 100% sikkerhet.

Ved å bruke en semi-automatisk tilnærming mister man lett kontrollen over detaljnivået ettersom det er fraseekstraksjonssystemet som foreslår og til en viss grad bestemmer hvilke begreper som skal brukes i ontologien. På grunn av dette vil kravet om tilstrekkelig detaljnivå bli vanskeligere å oppfylle ettersom deler av det menneskelige aspektet i beslutningstakingen forsvinner.

Ettersom begrepene brukt i de to ontologiene er hentet fra PMBOK alene vil det være langt letter å tilfredstille dette kravet for ontologiene utviklet i dette arbeidet ettersom dokumentet er såpass lite og oversiktlig. I ontologier bygget med begreper samlet inn fra store dokumentsamlinger mister man lettere kontrollen over innholdet, og dermed også detaljnivået. Det å holde oversikten over innholdet og detaljnivået i PMBOK har vært en forholdsvis enkel oppgave.

### Strenge hierarkier

For å oppfylle dette kravet må man fokusere på at hierarkiet i ontologien er riktig oppbygd og at subklasser dermed hører naturlig inn under klassene over i hierarkiet. Som nevnt i kapittel 3 og 4 har PMBOK en veldig fast struktur hvor hvert kapittel inneholder et spesifikt kunnskapsområde innen prosjektledelse. Selv om man ikke fokuserte spesifikt på å sette opp hierarkiene på bakgrunn av kapitlene, ble de brukt som en referanse i etterkant for å sjekke at hierarkiene

var noenlunde riktig satt opp. For å få verifisert at hierarkiet faktisk var korrekt benyttet man så en domeneekspert som gav sin tilbakemelding til strukturen.

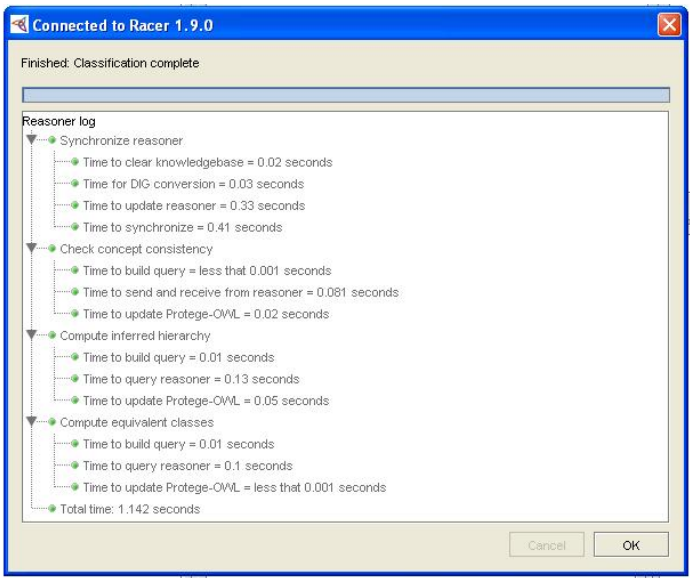
#### Oppsummering

Generelt sett kan man si at ontologiene oppfyller de gitte kravene på en god måte. Riktignok er kravet om attributter mellom begrepene ikke oppfylt, men utifra konteksten som ontologiene er tenkt brukt er ikke en oppfyllelse av dette kravet en nødvendighet. Man kan likevel skimte en viss kvalitetsforskjell mellom de to ettersom det er lettere å kontrollere detaljnivået i en manuelt konstruert ontologi, men som nevnt har detaljnivået i dette tilfellet vært forholdsvis greit å ha kontrollen over ettersom dokumentsamlingen har en begrenset størrelse.

Sannsynligheten for at en generell kvalitetsforskjell mellom de to ontologiene hadde blitt mye mer synlig dersom dokumentsamlingen ontologiene var konstruert på bakgrunn av hadde vært større synes veldig klart.

### 5.3.2 Konsistenssjekk

For å sikre at ontologiene er konsistente i forhold til OWLs syntaks er det blitt kjørt kontinuerlige konsistenssjekker for begge ontologiene gjennom hele byggeprosessen. Til dette har en brukt RacerPro [Racer-Systems, 2006], et resoneringsverktøy utviklet av Racer Systems GmbH & Co. KG. Figur 5.3 viser et skjermbilde fra Racer etter en utført test.




Figur 5.3: Konsistenssjekk med RacerPro

## 5.4 Del 3 - IR-evaluering

Dette avsnittet presenterer gjennomføringen, resultatene og evalueringen av resultatene fra IR-testen utført i arbeidet. Testen er gjennomført med hjelp fra seks medstudenter/stipendiater som har studert, og evaluert, resultatene fra et implementert IR-system som er støttet av begreper og fraser fra de to ontologiene. Ser en denne delen av evalueringen oppimot nivåene som er presentert i avsnitt 5.1 vil denne delen passe inn under nivå 3.

### 5.4.1 Aspekter ved IR-evaluering

Tradisjonelle IR tilnærminger blir blir evaluert utifra ytelseskriterier - evaluering av systemytelse og evaluering av gjenfinningsytelse. [Baeza-Yates and Ribeiro-Neto, 1999] Ved evaluering av systemytelse måler man vanligvis enheter som for eksempel responstid og størrelse på indeks. Gjenfinningskvalitet evalueres som oftest på bakgrunn av målinger av precision og recall.

Evaluering av systemytelse er ikke relevant for denne oppgaven. Dette fordi det er resultater for ontologistøttet IR som skal evalueres, og fordi det implementerte systemet er Lucene-basert og en evaluering av dette dermed ville blitt en evaluering av Lucene [Apache, 2006], noe som er uinteressant for denne oppgaven.

Målinger av precision og recall er utviklet for å kunne måle settet av relevante dokumenter utifra resultatsettet, og sammenligne dette med det relevante settet av dokumenter i hele samlingen. To åpenbare problemer med disse målingene er at relevanse er en subjektivt mål, og at det relevante dokumentsettet til en gitt spørring vanligvis ikke er kjent. I standard gjenfinningstester, slik som TREC-serien [NIST, 2006], bruker man domeneeksperter til å definere hvilke dokumenter som er relevant for en gitt spørring.

I tilnærmingen brukt i dette arbeidet har man valgt å sette brukerens (i dette tilfellet de seks medstudentene/stipendiatene som har evaluert resultatene) evaluering av de returnerte dokumentene i fokus. I denne sammenhengen er det viktig å påpeke at hver bruker vil ha sin subjektive mening om hva som er relevante dokumenter og hva som er ikke-relevante dokumenter utifra en gitt spørring. Brukernes oppfatning av relevans vil også endre seg etterhvert som de får mer kunnskap om domenet. [Brasethvik, 2004]

Med bakgrunn i diskusjonen over definerer vi følgende punkter som skal måles i denne delen av evalueringen:

- **Oppfattet relevans:** Oppfattet relevans er definert som en brukers subjektive oppfatning av et dokuments relevans på bakgrunn av en gitt spørring. I evalueringen skal dokumentene gis poeng på bakgrunn av denne relevansen.

- **Komparativ relevans:** Ved å bruke de samme spørringene, men med ulik ontologi i IR-systemet, vil man få to forskjellige resultatsett som gis poeng, og man kan dermed sammenligne de totale poengsummene.

Evalueringer basert på oppfattet og komparativ relevans er en vanlig evalueringsmetode i dagens kommersielle søkemotorer. [Gulla, 2003] [Hawking et al., 2001] Hvordan poengfordeling er satt opp, og hvordan man beregner de totale poengsummene blir nærmere beskrevet i 5.4.2.

### 5.4.2 Evalueringstilnærming

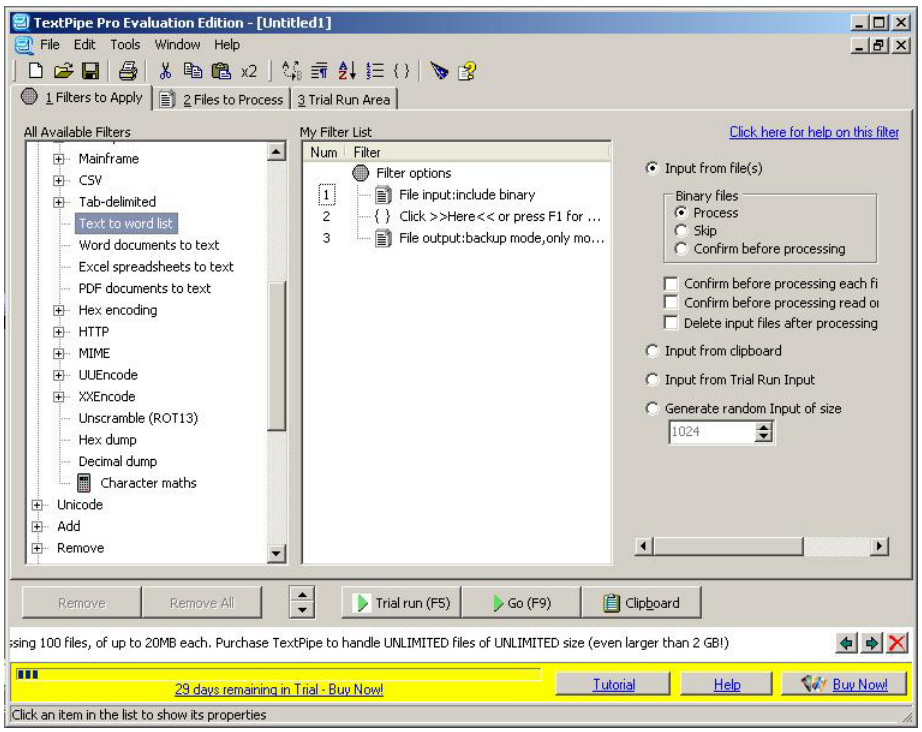
Dette avsnittet beskriver hvordan IR-testen ble satt opp og gjennomført. Det gis en beskrivelse av dokumentsamlingen, søkesystemet og spørringene som er blitt brukt i testen, samt en beskrivelse av hvordan poengene basert på relevans blir utdelt og hvordan ontologiene er blitt brukt i selve søket.

#### Dokumentsamlingen

Teksten som er brukt i testen består av de tolv kapitlene i boka "A Guide to the Project Management Body of Knowledge" (PMBOK) [(PMI), 2000], og de seks første kapitlene i boka "Organizational Project Management Maturity Model" (OMP3) [(PMI), 2003]. Kapitlene ble først omgjort til rene tekstfiler før man klippet og limte dem sammen til en dokumentsamling bestående av totalt 76 dokumenter.

Ettersom begge de to bøkene opprinnelig har både topp og bunntekst på hver enkelt side måtte dette fjernes. Dette ble gjort for at evaluatorene skulle få så "rene" dokumenter som mulig å forholde seg til. Til dette brukte man en evalueringskopi av programmet Textpipe Pro [DataMystic, 2006] som automatisk fjerner den teksten man spesifiserer. Dokumentsamlingen ligger vedlagt under tillegg F.

Textpipe Pro ble også brukt på en annen kopi av dokumentsamlingen hvor annen type støy ble fjernet fra dokumentene slik som tall, punktum, komma og lignende. Denne versjonen av dokumentsettet ble så brukt av det implementerte søkesystemet til å indeksere dokumentsamlingen. Figur 5.4 viser brukergrensesnittet til Textpipe Pro.




Figur 5.4: Textpipe Pro brukergrensesnitt

### Lucene-implementasjon

I forbindelse med evalueringsarbeidet ble det implementert et enkelt søkesystem basert på Apache Lucene. [Apache, 2006] Siden systemet i seg selv ikke er viktig for dette arbeidet vil det ikke bli evaluert og omtalt videre i denne rapporten. For å trekke en parallell til avsnitt 2.3 nevnes det at systemet benytter teknikker som indeksering, stemming og fjerning av stoppord, samt at det er basert på vektormodellen. Kildekoden for de implementerte klassene ligger vedlagt i tillegg E.

### Bruk av ontologi

I søkeprosessen er det tenkt at ontologien skal brukes til å utvide brukerens spørring med relaterte begreper og synonymer fra ontologien. Dette er illustrert i kodesnutten på neste side hvor en ser at søk etter frasen ”human resource”, som finnes i ontologien, blir utvidet med synonymet ”hr” og de fire relaterte begrepene ”organizational planning”, ”staff”, ”staff acquisition” og ”team development”. Merk at frasen det søkes etter og synonymet til denne frasen får høy vekting i søket (1.5), mens de relaterte begrepene blir vektet lavere (1.0). En komplett liste over spørringene og utvidelser/synonymer fra hver av ontologiene finner man i tillegg A.

```
public ArrayList GetTestSet3()
{
ArrayList phrases = new ArrayList();
phrases.add(new Phrase("human resource",1.5f));
phrases.add(new Phrase("hr",1.5f));
phrases.add(new Phrase("organizational planning",1.0f));
phrases.add(new Phrase("staff",1.0f));
phrases.add(new Phrase("staff acquisition",1.0f));
phrases.add(new Phrase("team development",1.0f));

return phrases;
}
```

Som vist simulerer en altså bare ontologiens tilstedeværelse i søkesystemet. Resultatene ville uansett blitt de samme dersom ontologien faktisk var tilstede, gitt at strategien for ontologiens bruk var den samme.

### Spørringer

I forbindelse med IR-evalueringen er det satt opp et sett med spørringer som vist i tabell 5.7. Ettersom testen er en del av evalueringen til ontologiene er spørringene hentet fra begreper og fraser som finnes i ontologiene. Av tabellen ser en at spørringene 1-10 opptrer i begge ontologiene, spørringene 11-13 opptrer kun i den manuelle ontologien, mens spørringene 14-16 bare opptrer i den semi-automatiske ontologien. Bakgrunnen for denne strategien er at en både får testet ontologiene mot hverandre, og ontologiene oppimot vanlig fritekstsøk uten ontologistøtte.

For å få muligheten til å undersøke resultatene fra IR-testen i større detalj er spørringene kategorisert av som følger:

**C1:** Spørringer som er generelle for domenet, og hvor brukeren vil kunne akseptere en rekke forskjellige dokumenter som relevant for spørringen. Med andre ord en relativt vag spørring innen domenet.

**C2:** Spørringer som er spesifikke for domenet, og hvor bruker en kun er ute etter et mindre sett dokumenter som han anser som relevant.

Det er viktig å merke seg at å kategorisere spørringer på denne måten er en heller vag vitenskap ettersom hver enkelt bruker har et eget syn på hva hans informasjonsbehov faktisk er. [Hawking et al., 2001] I dette tilfellet er det undertegnede som har stått for kategoriseringen på bakgrunn av opparbeid kunnskap

| | **Spørring** | **Kategori** | **Manuell** | **Semi-automatisk** |
|---|---|---|---|---|
| 1 | cost estimating | C2 | X | X |
| 2 | human resource | C1 | X | X |
| 3 | stakeholder | C1 | X | X |
| 4 | procurement | C1 | X | X |
| 5 | risk response | C2 | X | X |
| 6 | work breakdown structure | C2 | X | X |
| 7 | earned value | C2 | X | X |
| 8 | earned value management | C1 | X | X |
| 9 | communication management | C1 | X | X |
| 10 | performance | C1 | X | X |
| 11 | planning process | C2 | X | |
| 12 | control account plan | C2 | X | |
| 13 | project management information system | C2 | X | |
| 14 | time management | C1 | | X |
| 15 | integration management | C1 | | X |
| 16 | cost performance index | C2 | | X |

Tabell 5.7: Spørringene brukt til evalueringen

om domenet gjennom arbeidet.

### Poengberegning og evalueringsskjema

For å beregne resultatene til hver spørring brukes en kalkuleringsstrategi beskrevet i [Gulla, 2003]. For hver spørring gir evaluatorene poeng til hvert av de 5 høyest rangerte dokumentene utifra skjemaet vist i tabell 5.8. Disse resultatene blir så kombinert med en vekting gitt av rangeringen til dokumentet (vist i tabell 5.9), og man beregner så den totale poengsummen til spørringen utifra formel 5.1.

| **Poeng** | **Beskrivelse** |
|---|---|
| -1 | Søppel |
| 0 | Ikke-relevant |
| 1 | Relatert dokument |
| 2 | Bra |

Tabell 5.8: Poenggivning i forhold til relevans

| Rangering | Vekt |
|---|---|
| 1 | 10 |
| 2 | 8 |
| 3 | 5 |
| 4 | 3 |
| 5 | 1 |

Tabell 5.9: Poenggivning i forhold til posisjon

Den totale poengsummen til en spørring blir beregnet utfra formelen:

$$Q_s = \frac{1}{2} * \sum_{i=1}^{5} P_{D_i} * P_{P_i} \tag{5.1}$$

hvor $P_{D_i}$ er den individuelle poengsummen for dokument $D_i$, og $P_{P_i}$ er vektfaktoren for posisjon $P_i$.

Figur 5.5 viser det evalueringsskjemaet som ble gitt til evaluatorene. Alle oddetallsrader ble brukt til å fylle inn resultatene fra den manuelle ontologien, mens partallradene var forbeholdt den semi-automatiske. Som nevnt i 5.4.2 finnes ikke spørringene i radene 22, 24 og 26 i den semi-automatiske ontologien, mens spørringene i radene 27, 29 og 31 ikke finnes i den manuelle ontologien.

| # | Query | Result 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | cost estimating | | | | | |
| 2 | | | | | | |
| 3 | human resource | | | | | |
| 4 | | | | | | |
| 5 | stakeholder | | | | | |
| 6 | | | | | | |
| 7 | procurement | | | | | |
| 8 | | | | | | |
| 9 | risk response | | | | | |
| 10 | | | | | | |
| 11 | work breakdown structure | | | | | |
| 12 | | | | | | |
| 13 | earned value | | | | | |
| 14 | | | | | | |
| 15 | earned value management | | | | | |
| 16 | | | | | | |
| 17 | communication management | | | | | |
| 18 | | | | | | |
| 19 | performance | | | | | |
| 20 | | | | | | |
| 21 | planning process | | | | | |
| 22 | | | | | | |
| 23 | control account plan | | | | | |
| 24 | | | | | | |
| 25 | project management information system | | | | | |
| 26 | | | | | | |
| 27 | time management | | | | | |
| 28 | | | | | | |
| 29 | integration management | | | | | |
| 30 | | | | | | |
| 31 | cost performance index | | | | | |
| 32 | | | | | | |

| Score | Description |
|---|---|
| -1 | Trash |
| 0 | Non-relevant |
| 1 | Realted |
| 2 | Good |

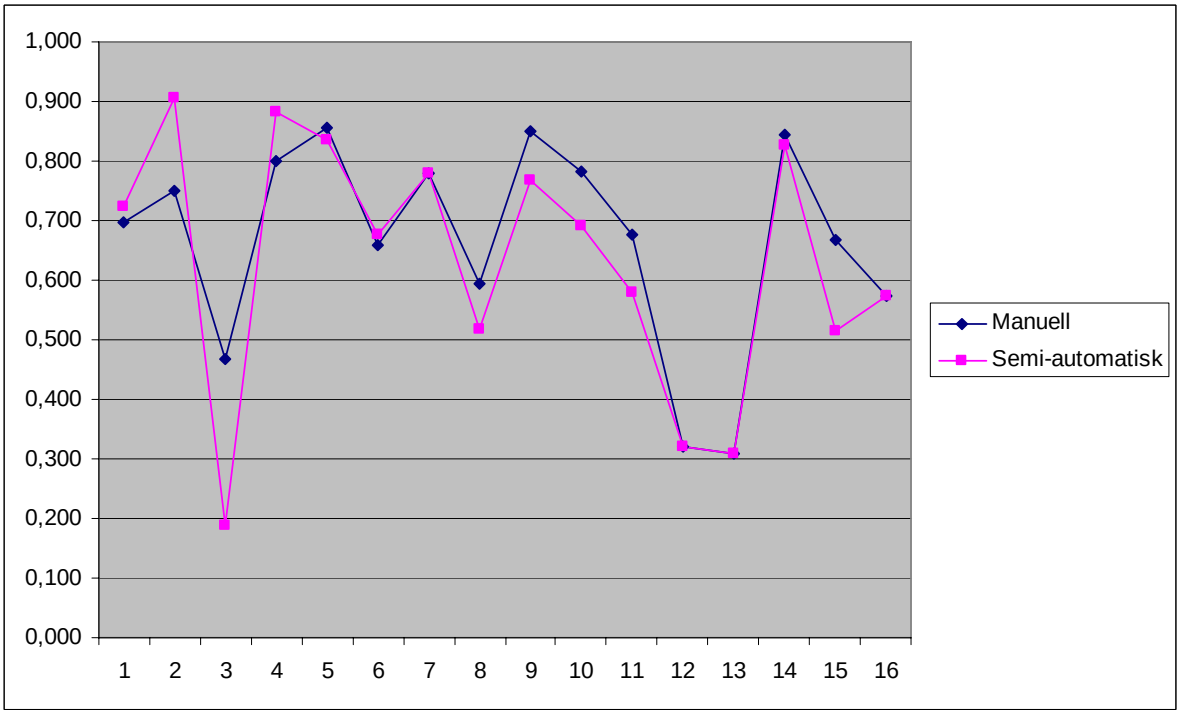
Figur 5.5: Evalueringsskjema

### 5.4.3 Resultater og diskusjon

Dette avsnittet presenterer resultatene og diskusjonen rundt IR-testen. Først presenteres de generelle resultatene og trendene, så ser en på mulige usikkerhetsmomenter før en til slutt studerer resultatene litt mer i detalj.

#### Overordnede resultater

Grafen i figur 5.6 viser de samlede resultatene fra IR-testen. Grafen viser de gjennomsnittlige resultatene for hver av de 16 spørringene etter test med begge de to ontologiene. De nøyaktige tallene bak grafen vises i tabell 5.10. Alle resultatene er normalisert ved at samtlige verdier for hver enkelt evaluator i det orginale resultatsettet er dividert med den høyest gitte poengsummen som vedkommende gav i evalueringen totalt sett. Normaliseringen utføres for at resultatene til enkeltpersoner ikke skal bli for dominerende. Noen er mer negative/positive enn andre og vil bruke -1 og 2 i mer utstrakt grad enn andre. For at dette ikke skal påvirke resultatet i for stor grad så har man valgt å normalisere resultatene. De komplette resultatene fra IR-testen, samt evalueringen fra hver enkelt evaluator finner man i henholdvis tillegg B og C.




Figur 5.6: Graf over snittet til de normaliserte resultatene

| # | Query | Results 1 | 2 | 3 | 4 | 5 | 6 | Snitt |
|---|---|---|---|---|---|---|---|---|
| 1 | cost estimating | 0,769 | 0,625 | 0,590 | 0,315 | 0,981 | 0,907 | 0,698 |
| 2 | | 0,519 | 0,729 | 0,513 | 0,796 | 0,815 | 0,963 | 0,723 |
| 3 | human resource | 0,712 | 0,896 | 0,744 | 0,444 | 0,852 | 0,852 | 0,750 |
| 4 | | 1,000 | 1,000 | 0,923 | 0,556 | 0,981 | 0,981 | 0,907 |
| 5 | stakeholder | 0,404 | 0,729 | 0,282 | 0,056 | 0,593 | 0,741 | 0,467 |
| 6 | | 0,096 | 0,500 | 0,128 | -0,500 | 0,259 | 0,648 | 0,189 |
| 7 | procurement | 0,731 | 0,917 | 0,949 | 0,796 | 0,704 | 0,704 | 0,800 |
| 8 | | 0,750 | 1,000 | 1,000 | 0,870 | 0,926 | 0,741 | 0,881 |
| 9 | risk response | 0,712 | 0,771 | 0,949 | 0,759 | 0,944 | 1,000 | 0,856 |
| 10 | | 0,673 | 0,729 | 0,897 | 0,759 | 0,944 | 1,000 | 0,834 |
| 11 | work breakdown structure | 0,500 | 0,771 | 0,590 | 0,352 | 0,981 | 0,759 | 0,659 |
| 12 | | 0,538 | 0,792 | 0,615 | 0,407 | 1,000 | 0,704 | 0,676 |
| 13 | earned value | 0,769 | 0,938 | 0,487 | 0,704 | 0,889 | 0,889 | 0,779 |
| 14 | | 0,769 | 0,938 | 0,487 | 0,704 | 0,889 | 0,889 | 0,779 |
| 15 | earned value management | 0,519 | 0,771 | 0,615 | 0,259 | 0,704 | 0,704 | 0,595 |
| 16 | | 0,404 | 0,708 | 0,590 | 0,148 | 0,630 | 0,630 | 0,518 |
| 17 | communication management | 1,000 | 1,000 | 0,821 | 0,481 | 0,907 | 0,889 | 0,850 |
| 18 | | 1,000 | 0,938 | 0,667 | 0,315 | 0,852 | 0,833 | 0,767 |
| 19 | performance | 0,712 | 0,792 | 0,538 | 0,852 | 1,000 | 0,796 | 0,782 |
| 20 | | 0,577 | 0,729 | 0,410 | 1,000 | 1,000 | 0,426 | 0,690 |
| 21 | planning process | 0,481 | 0,667 | 0,410 | 0,741 | 0,926 | 0,833 | 0,676 |
| 22 | | 0,500 | 0,583 | 0,410 | 0,463 | 0,852 | 0,667 | 0,579 |
| 23 | control account plan | 0,346 | 0,479 | 0,000 | -0,278 | 0,759 | 0,611 | 0,320 |
| 24 | | 0,346 | 0,583 | 0,000 | -0,185 | 0,667 | 0,519 | 0,322 |
| 25 | project management information system | 0,538 | 0,583 | 0,128 | -0,426 | 0,519 | 0,519 | 0,310 |
| 26 | | 0,538 | 0,583 | 0,128 | -0,426 | 0,519 | 0,519 | 0,310 |
| 27 | time management | 0,942 | 0,854 | 0,949 | 0,463 | 0,926 | 0,926 | 0,843 |
| 28 | | 0,923 | 1,000 | 0,923 | 0,315 | 0,889 | 0,907 | 0,826 |
| 29 | integration management | 0,788 | 0,750 | 1,000 | 0,074 | 0,722 | 0,667 | 0,667 |
| 30 | | 0,788 | 0,708 | 0,538 | -0,056 | 0,463 | 0,648 | 0,515 |
| 31 | cost performance index | 0,692 | 0,583 | 0,462 | 0,519 | 0,519 | 0,667 | 0,573 |
| 32 | | 0,692 | 0,583 | 0,462 | 0,519 | 0,519 | 0,667 | 0,573 |

Tabell 5.10: Samlede normaliserte resultater

Den eneste konklusjonen man kan dra utifra de generelle resultatene er at de to tilnærmingene er forholdsvis like. Ved å ta utgangspunkt i de av spørringene hvor en ser en tydelig forskjell, det vil si 2, 3, 4, 8, 9, 10, 11 og 15, ser man likevel en liten tendens til at den manuelle ontologien gir bedre resultater ettersom den vinner i 4 av de 8 tilfellene, mens den semi-automatiske vinner i 2 av 8. De 2 resterende punktene hvor en ser en tydelig forskjell i resultatene, det vil si i spørring 11 og 15, finnes ikke spørringen i en av de to ontologiene. I disse tilfellene blir det bare utført vanlig tekstsøk, og som man ser vinner ontologistøttet søk mot vanlig tekstsøk for spørring 11, mens det taper for spørring 15. Dette kan tyde på at en utvidelse fra ontologien vil kunne ha både positiv og negativ effekt i søkesitasjoner hvor dokumentsamlingen er så liten som i dette tilfellet. Dette fordi en utvidelse vil gjøre spørringen mer spesifikk, og med få dokumenter i samlingen kan spørringen da bli for spesifikk i forhold til dokumentsamlingen.

To grunner for den begrensede variasjonen i resultatene kan nevnes:

- **Dokumentsamlingens struktur:** Som nevnt i kapittel 3 er PMBOK delt inn slik at hvert kunnskapsområde får ett kapittel. Problemet med dette er at de relaterte begrepene og synonymene gjerne kun finnes i de dokumentene som er klippet/limt sammen fra akkurat det kunnskapskapitlet, og vil sjelden eller aldri opptre alene i et dokument uten at hovedbegrepet også finnes der. På denne måten mister man mye av effekten til ontologien ved at man ikke også finner dokumenter hvor kun det relaterte begrepet eller synonymet opptrer.
- **Løvnoder i ontologiene:** Noen av spørringene opptrer bare i en av ontologiene (spørringene 11-16), og er samtidig løvnode i den ontologien hvor den opptrer. Det vil si at de ikke har noen klasser under seg i hierarkiet, og dermed får man bare utvidelser fra synonymer som stort sett kun opptrer i det samme dokumentet som spørringen selv. Dermed får man ingen effekt av ontologien og man ender opp med et resultatsett som vil være ekvivalent med det man hadde fått ved rent tekstsøk.

### Usikkerhetsmomenter

I dette avsnittet presenteres mulige usikkerhetsmomenter i forhold til resultatene:

- Usikkerhet rundt evaluatorene
- Lite domene og få dokumenter

**Evaluatorene:** Som nevnt i 5.4.1 er den oppfattede relevansen av dokumentene for brukerne en av nøkkelpunktene for resultatene i denne IR-testen. Brukerens motivasjon til å gjennomføre evalueringen spiller derfor en viktig rolle, og lav motivasjon vil derfor kunne virke inn på resultatene. I dette

tilfellet har brukerne bestått av tre medstudenter og tre stipendiater fra IDI. I testen måtte evaluatorene lese de 49 forskjellige dokumentene som ble returnert på bakgrunn av IR-testen, noe som forteller at omfanget av evalueringen var ganske stort. Når man i tillegg vet at alle evaluatorene var opptatt med henholdvis egne diplomoppgaver og konferanseforberedelser i evalueringsperioden ser en det som sannsynlig at dette kan ha påvirket resultatet da de naturlig nok hadde mest fokus på egne arbeider og oppgaver i perioden. Videre er det også en fare for at evaluatorene ikke har hatt noen egeninteresse av hverken dokument eller spørring og ser derfor ingen nytteverdien i å sette seg inn i dokumentene, noe som også vil kunne påvirke motivasjonen i negativ retning.

Et annet viktig punkt er at ingen av evaluatorene er domeneeksperter og har liten eller ingen kjennskap til domenet fra før. Dette vil kunne være en stor usikkerhetsfaktor ettersom enkelte av evaluatorene faktisk kan risikere å ikke forstår hva som ligger i spørringen, og som en konsekvens av dette heller ikke har noen formening om hvilke type dokumenter som burde returneres. En konsekvens av dette kan være at et relevant dokument blir gitt en dårlig poengscore fordi evaluatoren ikke får en beskrivelse av hva spørringen betyr og dermed antar at dokumentet er dårlig/irrelevant i forhold til spørringen.

Antallet evaluatorer er også lite. Med flere evaluatorer ville usikkerheten som vedrører nettopp evaluatorene jevnet seg ut og den gjennomsnittlige resultat-scoren ville blitt mer sikker.

**Lite domene, få dokumenter:** Ettersom dokumentsamlingen de søkes i er såpass liten er det fare for at bruken av ontologien i søket faktisk kan få negative konsekvenser ettersom utvidelsene kan gjøre spørringen for spesifikk i forhold til dokumentsamlingen. I tillegg ser en at enkelte av spørringene har svært få treff, noe som vil gjøre at man må sette verdien 0 på de plassene som ikke inneholder dokumenter og dette vil påvirke variasjonen i resultatet.

### Resultater i detalj

Siden det var vanskelig å dra noen konklusjoner utifra de overordnede resultatene prøver en i dette avsnittet å undersøke resultatene litt mer detaljert. Spesielt vil en se på:

- Resultatene sett i lys av spørring-kategoriseringen presentert i 5.4.2.
- Resultatene sett i lys av hvor spørringen befinner seg i ontologien.
- Resultatene sett i lys av utvidelsene fra ontologiene.

| Spørring | Nivå | Kat. | Utvidelser |
|---|---|---|---|
| 1 | 3 | C2 | 1 |
| 2 | 2 | C1 | 1(syn)+4 |
| 3 | 2 | C1 | 10 |
| 4 | 2 | C1 | 8 |
| 5 | 3 | C2 | 2 |
| 6 | 3* | C2 | 1(syn) |
| 7 | 4 | C2 | 3(syn)+2 |
| 8 | 3 | C1 | 1(syn)+2 |
| 9 | 3 | C1 | 1 |
| 10 | 2 | C1 | 2 |
| 11 | 3 | C2 | 2 |
| 12 | 4* | C2 | 2(syn) |
| 13 | 3* | C2 | 1(syn) |
| 14 | N/A | C1 | N/A |
| 15 | N/A | C1 | N/A |
| 16 | N/A | C2 | N/A |

Tabell 5.11: Manuell ontologi

| Spørring | Nivå | Kat. | Utvidelser |
|---|---|---|---|
| 1 | 2 | C2 | 2 |
| 2 | 1 | C1 | 7 |
| 3 | 1 | C1 | 2 |
| 4 | 1 | C1 | 8 |
| 5 | 2 | C2 | 1 |
| 6 | 2* | C2 | 1(syn) |
| 7 | 2* | C2 | 1(syn) |
| 8 | 2* | C1 | 1(syn) |
| 9 | 2 | C1 | 2 |
| 10 | 1 | C1 | 2 |
| 11 | N/A | C2 | N/A |
| 12 | N/A | C2 | N/A |
| 13 | N/A | C2 | N/A |
| 14 | 1 | C1 | 1 |
| 15 | 2 | C1 | 1 |
| 16 | 3* | C2 | 1(syn) |

Tabell 5.12: Semi-automatisk ontologi

Tabellene 5.11 og 5.12 viser de bakgrunnsdata for henholdsvis den manuelle og semi-automatiske ontologien som ligger til grunn for undersøkelsene nevnt over. Man har valgt å se spesielt på de spørringene hvor det var tydelige forskjeller i resultat, det vil si 2, 3, 4, 8, 9, 10, 11 og 15. De gule feltene markerer hvilken av tilnærmingene som fikk best resultat for den gitte spørringen. De grønne feltene markerer spørringene med best resultat for de tilfellene hvor bare en av ontologiene inneholdt spørringen. Spørringene hvor nivået er merket med * indikerer at spørringen er en løvnode i hierarkiet.

Ved å studere tabellene helt overfladisk ser man en tendens til at den manuelle ontologien har best resultat, noe som styrker inntrykket fra de overordnede resultatene presentert i begynnelsen av delkapitlet.

**Spørring-kategorisering:** Ved å studere spørring-kategoriene presentert i 5.4.2 og så sammenligne disse med tabellene 5.11 og 5.12 ser man en klar tendens til at det er i de mer generelle spørringene man får forskjeller i resultater. Hele 7 av de 8 spørringene er i kategori C1. En av hovedgrunnene til dette er at de generelle spørringene ofte vil ligge høyt oppe i hierarkiet i ontologien, og får dermed muligheten til å utvide med mer spesifiserte begreper og synonymer. Siden de to ontologiene er laget utifra to forskjellige tilnærminger er det naturlig at hierarkiet blir forskjellig, dermed blir utvidelsene forskjellig, som igjen fører til ulike resultat.

**Spørringens plassering i ontologien:** Et annet punkt en legger merke til ved å studere tabellene er at ingen av spørringene som opptrer som løvnoder

i ontologiene vinner i noen av spørringene. Bortsett fra i spørring 8, hvor løvnoden taper, er heller ingen av løvnodene med i noen av resultatene som utmerker seg med noen vesentlig forskjell. En av grunnene til dette er at løvnodene naturlig nok ikke er spesialisert noe videre og har da heller ingen utvidelser utover synonymene som er lagt til. Som nevnt tidligere er synonymer ofte bare nevnt en gang i PMBOK, og opptrer da sammen med hovedbegrepet og utvidelsen vil da heller ikke få noen effekt i forhold til dokumentene som returneres.

Ved å studere nivå-kolonnen ser man at at alle spørringene som utmerker seg resultatmessig ligger fra midten av hierarkiet og oppover, noe som er naturlig ettersom en i punktet over så at det var de generelle spørringene som utmerket seg.

**Utvidelser fra ontologien:** Ved å studere antall utvidelser i de forskjellige resultatene ser en også at det er en tendens til overvekt av seier til de spørringene som har flest utvidelser. Dette høres naturlig ut, men faktum er at for mange utvidelser kan gi dårlige resultatsett dersom kvalitetskravene om riktig avpasset detaljnivå og strenge hierarki ikke er oppfylt, men ettersom tendensen viser at de med flest utvidelser vinner viser det at ontologiene i seg selv holder god kvalitet.

### 5.4.4 Ontologibasert søk vs. ordinært fritekstsøk

Mot slutten av arbeidet ble det gjennomført et vanlig fritekstsøk for alle de 16 spørringene. Av de 16 spørringene hadde bare 4 forskjeller i returnert dokumentsett. Hovedgrunnen til at at en ser så liten forskjell må ligge i at dokumentsamlingen er svært begrenset, samt at strukturen i PMBOK er slik den er. Som nevnt tidligere i rapporten er kunnskapsområdene, og dermed også begrepene i ontologien, veldig konsentrert i PMBOK. Dette gjør at utvidelsene stort sett alltid vil forekomme i sammen med spørringen i et dokument, og effekten av ontologien blir dermed begrenset. På den annen side ser en at spørringen "stakeholder", som både har utvidelser i begge ontologiene og hvor utvidelsene forekommer spredt utover dokumentsamlingen, kun har et felles dokument med det samlede resultatsettet til de to ontologiene. Dette kan peke mot at utvidelser av spørring fra en ontologi kan ha en effekt på resultatsettet i forhold til vanlig fritekstsøk, men ettersom med bare en spørring å støtte seg til kan man ikke dra noen konklusjoner.

### 5.4.5 Oppsummering

Generelt sett er det for liten forskjell i resultatene til at en kan si noe sikkert om hvilken av de to ontologiene som fungerer best i søkesammenheng. En kan likevel

se tendenser til at den manuelle ontologien gir best resultat, noe som henger sammen med de funnene man gjorde i sammenligningsevalueringen fra 5.2 og kvalitetsevaluering fra 5.3 hvor man også så klare tendenser til at den manuelle ontologien holder en høyere kvalitet.

To av hovedgrunnene til at variasjonen i resultatene blir så liten er:

- På grunn av liten dokumentsamling, samt at strukturen på PMBOK er slik at spørring og utvidelser/synonymer svært ofte forekommer i samme dokument, er det svært sannsynlig at ontologien ikke vil gi den effekten man ønsker, noe som vil gi lite variasjon resultatene.
- Datagrunnlaget er noe tynt i forhold til det en kunne ønske seg. Med en større gruppe evaluatorer ville datagrunnlaget blitt større, og man kunne dermed sett eventuelle variasjoner tydeligere.

I tillegg til punktene over knytter det seg usikkerhet rundt evaluatorene siden de hadde liten, eller ingen domenetkjennskap fra før. Samtidig var jobben med å gjennomføre selve evalueringen relativt omfattende, noe som kan ha påvirket motivasjonen til å gjennomføre evalueringen i negativ retning.

Et positivt trekk er at man ser en tydelig tendens forskjell i resultater for generelle spørringer. Dette er spørringer som normalt ligger høyt oppe i hierarkiet, og hvor man derfor kan utnytte ontologiens struktur til å utvide spørringen, noe som igjen tyder på at ontologien har en effekt i søket.

# Kapittel 6

# Konklusjon og videre arbeid

I dette arbeidet har man sett på to ulike tilnærminger for å konstruere ontologier tenkt brukt i tekstsøk; manuell og semi-automatisk tilnærming. Den manuelle ontologien er konstruert ut fra en grundig gjennomgang og studie av dokumentet (PMBOK) og begrepene i dokumentet, mens man for den semi-automatiske ontologien har brukt et fraseekstraksjonssystem til å trekke ut et sett nøkkelfraser fra teksten som så er brukt til å bygge ontologien.

For å vurdere hvilken av metodene som fungere best har man gjennomført en grundig tredelt evaluering. Den første delen besto i å sammenligne de to ontologiene utfra deres oppbygning og struktur, mens man i den andre delen så de to ontologiene i lys av et sett kvalitetskrav for søkeontologier. I den siste delen ble det gjennomført en IR-test hvor ontologiene ble brukt i praktisk tekstsøk. Resultatene herifra ble så analysert og evaluert.

## 6.1 Konklusjon

Målene for oppgaven som ble presentert i innledningskapitlet var som følger:

Forskningsmål:
Studere mulige effekter av en manuelt utviklet ontologi på søk i en gitt dokumentsamling.

Resultatmål:
Utvikle to ontologier; en utfra en rent manuell tilnærming, og en semi-automatisk, basert på ekstraherte ord og fraser fra en dokumentsamling.

- Gjøre en sammenligning av de to ontologiene utfra deres oppbygning og struktur.
- Evaluere ontologiene utfra et sett kvalitetskrav for søkeontologier.
- Gjøre en evaluering av ontologiene med bakgrunn i resultater fra tester i praktisk søk.

Effekten av den manuelt utviklede ontologien i en søkesituasjon har vært vanskelig å måle ettersom dokumentsamlingen og datagrunnlaget fra evalueringen blir noe knapp, og variasjonen i resultatene blir derfor liten. Man ser likevel tendenser til at bruk av utvidelser fra ontologien har en effekt i søket, men tendensene er for svake til at man kan trekke noen endelig konklusjon.

Som nevnt i innledningen til kapitlet er det i arbeidet utviklet to ontologier utfra to forskjellige tilnærminger; manuell tilnærming og semi-automatisk tilnærming. Evalueringen av disse viser en klar tendens til at den manuelt utvidlede ontologien generelt sett har høyere kvalitet, og dermed også har høyere potensiale som søkeontologi, noe tendensene i IR-evalueringen bekrefter.

## 6.2 Videre arbeid

For å kunne dra sikrere konklusjoner rundt ontologienes effekt i en søkesituasjon er det nødvendig å gjøre en ny IR-evaluering hvor man har en større dokumentsamling tilgjengelig. For at en slik evaluering skal bli best mulig er det nødvendig at evaluatorene kjenner domenet fra før, er motiverte og har egeninteresse av å gjennomføre evalueringen.

Utformingen av selve ontologien er også et punkt det kan jobbes videre med. Ved å legge til flere begreper/synonymer kan man få dypere hierarkier og dermed flere utvidelsesmuligheter. Man kan også se på muligheten til å legge flere OWL-elementer slik som attributter og således få muligheten til å koble sammen begreper som forekommer på ulike steder i hierarkiet.

IR-systemet som er brukt i dette arbeidet simulerer kun ontologiens tilstedeværelse, og et system hvor ontologien er inkorporert som en del av systemet er derfor ønskelig. Ved å gjøre dette får man muligheten til å bruke ontologien som en visuell hjelper for brukeren ved for eksempel å gjøre hierarkiet synlig for brukeren.

# Tillegg A

# Utvidelser av spørring

Tabell A.1 og A.2 viser utvidelsene av spørringene gjort i IR-testen på bakgrunn av innholdet i den manuelle- og semi-automatiske ontologien.

| Spørring | Synonymer | Relaterte begreper |
|---|---|---|
| cost estimating | - | resource planning |
| human resource | HR | organizational planning, staff, staff acquisition, team development |
| stakeholder | - | buyer, customer, external stakeholder, internal stakeholder, project staff, seller (vendor, supplier, subcontractor), sponsor |
| procurement | - | bidder conference (vendor conference, contractor conference, pre-bid conference), contract, procurement planning, solicitation, source selection |
| risk response | - | risk response plan (risk register) |
| work breakdown structure | WBS | - |
| earned value | budget cost of work performed, BCWP, EV | actual cost, planned value |
| earned value management | EVM | control account plan, earned value |
| communications management | - | communications management plan |
| performance | - | performance measurement, performance reporting |
| planning process | - | core process, facilitating process |
| control account plan | cost account plan, CAP | - |
| project management information system | PMIS | - |
| time management | - | - |
| integration management | - | - |
| cost performance index | - | - |

Tabell A.1: Utvidelser fra den manuelle ontologien

| **Spørring** | **Synonymer** | **Relaterte begreper** |
|---|---|---|
| cost estimating | - | estimate at completion, EAC |
| human resource | - | human resource management, organization chart, role, staff, staff assignment, team competencies, team development |
| stakeholder | - | buyer, seller |
| procurement | - | contract administration, contract closeout, procurement item, procurement management, procurement management plan, procurement planning, proposal, solicitation planning |
| risk response | - | risk response plan |
| work breakdown structure | breakdown structure | - |
| earned value | EV | - |
| earned value management | EVM | - |
| communications management | - | communications management plan, communications management figure |
| performance | - | performance reporting, team performance |
| planning process | - | - |
| control account plan | - | - |
| project management information system | - | - |
| time management | - | activity schedule development |
| integration management | - | integrated change control |
| cost performance index | CPI | - |

Tabell A.2: Utvidelser fra den semi-automatiske ontologien

# Tillegg B

# Resultater IR-test

Resultatene i venstre kolonne viser resultater etter test med den manuelt konstruerte ontologien, mens kolonnen til høyre viser resultat fra den semi-automatisk konstruerte ontologien. Tallene i parantes angir det totale antall treff som spørringen gav.

**Cost estimating** (21)
Fikk treff i dokument: dok29.txt
Score: 0.4444782
Fikk treff i dokument: dok30.txt
Score: 0.37966555
Fikk treff i dokument: dok31.txt
Score: 0.27218032
Fikk treff i dokument: dok32.txt
Score: 0.20217048
Fikk treff i dokument: dok74.txt
Score: 0.19112788

**Cost estimating** (18)
Fikk treff i dokument: dok33.txt
Score: 0.26399368
Fikk treff i dokument: dok31.txt
Score: 0.10614123
Fikk treff i dokument: dok32.txt
Score: 0.07883973
Fikk treff i dokument: dok29.txt
Score: 0.06898476
Fikk treff i dokument: dok51.txt
Score: 0.04927483

**Human resource** (24)
Fikk treff i dokument: dok38.txt
Score: 0.51213
Fikk treff i dokument: dok40.txt
Score: 0.30372477
Fikk treff i dokument: dok3.txt
Score: 0.20443928
Fikk treff i dokument: dok12.txt
Score: 0.16355143
Fikk treff i dokument: dok39.txt
Score: 0.11349804

**Human resource** (27)
Fikk treff i dokument: dok38.txt
Score: 0.4523837
Fikk treff i dokument: dok40.txt
Score: 0.42932746
Fikk treff i dokument: dok41.txt
Score: 0.33423278
Fikk treff i dokument: dok39.txt
Score: 0.26313818
Fikk treff i dokument: dok3.txt
Score: 0.10953901

**Stakeholder** (52)
Fikk treff i dokument: dok54.txt
Score: 0.36193827
Fikk treff i dokument: dok6.txt
Score: 0.07320959
Fikk treff i dokument: dok57.txt
Score: 0.067812376
Fikk treff i dokument: dok48.txt
Score: 0.06397398
Fikk treff i dokument: dok58.txt
Score: 0.058136493

**Stakeholder** (44)
Fikk treff i dokument: dok54.txt
Score: 0.64371896
Fikk treff i dokument: dok19.txt
Score: 0.23949763
Fikk treff i dokument: dok59.txt
Score: 0.22777498
Fikk treff i dokument: dok55.txt
Score: 0.2068675
Fikk treff i dokument: dok58.txt
Score: 0.18197069

**Procurement** (31)
Fikk treff i dokument: dok57.txt
Score: 0.6211812
Fikk treff i dokument: dok54.txt
Score: 0.14683169
Fikk treff i dokument: dok56.txt
Score: 0.13636094
Fikk treff i dokument: dok55.txt
Score: 0.08865435
Fikk treff i dokument: dok58.txt
Score: 0.08180939

**Procurement** (23)
Fikk treff i dokument: dok55.txt
Score: 0.53088874
Fikk treff i dokument: dok54.txt
Score: 0.5284322
Fikk treff i dokument: dok60.txt
Score: 0.5247762
Fikk treff i dokument: dok56.txt
Score: 0.512507
Fikk treff i dokument: dok3.txt
Score: 0.21734451

**Risk response** (12)
Fikk treff i dokument: dok52.txt
Score: 0.63071316
Fikk treff i dokument: dok53.txt
Score: 0.34381336
Fikk treff i dokument: dok48.txt
Score: 0.2358942
Fikk treff i dokument: dok47.txt
Score: 0.15889251
Fikk treff i dokument: dok51.txt
Score: 0.13553007

**Risk response** (12)
Fikk treff i dokument: dok53.txt
Score: 0.69670093
Fikk treff i dokument: dok52.txt
Score: 0.6646826
Fikk treff i dokument: dok48.txt
Score: 0.47801432
Fikk treff i dokument: dok47.txt
Score: 0.32197866
Fikk treff i dokument: dok51.txt
Score: 0.27463713

**Work breakdown structure** (12)
Fikk treff i dokument: dok24.txt
Score: 0.17903323
Fikk treff i dokument: dok30.txt
Score: 0.17054337
Fikk treff i dokument: dok22.txt
Score: 0.13781986
Fikk treff i dokument: dok32.txt
Score: 0.13781986
Fikk treff i dokument: dok4.txt
Score: 0.10336489

**Work breakdown structure** (12)
Fikk treff i dokument: dok24.txt
Score: 0.60876465
Fikk treff i dokument: dok30.txt
Score: 0.5798967
Fikk treff i dokument: dok22.txt
Score: 0.46862727
Fikk treff i dokument: dok32.txt
Score: 0.46862727
Fikk treff i dokument: dok21.txt
Score: 0.42221123

**Earned value** (9)
Fikk treff i dokument: dok45.txt
Score: 0.35884264
Fikk treff i dokument: dok33.txt
Score: 0.32135454
Fikk treff i dokument: dok14.txt
Score: 0.079454556
Fikk treff i dokument: dok53.txt
Score: 0.026426157
Fikk treff i dokument: dok15.txt
Score: 0.010570463

**Earned value** (7)
Fikk treff i dokument: dok45.txt
Score: 0.6798363
Fikk treff i dokument: dok33.txt
Score: 0.59522015
Fikk treff i dokument: dok14.txt
Score: 0.5295706
Fikk treff i dokument: dok53.txt
Score: 0.17613232
Fikk treff i dokument: dok15.txt
Score: 0.07045293

**Earned value management** (6)
Fikk treff i dokument: dok33.txt
Score: 0.6566969
Fikk treff i dokument: dok14.txt
Score: 0.4738133
Fikk treff i dokument: dok15.txt
Score: 0.22845747
Fikk treff i dokument: dok21.txt
Score: 0.10601918
Fikk treff i dokument: dok45.txt
Score: 0.04263013

**Earned value management** (4)
Fikk treff i dokument: dok14.txt
Score: 0.5267863
Fikk treff i dokument: dok33.txt
Score: 0.44412863
Fikk treff i dokument: dok15.txt
Score: 0.28378522
Fikk treff i dokument: dok21.txt
Score: 0.08632946

**Communication management** (9)
Fikk treff i dokument: dok44.txt
Score: 0.5973053
Fikk treff i dokument: dok43.txt
Score: 0.50931156
Fikk treff i dokument: dok45.txt
Score: 0.23903792
Fikk treff i dokument: dok42.txt
Score: 0.16201545
Fikk treff i dokument: dok15.txt
Score: 0.16121829

**Communication management** (9)
Fikk treff i dokument: dok44.txt
Score: 0.32073104
Fikk treff i dokument: dok45.txt
Score: 0.30016577
Fikk treff i dokument: dok42.txt
Score: 0.2888022
Fikk treff i dokument: dok43.txt
Score: 0.2734816
Fikk treff i dokument: dok15.txt
Score: 0.08656831

**Performance** (57)
Fikk treff i dokument: dok45.txt
Score: 0.50088036
Fikk treff i dokument: dok46.txt
Score: 0.42222962
Fikk treff i dokument: dok17.txt
Score: 0.40169135
Fikk treff i dokument: dok23.txt
Score: 0.36376393
Fikk treff i dokument: dok33.txt
Score: 0.3289861

**Performance** (57)
Fikk treff i dokument: dok41.txt
Score: 0.32408127
Fikk treff i dokument: dok45.txt
Score: 0.29545352
Fikk treff i dokument: dok59.txt
Score: 0.21006243
Fikk treff i dokument: dok46.txt
Score: 0.18229657
Fikk treff i dokument: dok17.txt
Score: 0.17691915

**Planning process** (19)
Fikk treff i dokument: dok12.txt
Score: 0.47203755
Fikk treff i dokument: dok35.txt
Score: 0.08139583
Fikk treff i dokument: dok13.txt
Score: 0.03023014
Fikk treff i dokument: dok74.txt
Score: 0.02671992
Fikk treff i dokument: dok11.txt
Score: 0.026451372

**Planning process** (19)
Fikk treff i dokument: dok13.txt
Score: 0.19774827
Fikk treff i dokument: dok12.txt
Score: 0.18497656
Fikk treff i dokument: dok74.txt
Score: 0.17478643
Fikk treff i dokument: dok11.txt
Score: 0.17302974
Fikk treff i dokument: dok30.txt
Score: 0.17302974

**Control account plan** (3)
Fikk treff i dokument: dok33.txt
Score: 0.1329076
Fikk treff i dokument: dok15.txt
Score: 0.106326066
Fikk treff i dokument: dok74.txt
Score: 0.054062154

**Control account plan** (2)
Fikk treff i dokument: dok33.txt
Score: 0.20374663
Fikk treff i dokument: dok15.txt
Score: 0.1629973

**Project management information system** (3)
Fikk treff i dokument: dok16.txt
Score: 0.08733705
Fikk treff i dokument: dok17.txt
Score: 0.08733705
Fikk treff i dokument: dok15.txt
Score: 0.0582247

**Project management information system** (3)
Fikk treff i dokument: dok16.txt
Score: 0.24797581
Fikk treff i dokument: dok17.txt
Score: 0.24797581
Fikk treff i dokument: dok15.txt
Score: 0.16531721

**Time management** (7)
Fikk treff i dokument: dok24.txt
Score: 0.2613926
Fikk treff i dokument: dok28.txt
Score: 0.13776599
Fikk treff i dokument: dok27.txt
Score: 0.11810973
Fikk treff i dokument: dok3.txt
Score: 0.09741527
Fikk treff i dokument: dok8.txt
Score: 0.09741527

**Time management** (46)
Fikk treff i dokument: dok24.txt
Score: 0.43833408
Fikk treff i dokument: dok27.txt
Score: 0.31734028
Fikk treff i dokument: dok3.txt
Score: 0.18087749
Fikk treff i dokument: dok26.txt
Score: 0.112725824
Fikk treff i dokument: dok25.txt
Score: 0.10525596

**Integration management** (4)
Fikk treff i dokument: dok14.txt
Score: 0.461113
Fikk treff i dokument: dok15.txt
Score: 0.20621601
Fikk treff i dokument: dok17.txt
Score: 0.15466201
Fikk treff i dokument: dok3.txt
Score: 0.128885

**Integration management** (11)
Fikk treff i dokument: dok17.txt
Score: 0.57883865
Fikk treff i dokument: dok14.txt
Score: 0.56587946
Fikk treff i dokument: dok3.txt
Score: 0.24767222
Fikk treff i dokument: dok59.txt
Score: 0.13738894
Fikk treff i dokument: dok36.txt
Score: 0.12953155

**Cost performance index** (2)
Fikk treff i dokument: dok45.txt
Score: 0.36245406
Fikk treff i dokument: dok33.txt
Score: 0.2562937

**Cost performance index** (2)
Fikk treff i dokument: dok45.txt
Score: 0.4838072
Fikk treff i dokument: dok33.txt
Score: 0.39459574

# Tillegg C

# Resultat brukerevaluering

Tabell C.1 til og med C.6 viser poengene de ulike evaluatorene gav til returnerte dokumenter fra IR-testen. Poenggivningen følger skjemaet skissert i tabell 5.8. Resultatene er sortert slik at kolonne 1 representerer poengsummen til det høyest rangerte dokumentet, kolonne 2 representerer poengsummen til det nest-høyest rangerte dokumentet osv. I selve testen var dokumentene sortert tilfeldig for å unngå at rangeringen påvirket evaluatorens poengsetting. Blanke felter indikerer at det ble returnert mindre enn 5 dokumenter.

Tabell C.7 og figur C.1 viser dataene fra brukerevalueringen før normalieringen presentert i 5.4.3 ble utført. Kolonnene i C.7 representerer resultatene for hver av evaluatorene.

| # | Query | Result | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | cost estimating | 2 | 1 | 2 | 1 | 0 |
| 2 | | 0 | 2 | 1 | 2 | 0 |
| 3 | human resource | 2 | 2 | 0 | 0 | 2 |
| 4 | | 2 | 2 | 2 | 2 | 0 |
| 5 | stakeholder | 0 | 2 | 1 | 0 | 0 |
| 6 | | 0 | 0 | 1 | 0 | 0 |
| 7 | procurement | 1 | 2 | 1 | 2 | 1 |
| 8 | | 2 | 2 | 0 | 1 | 0 |
| 9 | risk response | 2 | 1 | 1 | 1 | 1 |
| 10 | | 1 | 2 | 1 | 1 | 1 |
| 11 | work breakdown structure | 1 | 1 | 1 | 1 | 0 |
| 12 | | 1 | 1 | 1 | 1 | 2 |
| 13 | earned value | 2 | 2 | 0 | 1 | 1 |
| 14 | | 2 | 2 | 0 | 1 | 1 |
| 15 | earned value management | 2 | 0 | 1 | 0 | 2 |
| 16 | | 0 | 2 | 1 | 0 | |
| 17 | communication management | 2 | 2 | 2 | 2 | 0 |
| 18 | | 2 | 2 | 2 | 2 | 0 |
| 19 | performance | 2 | 1 | 1 | 1 | 1 |
| 20 | | 1 | 2 | 0 | 1 | 1 |
| 21 | planning process | 2 | 0 | 1 | 0 | 0 |
| 22 | | 1 | 2 | 0 | 0 | 0 |
| 23 | control account plan | 1 | 1 | 0 | | |
| 24 | | 1 | 1 | | | |
| 25 | project management information system | 1 | 1 | 2 | | |
| 26 | | 1 | 1 | 2 | | |
| 27 | time management | 2 | 2 | 2 | 1 | 0 |
| 28 | | 2 | 2 | 1 | 2 | 1 |
| 29 | integration management | 2 | 1 | 2 | 1 | |
| 30 | | 2 | 2 | 1 | 0 | 0 |
| 31 | cost performance index | 2 | 2 | | | |
| 32 | | 2 | 2 | | | |

Tabell C.1: Resultater evaluator 1

| # | Query | Result | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | cost estimating | 1 | 1 | 2 | 1 | -1 |
| 2 | | 1 | 2 | 1 | 1 | 1 |
| 3 | human resource | 2 | 2 | 1 | 0 | 2 |
| 4 | | 2 | 2 | 1 | 2 | 1 |
| 5 | stakeholder | 1 | 2 | 1 | 1 | 1 |
| 6 | | 1 | 1 | 1 | 0 | 1 |
| 7 | procurement | 1 | 2 | 2 | 2 | 2 |
| 8 | | 2 | 2 | 1 | 2 | 1 |
| 9 | risk response | 2 | 1 | 1 | 1 | 1 |
| 10 | | 1 | 2 | 1 | 1 | 1 |
| 11 | work breakdown structure | 2 | 1 | 1 | 1 | 1 |
| 12 | | 2 | 1 | 1 | 1 | 2 |
| 13 | earned value | 2 | 2 | 1 | 1 | 1 |
| 14 | | 2 | 2 | 1 | 1 | 1 |
| 15 | earned value management | 2 | 1 | 1 | 1 | 1 |
| 16 | | 1 | 2 | 1 | 1 | |
| 17 | communication management | 2 | 2 | 1 | 2 | 1 |
| 18 | | 2 | 1 | 2 | 2 | 1 |
| 19 | performance | 2 | 1 | 1 | 1 | 2 |
| 20 | | 1 | 2 | 1 | 1 | 1 |
| 21 | planning process | 2 | 1 | 1 | -1 | 2 |
| 22 | | 1 | 2 | -1 | 2 | 1 |
| 23 | control account plan | 2 | 1 | -1 | | |
| 24 | | 2 | 1 | | | |
| 25 | project management information system | 1 | 1 | 2 | | |
| 26 | | 1 | 1 | 2 | | |
| 27 | time management | 2 | 1 | 2 | 1 | 0 |
| 28 | | 2 | 2 | 1 | 2 | 1 |
| 29 | integration management | 2 | 1 | 1 | 1 | |
| 30 | | 1 | 2 | 1 | 1 | 0 |
| 31 | cost performance index | 2 | 1 | | | |
| 32 | | 2 | 1 | | | |

Tabell C.2: Resultater evaluator 2

| # | Query | Result | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | cost estimating | 1 | 0 | 2 | 1 | 0 |
| 2 | | 1 | 1 | 0 | 1 | -1 |
| 3 | human resource | 2 | 1 | 0 | 0 | 1 |
| 4 | | 2 | 1 | 1 | 1 | 0 |
| 5 | stakeholder | 0 | 2 | -1 | 0 | 0 |
| 6 | | 0 | 0 | 1 | 0 | 0 |
| 7 | procurement | 1 | 2 | 1 | 2 | 0 |
| 8 | | 2 | 2 | 0 | 1 | 0 |
| 9 | risk response | 2 | 1 | 1 | 1 | 1 |
| 10 | | 1 | 2 | 1 | 1 | 1 |
| 11 | work breakdown structure | 1 | 1 | 1 | 0 | 0 |
| 12 | | 1 | 1 | 1 | 0 | 1 |
| 13 | earned value | 1 | 1 | 0 | 0 | 1 |
| 14 | | 1 | 1 | 0 | 0 | 1 |
| 15 | earned value management | 1 | 1 | 1 | 0 | 1 |
| 16 | | 1 | 1 | 1 | 0 | |
| 17 | communication management | 1 | 2 | 0 | 2 | 0 |
| 18 | | 1 | 0 | 2 | 2 | 0 |
| 19 | performance | 2 | 0 | 0 | 0 | 1 |
| 20 | | 0 | 2 | 0 | 0 | 0 |
| 21 | planning process | 0 | 1 | 1 | 1 | 0 |
| 22 | | 1 | 0 | 1 | 0 | 1 |
| 23 | control account plan | 0 | 0 | 0 | | |
| 24 | | 0 | 0 | | | |
| 25 | project management information system | 0 | 0 | 1 | | |
| 26 | | 0 | 0 | 1 | | |
| 27 | time management | 2 | 1 | 1 | 1 | 1 |
| 28 | | 2 | 1 | 1 | 1 | 0 |
| 29 | integration management | 2 | 2 | 0 | 1 | |
| 30 | | 0 | 2 | 1 | 0 | 0 |
| 31 | cost performance index | 1 | 1 | | | |
| 32 | | 1 | 1 | | | |

Tabell C.3: Resultater evaluator 3

| # | Query | Result 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | cost estimating | 1 | -1 | 2 | 2 | -1 |
| 2 | | 2 | 1 | 2 | 1 | 2 |
| 3 | human resource | 2 | 1 | -1 | 0 | 1 |
| 4 | | 2 | 1 | 0 | 1 | -1 |
| 5 | stakeholder | -1 | 2 | -1 | 1 | -1 |
| 6 | | -1 | -1 | -1 | -1 | -1 |
| 7 | procurement | 1 | 2 | 2 | 2 | 1 |
| 8 | | 2 | 2 | 1 | 2 | 0 |
| 9 | risk response | 2 | 2 | 1 | 0 | 0 |
| 10 | | 2 | 2 | 1 | 0 | 0 |
| 11 | work breakdown structure | 2 | 1 | -1 | -1 | -1 |
| 12 | | 2 | 1 | -1 | -1 | 2 |
| 13 | earned value | 2 | 2 | 0 | 1 | -1 |
| 14 | | 2 | 2 | 0 | 1 | -1 |
| 15 | earned value management | 2 | 0 | -1 | -1 | 2 |
| 16 | | 0 | 2 | -1 | -1 | |
| 17 | communication management | 1 | 2 | -1 | 2 | -1 |
| 18 | | 1 | -1 | 2 | 2 | -1 |
| 19 | performance | 2 | 1 | 2 | 2 | 2 |
| 20 | | 2 | 2 | 2 | 2 | 2 |
| 21 | planning process | 2 | 2 | 1 | -1 | 1 |
| 22 | | 1 | 2 | -1 | 1 | 1 |
| 23 | control account plan | -1 | 0 | -1 | | |
| 24 | | -1 | 0 | | | |
| 25 | project management information system | -1 | -1 | -1 | | |
| 26 | | -1 | -1 | -1 | | |
| 27 | time management | 2 | 1 | 0 | -1 | 0 |
| 28 | | 2 | 0 | -1 | 1 | -1 |
| 29 | integration management | 2 | -1 | -1 | -1 | |
| 30 | | -1 | 2 | -1 | -1 | -1 |
| 31 | cost performance index | 2 | 1 | | | |
| 32 | | 2 | 1 | | | |

Tabell C.4: Resultater evaluator 4

| # | Query | Result | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | cost estimating | 2 | 2 | 2 | 2 | 1 |
| 2 | | 1 | 2 | 2 | 2 | 2 |
| 3 | human resource | 2 | 2 | 1 | 1 | 2 |
| 4 | | 2 | 2 | 2 | 2 | 1 |
| 5 | stakeholder | 1 | 2 | 0 | 2 | 0 |
| 6 | | 0 | 1 | 1 | 0 | 1 |
| 7 | procurement | 1 | 2 | 1 | 2 | 1 |
| 8 | | 2 | 2 | 2 | 1 | 1 |
| 9 | risk response | 2 | 2 | 2 | 1 | 2 |
| 10 | | 2 | 2 | 2 | 1 | 2 |
| 11 | work breakdown structure | 2 | 2 | 2 | 2 | 1 |
| 12 | | 2 | 2 | 2 | 2 | 2 |
| 13 | earned value | 2 | 2 | 1 | 2 | 1 |
| 14 | | 2 | 2 | 1 | 2 | 1 |
| 15 | earned value management | 2 | 1 | 1 | 1 | 2 |
| 16 | | 1 | 2 | 1 | 1 | |
| 17 | communication management | 2 | 2 | 1 | 2 | 2 |
| 18 | | 2 | 1 | 2 | 2 | 2 |
| 19 | performance | 2 | 2 | 2 | 2 | 2 |
| 20 | | 2 | 2 | 2 | 2 | 2 |
| 21 | planning process | 2 | 2 | 2 | 1 | 1 |
| 22 | | 2 | 2 | 1 | 1 | 2 |
| 23 | control account plan | 2 | 2 | 1 | | |
| 24 | | 2 | 2 | | | |
| 25 | project management information system | 1 | 1 | 2 | | |
| 26 | | 1 | 1 | 2 | | |
| 27 | time management | 2 | 2 | 2 | 1 | 1 |
| 28 | | 2 | 2 | 1 | 2 | 1 |
| 29 | integration management | 2 | 2 | 0 | 1 | |
| 30 | | 0 | 2 | 1 | 1 | 1 |
| 31 | cost performance index | 2 | 1 | | | |
| 32 | | 2 | 1 | | | |

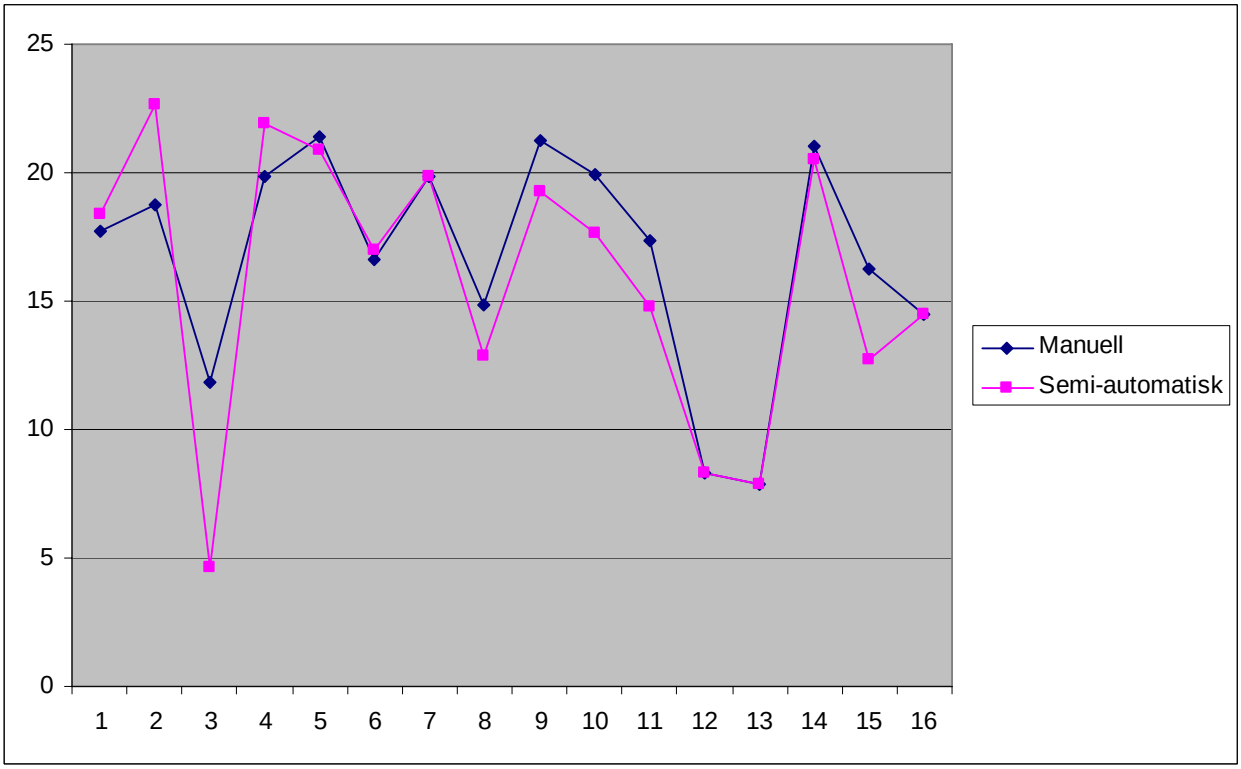
Tabell C.5: Resultater evaluator 5

| # | Query | Result | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 1 | cost estimating | 2 | 2 | 2 | 1 | 0 |
| 2 | | 2 | 2 | 2 | 2 | 0 |
| 3 | human resource | 2 | 2 | 1 | 1 | 2 |
| 4 | | 2 | 2 | 2 | 2 | 1 |
| 5 | stakeholder | 1 | 2 | 2 | 1 | 1 |
| 6 | | 1 | 1 | 2 | 2 | 1 |
| 7 | procurement | 1 | 2 | 1 | 2 | 1 |
| 8 | | 2 | 2 | 0 | 1 | 1 |
| 9 | risk response | 2 | 2 | 2 | 2 | 2 |
| 10 | | 2 | 2 | 2 | 2 | 2 |
| 11 | work breakdown structure | 2 | 1 | 2 | 1 | 0 |
| 12 | | 2 | 1 | 1 | 1 | 2 |
| 13 | earned value | 2 | 2 | 1 | 2 | 1 |
| 14 | | 2 | 2 | 1 | 2 | 1 |
| 15 | earned value management | 2 | 1 | 1 | 1 | 2 |
| 16 | | 1 | 2 | 1 | 1 | |
| 17 | communication management | 2 | 2 | 1 | 2 | 1 |
| 18 | | 2 | 1 | 2 | 2 | 1 |
| 19 | performance | 2 | 1 | 2 | 1 | 2 |
| 20 | | 0 | 2 | 1 | 0 | 2 |
| 21 | planning process | 2 | 2 | 1 | 1 | 1 |
| 22 | | 1 | 2 | 1 | 1 | 2 |
| 23 | control account plan | 2 | 1 | 1 | | |
| 24 | | 2 | 1 | | | |
| 25 | project management information system | 1 | 1 | 2 | | |
| 26 | | 1 | 1 | 2 | | |
| 27 | time management | 2 | 2 | 2 | 1 | 1 |
| 28 | | 2 | 2 | 1 | 2 | 2 |
| 29 | integration management | 2 | 1 | 1 | 1 | |
| 30 | | 1 | 2 | 1 | 1 | 1 |
| 31 | cost performance index | 2 | 2 | | | |
| 32 | | 2 | 2 | | | |

Tabell C.6: Resultater evaluator 6

| # | Query | Results | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | Snitt |
| 1 | cost estimating | 20,5 | 15 | 11,5 | 8,5 | 26,5 | 24,5 | 17,75 |
| 2 | | 13,5 | 17,5 | 10 | 21,5 | 22 | 26 | 18,42 |
| 3 | human resource | 18,5 | 21,5 | 14,5 | 12 | 23 | 23 | 18,75 |
| 4 | | 26 | 24 | 18 | 15 | 26,5 | 26,5 | 22,67 |
| 5 | stakeholder | 10,5 | 17,5 | 5,5 | 1,5 | 16 | 20 | 11,83 |
| 6 | | 2,5 | 12 | 2,5 | -13,5 | 7 | 17,5 | 4,67 |
| 7 | procurement | 19 | 22 | 18,5 | 21,5 | 19 | 19 | 19,83 |
| 8 | | 19,5 | 24 | 19,5 | 23,5 | 25 | 20 | 21,92 |
| 9 | risk response | 18,5 | 18,5 | 18,5 | 20,5 | 25,5 | 27 | 21,42 |
| 10 | | 17,5 | 17,5 | 17,5 | 20,5 | 25,5 | 27 | 20,92 |
| 11 | work breakdown structure | 13 | 18,5 | 11,5 | 9,5 | 26,5 | 20,5 | 16,58 |
| 12 | | 14 | 19 | 12 | 11 | 27 | 19 | 17,00 |
| 13 | earned value | 20 | 22,5 | 9,5 | 19 | 24 | 24 | 19,83 |
| 14 | | 20 | 22,5 | 9,5 | 19 | 24 | 24 | 19,83 |
| 15 | earned value management | 13,5 | 18,5 | 12 | 7 | 19 | 19 | 14,83 |
| 16 | | 10,5 | 17 | 11,5 | 4 | 17 | 17 | 12,83 |
| 17 | communication management | 26 | 24 | 16 | 13 | 24,5 | 24 | 21,25 |
| 18 | | 26 | 22,5 | 13 | 8,5 | 23 | 22,5 | 19,25 |
| 19 | performance | 18,5 | 19 | 10,5 | 23 | 27 | 21,5 | 19,92 |
| 20 | | 15 | 17,5 | 8 | 27 | 27 | 11,5 | 17,67 |
| 21 | planning process | 12,5 | 16 | 8 | 20 | 25 | 22,5 | 17,33 |
| 22 | | 13 | 14 | 8 | 12,5 | 23 | 18 | 14,75 |
| 23 | control account plan | 9 | 11,5 | 0 | -7,5 | 20,5 | 16,5 | 8,33 |
| 24 | | 9 | 14 | 0 | -5 | 18 | 14 | 8,33 |
| 25 | project management information system | 14 | 14 | 2,5 | -11,5 | 14 | 14 | 7,83 |
| 26 | | 14 | 14 | 2,5 | -11,5 | 14 | 14 | 7,83 |
| 27 | time management | 24,5 | 20,5 | 18,5 | 12,5 | 25 | 25 | 21,00 |
| 28 | | 24 | 24 | 18 | 8,5 | 24 | 24,5 | 20,50 |
| 29 | integration management | 20,5 | 18 | 19,5 | 2 | 19,5 | 18 | 16,25 |
| 30 | | 20,5 | 17 | 10,5 | -1,5 | 12,5 | 17,5 | 12,75 |
| 31 | cost performance index | 18 | 14 | 9 | 14 | 14 | 18 | 14,50 |
| 32 | | 18 | 14 | 9 | 14 | 14 | 18 | 14,50 |

Tabell C.7: Samlede resultater




Figur C.1: Graf over snittet til resultatene

# Tillegg D

# Ekstraherte ord/fraser

Dette tillegget viser hvilke begreper og fraser som ble trukket ut av fraseekstraksonssystemet for hvert av de 12 kapitlene i PMBOK.

```
************************
chapter1.txt_tagged.txt
************************
01: project management (0.0289)
02: elaboration (0.0165)
03: operations (0.0152)
04: profession (0.0142)
05: program (0.011)
06: thousands (0.0106)
07: other management disciplines (0.0106)
08: does (0.0106)
09: processing units (0.0106)
10: program management (0.0106)
11: income residents (0.0106)
12: related endeavors (0.0106)
13: objective (0.0087)
14: facility (0.0082)
15: development project (0.0082)

************************
chapter2.txt_tagged.txt
************************
01: project management context (0.0188)
02: project life cycle (0.0119)
03: feasibility study (0.0118)
04: end (0.0099)
```

```
05: standards (0.0099)
06: vice president (0.0091)
07: product life cycle (0.0091)
08: regulations (0.0087)
09: organiza¡tion (0.0081)
10: contract terms (0.0070)
11: project office (0.0070)
12: government agencies (0.0070)
13: style (0.0062)
14: vision (0.0062)
15: manufacturing (0.0062)

************************
chapter3.txt_tagged.txt
************************
01: core processes (0.024)
02: project management process groups (0.024)
03: project management knowledge areas (0.0185)
04: quality standards (0.0185)
05: mapping (0.016)
06: facilitating processes (0.012)
07: management processes figure (0.012)
08: plan forrisk management (0.012)
09: project management processes figure (0.012)
10: work periodsthat (0.012)
11: list procurement planning (0.012)
12: com¡munications plan (0.012)
13: project level (0.012)
14: executing processes (0.012)
15: group skills (0.012)

************************
chapter4.txt_tagged.txt
************************
01: change control system (0.0162)
02: performance measurement baselines (0.0142)
03: project plan execution (0.01)
04: project baseline (0.0095)
05: subsidiary management plans (0.0095)
06: management control plans (0.0095)
07: project control (0.0095)
08: 5project management information system (0.0095)
09: integration management (0.0095)
```

```
10: work authorization system (0.0095)
11: integrated change control (0.0087)
12: project execution (0.0087)
13: scope management plan (0.0073)
14: evm (0.0073)
15: pmis (0.0063)

************************
chapter5.txt_tagged.txt
************************
01: scope management (0.0243)
02: wbs (0.0161)
03: business need (0.0138)
04: constituent components (0.0138)
05: product description (0.0134)
06: scope change control (0.0118)
07: scope management plan (0.0107)
08: scope changes (0.0101)
09: project charter (0.0101)
10: scope definition (0.0094)
11: product scope (0.0080)
12: scope planning (0.0080)
13: breakdown structure (0.0080)
14: work results (0.0080)
15: scope statement (0.0078)

************************
chapter6.txt_tagged.txt
************************
01: activity list (0.0222)
02: project network diagrams (0.0183)
03: activity durations (0.0141)
04: dependencies (0.0135)
05: work periods (0.013)
06: schedule development (0.0128)
07: time management (0.0121)
08: days (0.0104)
09: review technique (0.0104)
10: software project (0.0104)
11: activity sequencing (0.0102)
12: project duration (0.0101)
13: estimates (0.0101)
14: activity definition (0.0096)
```

```
15: project schedule (0.0094)

************************
chapter7.txt_tagged.txt
************************
01: cost baseline (0.0291)
02: cost estimates (0.0258)
03: earned value (0.0167)
04: cost management (0.0161)
05: eac (0.0139)
06: cost performance (0.0125)
07: cost control (0.0122)
08: bac (0.0111)
09: work packages (0.0102)
10: project cost management (0.0102)
11: resource planning (0.01)
12: project activities (0.0097)
13: cost variances (0.0097)
14: cpi (0.0086)
15: actuals (0.0086)

************************
chapter8.txt_tagged.txt
************************
01: quality assurance (0.0351)
02: quality policy (0.0272)
03: quality management (0.0272)
04: quality planning (0.0194)
05: quality improvement (0.0181)
06: control limit (0.0181)
07: quality standards (0.0175)
08: quality management plan (0.0167)
09: sampling (0.014)
10: grade (0.0121)
11: iso (0.0117)
12: quality control (0.0109)
13: errors (0.0092)
14: process flow charts (0.0090)
15: quality control measurements (0.0090)

************************
chapter9.txt_tagged.txt
************************
```

```
01: staff (0.0205)
02: organization chart (0.02)
03: reporting relationships (0.02)
04: human resource management (0.02)
05: competencies (0.0143)
06: literature (0.0133)
07: roles (0.0129)
08: team development (0.0123)
09: staff assignments (0.0116)
10: recognition systems (0.0116)
11: team performance (0.01)
12: team competencies (0.01)
13: project man¡agement team (0.01)
14: employee groups (0.01)
15: staff acquisition (0.0092)

************************
chapter10.txt_tagged.txt
************************
01: communications management plan (0.0251)
02: administrative closure (0.0216)
03: communications planning (0.0209)
04: project records (0.0209)
05: communications management (0.0162)
06: project success (0.0162)
07: performance reporting (0.0132)
08: communications management figure (0.0108)
09: information distribution (0.0099)
10: cpi (0.0083)
11: project status (0.0083)
12: project reports (0.0083)
13: archiving (0.0083)
14: planned value (0.0083)
15: memos (0.0083)

************************
chapter11.txt_tagged.txt
************************
01: risk management (0.0228)
02: risk response (0.0228)
03: risk analysis (0.0213)
04: risk identification (0.016)
05: risk response plan (0.0142)
```

```
06: probability (0.0137)
07: project risks (0.0128)
08: consequences (0.0127)
09: risk events (0.0124)
10: scale (0.0117)
11: project objective (0.0103)
12: risk monitoring (0.0089)
13: response planning (0.0089)
14: risk management plan (0.0088)
15: risk management processes (0.0083)

************************
chapter12.txt_tagged.txt
************************
01: seller (0.0245)
02: procurement items (0.0229)
03: contract closeout (0.0166)
04: procurement management (0.0158)
05: contract administration (0.0158)
06: proposals (0.0153)
07: source selection (0.0136)
08: sow (0.0136)
09: procurement planning (0.0133)
10: solicitation planning (0.0125)
11: buyer (0.0122)
12: arrangements (0.0113)
13: procurement management plan (0.0105)
14: evaluation criteria (0.0102)
15: payment system (0.0102)
```

# Tillegg E

# Kildekode

Listing E.1: Globals.java

```
package noOntoSearch;

public class Globals
{

        public static String indexDirectory = "c:\\index";
        public static String contentFieldName = "content";
        public static String filenameFieldName = "filename";
}
```

Listing E.2: Phrase.java

```
package noOntoSearch;

public class Phrase
{
        String[] words;
        float boost;

        public Phrase(String words, float boostFactor)
        {
                this.words = words.split("␣");
                boost = boostFactor;
        }

        public float getBoost()
        {
                return boost;
        }

        public String[] getWords()
        {
                return words;
        }
}
```

Listing E.3: Indexer.java

```
package noOntoSearch;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.index.*;

public class Indexer {

        public Indexer() {}

        public void indexFiles(String directory) {
```

```
                try {

                IndexWriter iw = new IndexWriter(Globals.indexDirectory,
                                                 new PorterStemAnalyzer(),true);

                for(int i=1;i<=76;i++ ) {

                        String fileName = directory+"dok"+i+".txt";
                        FileReader fr = new FileReader(fileName);
                        BufferedReader reader = new BufferedReader(fr);
                        String content = "";
                        String line = "";
                        while((line = reader.readLine()) != null) {
                                content = content.concat(line.concat(" "));
                                }

                        Document doc = new Document();
                        doc.add(new Field(Globals.contentFieldName,content,Field.Store.YES,
                                                        Field.Index.TOKENIZED));
                        doc.add(new Field(Globals.filenameFieldName,fileName,Field.Store.YES,
                                                        Field.Index.TOKENIZED));
                        iw.addDocument(doc);
                        }

                        iw.optimize();
                        iw.close();
                }
                catch (IOException e) {
                        e.printStackTrace();
                        }

        }
}
```

Listing E.4: PhraseSearcher.java

```
package noOntoSearch;

import java.io.IOException;
import java.util.ArrayList;

import org.apache.lucene.document.Document;
import org.apache.lucene.index.IndexReader;
import org.apache.lucene.index.Term;
import org.apache.lucene.search.BooleanClause;
import org.apache.lucene.search.BooleanQuery;
import org.apache.lucene.search.Hits;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.PhraseQuery;
import org.apache.lucene.analysis.PorterStemmer;

public class PhraseSearcher {


        public PhraseSearcher() {}

        public void searchForPhrase(ArrayList phrases)
        {
                BooleanQuery bq = null;
                try {
                        PorterStemmer stemmer = new PorterStemmer();
                        IndexReader ir = IndexReader.open(Globals.indexDirectory);
                        IndexSearcher   searcher = new IndexSearcher(ir);

                        bq = new BooleanQuery();
                        for(int i = 0; i < phrases.size();i++)
                        {
                                Phrase phrase = (Phrase)phrases.get(i);
                                PhraseQuery pq = new PhraseQuery();
                                for(int j = 0; j < phrase.getWords().length;j++)
                                {
                                pq.add(new Term(Globals.contentFieldName,
                                        stemmer.stem(phrase.getWords()[j])));
                                }
                                pq.setBoost(phrase.getBoost());
                                bq.add(pq,BooleanClause.Occur.SHOULD);
                        }
                        Hits hits = searcher.search(bq);
                        System.out.println("Antall treff: "+hits.length());
                        for(int i = 0; i < hits.length(); i++)
                        {
                                Document doc = hits.doc(i);
                                System.out.println("Fikk treff i dokument: "
                                  +doc.getField(Globals.filenameFieldName).stringValue());
                                System.out.println("Score: "+ hits.score(i));
```

```
                }

            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
        public static void main(String[] args)
        {
            Indexer indexer = new Indexer();
            indexer.indexFiles("c:\\ontosearch\\ontosearch\\dok\\docs\\");
            PhraseSearcher searcher = new PhraseSearcher();
            searcher.searchForPhrase(searcher.GetTestSet3());
        }
        public ArrayList GetTestSet1()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("cost estimating",1.5f));
            phrases.add(new Phrase("resource planning",1.0f));

            return phrases;
        }

        public ArrayList GetTestSet2()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("cost estimating",1.5f));
            phrases.add(new Phrase("estimate at completion",1.0f));
            phrases.add(new Phrase("eac",1.0f));

            return phrases;
        }

        public ArrayList GetTestSet3()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("human resource",1.5f));
            phrases.add(new Phrase("hr",1.5f));
            phrases.add(new Phrase("organizational planning",1.0f));
            phrases.add(new Phrase("staff",1.0f));
            phrases.add(new Phrase("staff acquisition",1.0f));
            phrases.add(new Phrase("team development",1.0f));

            return phrases;
        }

        public ArrayList GetTestSet4()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("human resource",1.5f));
            phrases.add(new Phrase("staff",1.0f));
            phrases.add(new Phrase("human resource management",1.0f));
            phrases.add(new Phrase("organizational chart",1.0f));
            phrases.add(new Phrase("role",1.0f));
            phrases.add(new Phrase("staff assignment",1.0f));
            phrases.add(new Phrase("team competencies",1.0f));
            phrases.add(new Phrase("team development",1.0f));

            return phrases;
        }

        public ArrayList GetTestSet5()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("stakeholder",1.5f));
            phrases.add(new Phrase("buyer",1.0f));
            phrases.add(new Phrase("customer",1.0f));
            phrases.add(new Phrase("external stakeholder",1.0f));
            phrases.add(new Phrase("internal stakeholder",1.0f));
            phrases.add(new Phrase("project staff",1.0f));
            phrases.add(new Phrase("seller",1.0f));
            phrases.add(new Phrase("vendor",1.0f));
            phrases.add(new Phrase("supplier",1.0f));
            phrases.add(new Phrase("subcontractor",1.0f));
            phrases.add(new Phrase("sponsor",1.0f));

            return phrases;
        }

        public ArrayList GetTestSet6()
        {
            ArrayList phrases = new ArrayList();
            phrases.add(new Phrase("stakeholder",1.5f));
            phrases.add(new Phrase("buyer",1.0f));
            phrases.add(new Phrase("seller",1.0f));
            return phrases;
        }
```

```
public ArrayList GetTestSet7()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("procurement",1.5f));
        phrases.add(new Phrase("bidder conference",1.0f));
        phrases.add(new Phrase("vendor conference",1.0f));
        phrases.add(new Phrase("contractor conference",1.0f));
        phrases.add(new Phrase("pre-bid conference",1.0f));
        phrases.add(new Phrase("contract",1.0f));
        phrases.add(new Phrase("procurement planning",1.0f));
        phrases.add(new Phrase("solicitation",1.0f));
        phrases.add(new Phrase("source selection",1.0f));

        return phrases;
}

public ArrayList GetTestSet8()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("procurement",1.5f));
        phrases.add(new Phrase("contract administration",1.0f));
        phrases.add(new Phrase("contract closeout",1.0f));
        phrases.add(new Phrase("procurement item",1.0f));
        phrases.add(new Phrase("procurement management",1.0f));
        phrases.add(new Phrase("procurement management plan",1.0f));
        phrases.add(new Phrase("procurement planning",1.0f));
        phrases.add(new Phrase("proposal",1.0f));
        phrases.add(new Phrase("solicitation planning",1.0f));

        return phrases;
}

public ArrayList GetTestSet9()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("risk response",1.5f));
        phrases.add(new Phrase("risk response plan",1.0f));
        phrases.add(new Phrase("risk register",1.0f));

        return phrases;
}

public ArrayList GetTestSet10()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("risk response",1.5f));
        phrases.add(new Phrase("risk response plan",1.0f));

        return phrases;
}

public ArrayList GetTestSet11()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("work breakdown structure",1.5f));
        phrases.add(new Phrase("wbs",1.5f));

        return phrases;
}

public ArrayList GetTestSet12()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("work breakdown structure",1.5f));
        phrases.add(new Phrase("breakdown structure",1.5f));

        return phrases;
}

public ArrayList GetTestSet13()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("earned value",1.5f));
        phrases.add(new Phrase("budget cost of work performed",1.5f));
        phrases.add(new Phrase("bcwp",1.5f));
        phrases.add(new Phrase("ev",1.5f));
        phrases.add(new Phrase("actual cost",1.0f));
        phrases.add(new Phrase("planned value",1.0f));

        return phrases;
}

public ArrayList GetTestSet14()
{
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("earned value",1.5f));
        phrases.add(new Phrase("ev",1.5f));
```

```
        return phrases;
    }

    public ArrayList GetTestSet15()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("earned value management",1.5f));
        phrases.add(new Phrase("evm",1.5f));
        phrases.add(new Phrase("control account plan",1.0f));
        phrases.add(new Phrase("earned value",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet16()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("earned value management",1.5f));
        phrases.add(new Phrase("evm",1.5f));

        return phrases;
    }

    public ArrayList GetTestSet17()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("communications management",1.5f));
        phrases.add(new Phrase("communications management plan",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet18()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("communications management",1.5f));
        phrases.add(new Phrase("communications management plan",1.0f));
        phrases.add(new Phrase("communications management figure",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet19()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("performance",1.5f));
        phrases.add(new Phrase("performance measurement",1.0f));
        phrases.add(new Phrase("performance reporting",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet20()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("performance",1.5f));
        phrases.add(new Phrase("team performance",1.0f));
        phrases.add(new Phrase("performance reporting",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet21()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("planning process",1.5f));
        phrases.add(new Phrase("core process",1.0f));
        phrases.add(new Phrase("facilitating process",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet22()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("planning process",1.0f));

        return phrases;
    }

    public ArrayList GetTestSet23()
    {
        ArrayList phrases = new ArrayList();
        phrases.add(new Phrase("control account plan",1.5f));
        phrases.add(new Phrase("cost account plan",1.5f));
        phrases.add(new Phrase("cap",1.5f));
```

```
                return phrases;
        }

        public ArrayList GetTestSet24()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("control account plan",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet25()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("project management information system",1.5f));
                phrases.add(new Phrase("pmis",1.5f));

                return phrases;
        }

        public ArrayList GetTestSet26()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("project management information system",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet27()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("time management",1.5f));
                phrases.add(new Phrase("activity",1.0f));
                phrases.add(new Phrase("schedule development",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet28()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("time management",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet29()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("integration management",1.5f));
                phrases.add(new Phrase("integration change control",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet30()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("integration management",1.0f));

                return phrases;
        }

        public ArrayList GetTestSet31()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("cost performance index",1.5f));
                phrases.add(new Phrase("cpi",1.5f));

                return phrases;
        }

        public ArrayList GetTestSet32()
        {
                ArrayList phrases = new ArrayList();
                phrases.add(new Phrase("cost performance index",1.0f));

                return phrases;
        }
}
```

# Tillegg F

# Digitalt vedlegg

Vedlagt denne rapporten ligger en zip-fil som inneholder følgende:

- De to ontologiene, samt Protégé prosjektfiler.
- Dokumentsamlingen brukt i IR-testen.
- LaTeX-kode for denne rapporten.
- Regneark med domeneekspertens vurdering av de ulike begrepene brukt i ontologiene.

# Bibliografi

[Apache, 2006] Apache (2006). Welcome to Apache Lucene. `http://lucene.apache.org/`.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley. Available from: `citeseer.ist.psu.edu/baeza-yates99modern.html`.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. Scientific America.

[Blomqvist, 2005] Blomqvist, E. (2005). Fully Automatic Construction of Enterprise Ontologies Using Design Patterns: Initial Method and First Experiences. In *OTM Conferences (2)*, pages 1314–1329.

[Borch, 2005] Borch, H. (2005). Automatic Keyphrase Extraction. Prosjektoppgave i fag TDT4730 IDI/NTNU.

[Brank et al., 2005] Brank, J., Grobelnik, M., and Mladenic, D. (2005). A Survey of Ontology Evaluation Techniques. Conference on Data Mining and Data Warehouses (SiKDD 2005).

[Brasethvik, 2004] Brasethvik, T. (2004). *Conceptual modelling for domain specific document description and retrieval*. Doktoravhandling for graden doktor ingeniør, IDI/NTNU.

[Brewster et al., 2004] Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). Data driven ontology evaluation. International Conference on Language Resources and Evaluation, Lisbon, 2004.

[Brill, 1992] Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155. Association for Computational Linguistics.

[DataMystic, 2006] DataMystic (2006). Search and Replace, Data Extraction, Data Transformation, Report Mining and Text Manipulation Software. `http://www.crystalsoftware.com.au/textpipe.html`.

[de Almeida Falbo et al., 1998] de Almeida Falbo, R., de Menezes, C. S., and Rocha, A. R. (1998). A systematic approach for building ontologies. In *IBERAMIA '98: Proceedings of the 6th Ibero-American Conference on AI*, pages 349–360, London, UK. Springer-Verlag.

[Denny, 2002] Denny, M. (2002). Ontology Building: A Survey of Editing Tools. `http://www.xml.com/pub/a/2002/11/06/ontologies.html`.

[Gomez-Perez, 1994] Gomez-Perez, A. (1994). Some Ideas and Examples to Evaluate Ontologies. Available from: `citeseer.ist.psu.edu/gomez-perez94some.html`.

[Gomez-Perez, 1996] Gomez-Perez, A. (1996). Towards a Framework to Verify Knowledge Sharing Technology. Expert Systems with Applications, 11(4):519–529.

[Grimnes, 2005] Grimnes, G. (2005). Metoder for ontologikonstruksjon i Statoil. Prosjektoppgave i fag TDT4730 IDI/NTNU.

[Guarino, 1998] Guarino, N. (1998). Formal Ontology and Information Systems. Available from: `citeseer.csail.mit.edu/guarino98formal.html`.

[Gulla, 2003] Gulla, J. A. (2003). IR Systems Evaluation. Kursmateriale TDT4215, IDI, NTNU.

[Gulla et al., 2002] Gulla, J. A., Auran, P. G., and Risvik, K. M. (2002). Linguistic Techniques in Large-Scale Search Engines. Fast Search & Transfer ASA.

[Gulla et al., 2006] Gulla, J. A., Borch, H. O., and Ingvaldsen, J. E. (2006). Unsupervised Keyphrase Extraction for Search Ontologies. 9th Int. Conf. on Business Information Systems.

[Hawking et al., 2001] Hawking, D., Craswell, N., Bailey, P., and Griffihs, K. (2001). Measuring Search Engine Quality. *Information Retrieval*, 4(1):33–59. Available from: `citeseer.ist.psu.edu/hawking01measuring.html`.

[Holt and Chung, 1999] Holt, J. D. and Chung, S. M. (1999). Efficient mining of association rules in text databases. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 234–242, New York, NY, USA. ACM Press.

[Jenz and Partner, 2006] Jenz and Partner (2006). The Open Source Business Management Ontology. `http://www.bpiresearch.com/Resources/RE_OSSOnt/re_ossont.htm`.

[Lozano-Tello and Gomez-Perez, 2004] Lozano-Tello, A. and Gomez-Perez, A. (2004). ONTOMETRIC: A Method to Choose the Appropriate Ontology.

[Maedche and Staab, 2002] Maedche, A. and Staab, S. (2002). Measuring Similarity between Ontologies. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK. Springer-Verlag.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

[McGuineness, 2001] McGuineness, D. L. (2001). Ontologies Come of Age. MIT Press.

[Mindswap, 2006] Mindswap (2006). SWOOP - hypermedia-based OWL Ontology Browser and Editor. `http://www.mindswap.org/2004/SWOOP/`.

[Mooers, 1950] Mooers, C. (1950). Information Retrieval viewed as Temporal Signalling. Recited in [Savino and Sebastiani, 1998].

[NIST, 2006] NIST (2006). Text REtrieval Conference (TREC) Home Page. `http://trec.nist.gov/`.

[Paralic and Kostial, 2005] Paralic, J. and Kostial, I. (2005). Ontology-based Information Retrieval. Technical University of Kosice.

[(PMI), 2000] (PMI), P. M. I. (2000). A Guide to the Project Management Body of Knowledge.

[(PMI), 2003] (PMI), P. M. I. (2003). Organizational project management maturity model.

[(PMI), 2006] (PMI), P. M. I. (2006). The largest project management association. `http://www.pmi.org/info/default.asp`.

[Porzel and Malaka, 2004] Porzel, R. and Malaka, R. (2004). A Task-based Approach for Ontology Evaluation. ECAI Workshop on Ontology Learning and Population, Valencia, Spain, 2004.

[Racer-Systems, 2006] Racer-Systems (2006). RacerPro - an OWL reasoner and inference server for the Semantic Web! `http://www.racer-systems.com/`.

[Savino and Sebastiani, 1998] Savino, P. and Sebastiani, F. (1998). Essential bibliography on multimedia information retrieval, categorisation and filtering. In Slides of the 2nd European Digital Libraries Conference Tutorial on Multimedia Information Retrieval 1998.

[Schneider and Dean, 2004] Schneider, G. and Dean, M. (2004). OWL Web Ontology Language Reference. `http://www.w3c.org/TR/owl-ref`.

[Stanford, 2006] Stanford, M. I. (2006). The Protégé Ontology Editor and Knowledge Acquisition System. `http://protege.stanford.edu/`.

[Thommasen et al., 2005] Thommasen, S. L., Gulla, J. A., and Strasunskas, D. (2005). Ontology Support for Query Interpretation. IDI/NTNU.

[TopQuadrant, 2006] TopQuadrant (2006). Topbraid Composer. `http://www.topbraidcomposer.com/index.html`.

[Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, Methods and Applications. University of Edinburgh.

[Uschold and King, 1995] Uschold, M. and King, M. (1995). Towards a Methodology for Building Ontologies. University of Edinburgh.

[W3C, 2004] W3C (2004). OWL Web Ontology Language Overview. `http://www.w3.org/TR/owl-features/`.

[W3C, 2006] W3C (2006). World Wide Web Consortium. `http://www.w3c.org`.