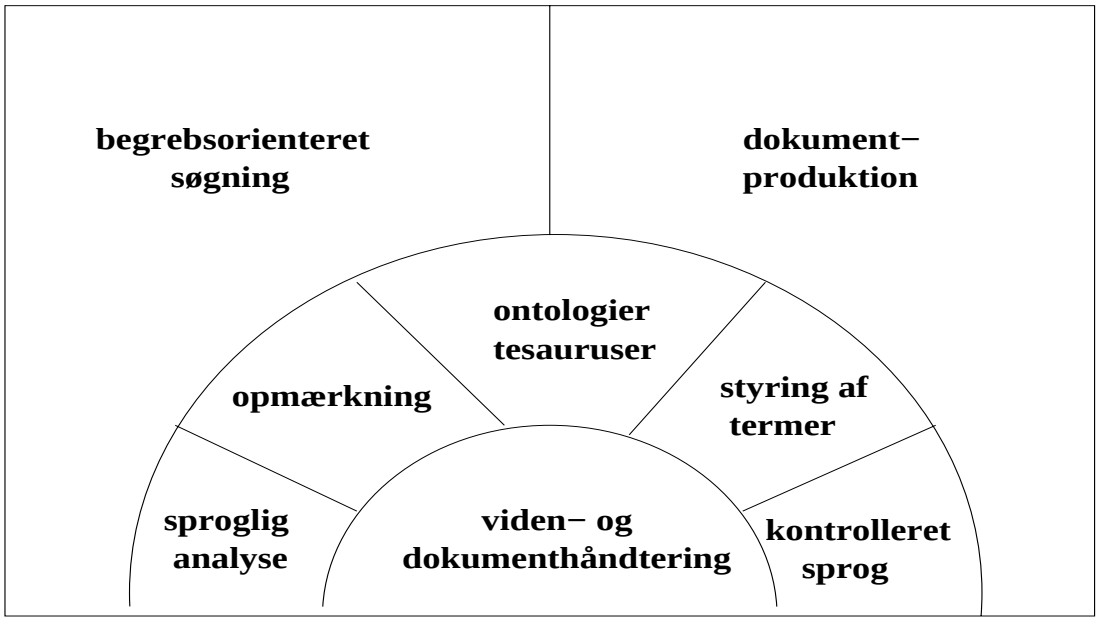


Statistiske metoder der kan understøtte opbygning af tekstbaserede ontologier

Costanza Navarretta

VID - rapport 7
Center for Sprogteknologi
Februar 2005

Om VID: Viden- og Dokumenthåndtering med sprogteknologi

Der er et udtalt behov hos danske virksomheder for at kunne supplere deres eksisterende sproglige kompetence og viden med sprogteknologiske værktøjer og metoder som dels kan støtte medarbejderne, dels forankre viden og processer i virksomhedens IT-systemer, dels danne grundlag for den udvikling der kræves hvis virksomhederne skal overleve og vokse i den stadigt mere globaliserede økonomi.

VID-projektet er et forsknings- og udviklingsprojekt der har til formål at udforske de forskellige muligheder som sprogteknologi frembyder inden for informationssøgning og dokumentproduktion, og at understøtte de deltagende virksomheder i at udvikle værktøjer til bedre udnyttelse af egen viden, samt til bedre og mere effektiv produktion af dokumentation, herunder flersproget dokumentation. Foruden CST omfatter projektet på den ene side virksomhederne Bang & Olufsen A/S, Zacco A/S og Nordea A/S, som i dette projekt udgør teknologiens brugere, på den anden Navigo Systems A/S og Ankiro, som er teknologiproducenter. Projektet omfatter følgende forskningsopgaver:

- analyse af de tekstuelle data virksomhederne skal kunne håndtere for at kunne fastlægge tesauruser/ontologier for de relevante semantiske domæner, undersøgelse af den bedst egnede formalisme/teknologi til at udtrykke disse;
- afdækning og videreudvikling af sprogteknologiske komponenter til brug for automatisk tekstklassifikation og begrebsorienteret informationssøgning, indbefattende tilpasning af sprogteknologiske 'basismoduler' til opmærkning af tekst;
- udforskning af flertydighed i tekstuelle data som kan vanskeliggøre informationssøgning; ligeledes den omvendte problematik: at samme indhold kan udformes forskelligt rent sprogligt og derfor kan være svær at fremfinde i store datamængder;
- forskning inden for kontrolleret sprog - også set i et flersproget perspektiv - til brug for dokumentproduktion; herunder analyse af den sprogstil og tone som virksomhederne ønsker at anvende, samt opstilling af modeller for dette sprog;
- undersøgelse af hvilke sprogteknologiske metoder der kan anvendes til denne kvalitetssikring af dokumentproduktionen i form af f.eks. termstyring og grammatikkontrol.

Projektet er støttet af Center for IT-forskning og løber i perioden 2003-2004.

Indhold

Kapitel 1

Indledning

At opbygge ontologier er en tids- og resursekrævende proces, selvom ontologierne kun modellerer begrænsede domæner. Traditionelt opbygges ontologier på baggrund af ekspertviden, men i den seneste tid har man forsøgt at opbygge eller evaluere ontologier ved at anvende store tekstsamlinger (korpora) der tilhører de pågældende domæner (Buitelar, Olejnik, Hutanu, Schutz, Declerck & Sintek 2004, Pedersen, Navarretta & Henriksen 2004). Fordelene ved at inddrage tekstkorpora i opbygning af ontologier er mange. Først og fremmest kan korpora støtte og supplere den menneskelige introspektion i samlingen af den grundlæggende domænevokabular (både viden om begreber (klasser) og relationerne som holder mellem disse begreber). Brugen af korpora som videnkilde kan forhøje konsistensen og kvaliteten af de opbyggede ontologier. Processen i at opbygge ontologier kan blive mindre tids- og resursekrævende fordi uddragelsen af information fra tekster kan delvis automatiseres. Endelig afspejler teksterne den reelle brug af domænesproget. At afdække denne brug er især vigtigt når man bygger ontologier der skal anvendes i applikationer der tillader brugere at anvende naturssprogsudtryk i brugergrænsefladerne. Begrænsningerne ved at anvende tekster til ontologiopbygningen er føgende: det er ikke alt den nødvendige domæneviden der er udtrykt i tekster; det kan være svært for ontologiopbyggere at få overblik over et domæne som de ikke er eksperter i ud fra tekster alene. På grund af disse begrænsninger bør tekster ikke betragtes som den eneste videnkilde til opbygning af ontologier og domæneeksperter bør stadig deltage aktivt i denne proces. I (Jongejan, Pedersen & Navarretta 2004, Navarretta, Pedersen & Hansen 2004, Pedersen et al. 2004) beskrev vi hvordan termer og generelle ord som er centrale i domænet af patentbehandling blev, semiautomatisk uddraget fra et korpus bestående af standarddokumenter om patentbehandling, samlet af sagsbehandlere i Zacco A/S. De ord og termer som fandtes i patentkorpusset, suppleret med termer angivet af domæneeksperterne, har dannet grundlaget for en ontologi som modellerer domænet.

I denne rapport beskriver vi statistiske metoder til at støtte gruppering af domænerelevante termer/ord på baggrund af deres forekomster i tekster. Disse metoder går under navnet af "clusteringsalgoritmer". Rapporten indeholder først en generel introduktion til brugen af de mest grundlæggende statistiske metoder til automatisk at gruppere lingvistiske data ud fra deres forekomster i tekster (kapitel 2). Dernæst beskrives i de mest anvendte typer af statistiske algoritmer til at gruppere data semantisk. I kapitel 3 beskrives de resultater vi har opnået ved at anvende clusteringsalgoritmer på Zaccos standard patentdokumenter. I kapitlet beskrives også resultaterne af at anvende avancerede clusteringsmetoder på engelske, opmærkede korpora for at finde semantisk relaterede ord til engelske ord som er oversættelse af nogle af de danske centrale ord i patentdomænet. I kapitlet sammenligner vi de automatisk opnåede grupperinger af semantisk relaterede ord med de grupperinger som blev fundet ved manuelt at analysere det samme korpus (Pedersen et al. 2004). Rapporten afsluttes med en kort konklusion og perspektivering.

Kapitel 2

Statistiske metoder til at gruppere data

De seneste årtier er det blevet mere og mere almindeligt at anvende statistiske metoder og algoritmer i natursprogsbehandling. Eksempler på de områder som statistik anvendes på, er talegenkendelse, analyse af tekstkorpora, tagging, parsing, maskinoversættelse, tekstforståelse, informationsuddragelse og automatisk katalogisering.

Inden for lingvistik anvendes statistik ofte til at beskrive hvordan sproget bliver brugt i det virkelige liv, fx hvor tit bestemte udtryk anvendes i bestemte resurser produceret af bestemte sprogbrugere (deskriptiv statistik). Statistik kan dog også anvendes til at forudsige sproglige fænomener i bestemte kontekster, og denne anvendelse er blevet mere og mere udbredt i den automatiske natursprogsbehandling.

Alle statistiske algoritmer baseres på sandsynlighedsteorien som angiver sandsynligheden for at et bestemt fænomen kan forekomme i en bestemt kontekst ud fra de data som man har tidligere set. Sandsynlighedsteorien ligger til grund for opbygningen af sprogmodeller som anvendes til at forudsige ukendte data. Sandsynligheden for og nøjagtigheden af en sprogmodel afhænger af mængden af de data som er blevet brugt til at definere modellen. Desto flere data der ligger til grund for en sprogmodel, desto mere sandsynlig er modellen.

De mest anvendte statistiske modeller inden for natursprogsbehandling er de såkaldte n-gramsmodeller se blandt andre (Church 1988, Brown, Cocke, Pietra, Pietra, Jelinek, Lafferty, Mercer & Roossin 1990, Jelinek 1990). Også i n-gramsmodeller anvendes kendte data til at forudsige endnu ukendte data. De lingvistiske data som kan modelleres med n-gramsmodeller er mange og inkluderer fonemer, bogstaver og/eller tegn, ord, kombination af ord, sætninger, afsnit. Fx er det muligt at forudsige forekomsten af et ord o_n ved at kigge på de forudgående ord

(ordets historie) og beregne sandsynlighedsfunktionen Pr for forekommende ord med formlen i (1):

(1) $$Pr(o_n \mid o_1 \dots o_{n-1}) = \frac{Pr(o_1 \dots o_n)}{Pr(o_1 \dots o_{n-1})}$$

Desværre vokser parametrene i n-gramsmodellerne ret hurtigt når modellerne anvendes på store mængder data. En løsning på dette problem er blevet foreslået af Markov (Markov 1913) og kaldes for *Markovsforudsætning*en. I følge Markovsforudsætningen er det muligt at forudsige et objekt ved udelukkende at kigge på dets seneste historie, dvs. man kan reducere mængden af de observerede data til få objekter. Derfor kaldes ngramsmodeller også for *Markovskæder*. I de mest anvendte n-gramsmodeller anvendes kun to, tre eller fire objekter (fx. fonemer, tegn, ord) til at forudsige kommende data, dvs man arbejder med n-gramsmodeller hvor n er lige med 2, 3 eller 4. De tilsvarende sprogmodeller kaldes da henholdsvis bigrams-, trigrams- og fire-gramsmodeller.

Ved hjælp af n-gramsmodeller kan man blandt andet identificere klynger (*clusters*) af ord der ligner hinanden, og disse metoder anvendes i forskellige applikationer såsom talesprogsgenkendelse, tagging, stokastisk parsing, mm.

Simple n-gramsmodeller kan anvendes til at gruppere ord der optræder i samme kontekster. Resultatet af at anvende en simpelt n-gramsmodel på tekster vil for eksempel være at indsætte ord der tilhører den samme ordklasse (fx. præpositioner, personlige pronominer, artikler) i de samme grupper. Mere problematisk er grupperinger af mere sofistikerede fænomener, som fx. indholdsord i forhold til deres betydning. Dette er især problematisk fordi indholdsord med bestemte betydninger ikke forekommer så hyppigt. det faktum at nogle lingvistiske data forekommer sjældent kaldes for *the data sparseness problem*. Der findes forskellige metoder for at tage højde for dette problem. Disse metoder kaldes diskonteringsmetoder eller jævningsmetoder (*smoothing*).

I det følgende beskriver vi statistiske algoritmer til automatisk at opdele grupper af ord som "ligner"hinanden semantisk. Disse algoritmer går under navnet *clustering*. Hovedkilde for vores beskrivelse er Manning & Schütze (1999).

2.1 Clusteringsalgoritmer

Clusteringsalgoritmer opdeler data i grupper eller klynger (clusters) på basis af graden af lighed mellem de enkelte data. Dvs. at objekter som ligner hinanden mest, indsættes i samme gruppe, mens objekter der er meget forskellige, placeres i adskilte grupper (se figure 2.1).

I opbygningen af ontologier og/eller i klassificeringsapplikationer fokuseres der

Figur 2.1: Clustering

på semantisk lighed. Man antager at ord der semantisk ligner hinanden, ofte optræder i lignende kontekster. Mere præcist defineres semantisk lighed som graden hvorpå ord kan erstatte hinanden i samme kontekst (G.A.Miller & W.G.Charles 1991).

Ligheden i clusteringsalgoritmer defineres via attributter og værdier. Mængden af attributter og værdier kaldes datarepræsentationsmodellen.

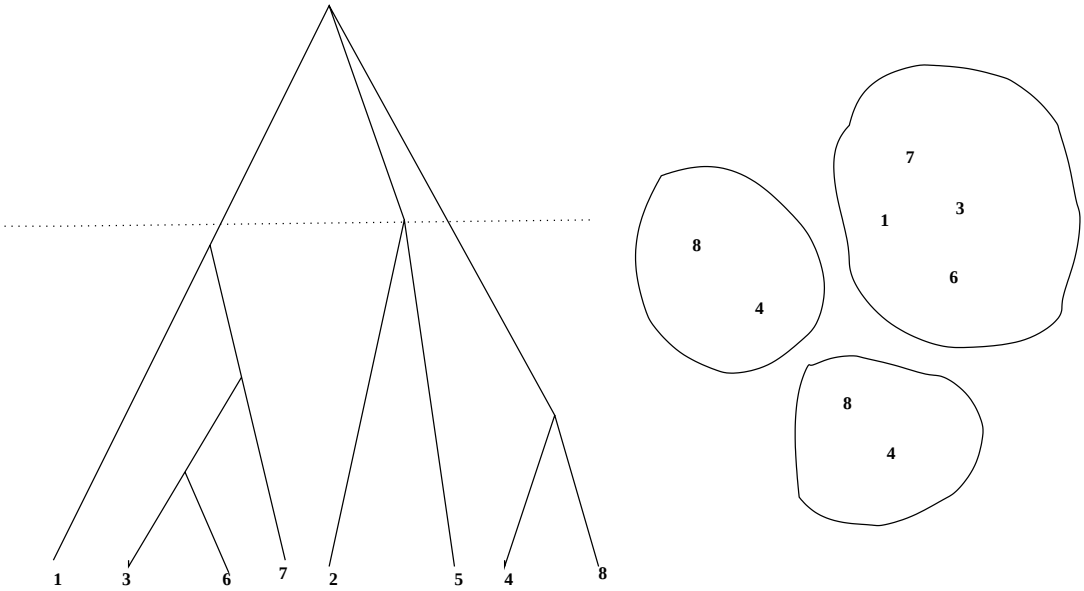
Clusteringsalgoritmer har mange lighedspunkter med klassifikationsalgoritmer. Forskellen mellem de to typer algoritmer er at klassifikationsalgoritmer kræver en mængde af opmærkede eksempler (både positive og negative eksempler) for hver klassifikationsgruppe, mens clusteringsalgoritmer ikke forudsætter præeksisterende træningsdata. Derfor kaldes clustering for "ikke overvåget eller automatisk klassificering"(*unsupervised or automatic classification*).

Clusteringsalgoritmer, som andre statistiske metoder, bruges til at analysere data ud fra deres forekomster eller til at generalisere over data.

Der findes to hovedtyper af clusteringsalgoritmer: hierarkiske og ikke-hierarkiske (eller flade) algoritmer. I hierarkiske algoritmer bliver data organiseret i hierarkisk ordnede grupper, således at en knude i hierarkien er en subklasse af moderknuden. I ikke-hierarkiske clusteringsalgoritmer bliver data opdelt i grupper som ikke har nogen indbyrdes relation. De to algoritmetyper er illustreret i figur 2.2.

I nogle clusteringsalgoritmer kan objekter kun tilhøre en gruppe. Disse algoritmer går under navnet af "hårde algoritmer"(*hard clustering*). I andre algoritmer kan det samme objekt forekomme i flere grupper. Disse algoritmer kaldes for bløde clusteringsalgoritmer (*soft clustering*). Hierarkiske algoritmer er (næsten) altid *hårde*, mens ikke-hierarkiske algoritmer kan være hårde eller bløde.

Der findes en stor mængde af clusteringsalgoritmer inden for både den hierarkiske og den ikke-hierarkiske type. Desuden er der blevet defineret *hybride* algoritmer som kombinerer top-down og bottom-up strategier på forskellige måder og niveauer.



Figur 2.2: Hierarkisk og ikke-hierarkisk clustering

2.1.1 Lighed

Clusteringsalgoritmer inddeler ord i grupper ved at måle ordenes lighedsgrad. Det er almindeligt i clusteringsalgoritmer at repræsentere ord som vektorer i et multidimensionelt rum. For eksempel kan man repræsentere ord som vektorer i dokumentrummet (hvor mange gange ord forekommer i hvert dokument), i ordrummet (forekomst med andre ord i hele korpus) og i rummet af grammatiske relationer. Man kan dog kun bruge grammatiske relationer hvis ens data er opmærket med oplysninger om disse relationer. Eksempler på substantiver repræsenteret i forhold til de adjektiver der modificerer dem, er givet i tabel 2.1. Når ord repræsenteres

adj	patent/er	patentansøgning/er	kvittering/er	skrivelse/r
europæisk/e	73	46	0	0
officiel/lle	0	0	16	62

Tabel 2.1: Substantiver repræsenteret i forhold til modificerende adjektiver

som vektorer, kan semantisk lighed beregnes som lighed mellem disse vektorer.

I den enkleste repræsentation betragtes ord som binære vektorer, dvs. vektorer hvis indgange kun kan indeholde 0 eller 1 (fx. $\vec{x} = \langle 0001110 \rangle$). Man kan da nøjes med at tage de indgange i betragtning som ikke har nul-værdier. Lighedsgraden af to binære vektorer X og Y kan beregnes med følgende mængdeoperationer:

- **tilpasningskoefficient** (matching coefficient)
- **Dice-koefficient**
- **Jaccard- (eller Tanimoto-) koefficient**
- **overlapskoefficient**
- **kosinus**

Ved anvendelsen af tilpasningskoefficient tælles antallet af dimensioner hvor både X og Y ikke er lige nul:

$$|X \cap Y|$$

I Dice-koefficient-operationen normaliseres på længden af vektorerne ved at dividerere med antallet af indgange som ikke er lig nul:

$$\frac{2|X \cap Y|}{|X|+|Y|}$$

Ved anvendelsen af Jaccard-koefficienten straffes få fælles indgange højere end i Dice-koefficient-operationen:

$$\frac{|X \cap Y|}{|X \cup Y|}$$

Overlapskoefficienten har værdi lig 1 hvis hver indgang med værdi forskellig fra 0 i den første vektor også er forskellig fra nul i den anden vektor og viceversa:

$$\frac{|X \cap Y|}{min(|X|,|Y|)}$$

Kosinus giver de samme resultater som Dice-koefficienten for vektorer med det samme antal indgange som ikke indeholder nul. I kosinus, dog, straffes tilfælde af forskellige typer indgange i de to vektorer mindre end i Dice-koefficient-operationen:

$$\frac{|X \cap Y|}{\sqrt{|X| x |Y|}}$$

Bedre og mere præcise resultater opnås dog hvis ord repræsenteres som vektorrum af reelle tal fordi man kan angive flere oplysninger end til-stede/ikke-til-stede. En vektor af reelle tal $\vec{x}$ med n dimensioner består i serier af n reelle tal, hvor x_i er

det i^{te} element i $\vec{x}$ ($\vec{x}$'s værdi i den i^{te} dimension):

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix}.$$

Længden af en vektor defineres som i (2).

$$(2) \quad |\vec{x}| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Dot-produktet af to vektorer defineres som i (3).

$$(3) \quad \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} \vec{x}\vec{y}.$$

Kosinus af to vektorer beregnes som i (4).

$$(4) \quad cos(\vec{x}, \vec{y} = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} \vec{x}\vec{y}}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

En vektor siges at være normaliseret hvis dens længde følger den euklidiske norm, dvs. $\vec{x} = \sum_{i=1}^{n} x_i^2 = 1$. Kosinus for normaliserede vektorer er lige dot-produktet.

Den euklidiske afstand mellem to vektorer måler hvor langt væk vektorerne er fra hinanden i vektorrummet. Den æuklidiske afstand er givet i (5).

$$(5) \quad |\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Kosinus anvendt på normaliserede vektorer giver samme lighedsvægtning som den der måles med den euklidiske afstand.

2.1.2 Hierarkiske algoritmer

Hierarkiske algoritmer følger "bottom-up" eller "top-down" strategier. Bottom-up strategier starter med at indsætte hvert objekt (ord) i dets egen klynge. Dernæst samles klynger med den højeste lighed. Bottom-up strategier kaldes for *agglomerative*.

Top-down strategier starter med en eneste klynge som indeholder alle objekter. Data bliver dernæst splittet i forskellige undergrupper på baggrund af objekternes indbyrdes forskellighed. Top-down strategier kaldes for *divisive*.

Resultatet af hierarkiske clusteringsalgorimer er klynger af objekter organiseret i en træstruktur kaldet et dendogram. Træroden i dendogrammet er mængden

af alle ord, mens de terminale knuder er de enkelte ord. Ikke-terminale knuder består af grupper indeholdende objekter fra deres datterknuder. Ethvert niveau i dendogrammet repræsenterer derfor en opdeling af data i forskellige klynger. Lighedsfunktionen i hierarkiske algoritmer er altid monotonisk således at lighedsfunktionen ikke bliver større eller mindre under samlings- eller delingsprocessen: $\forall c, c', c'' \subseteq S : min(sim(c, c')) \geq sim(c, c' \cup c'')$

De mest anvendte lighedsfunktionstyper i hierarkiske metoder er følgende:

enkel sammenknytning (simple link): lighed mellem de mest lige gruppemedlemmer

komplet sammenknytning(complete link): lighed mellem de mest ulige gruppemedlemmer

gennemsnitlig sammenknytning(group average): gennemsnitlig lighed mellem gruppemedlemmer, dvs. $cos(x, y)$

De forskellige lighedsfunktionstyper resulterer i forskellige grupperinger af de samme data.

2.1.3 Ikke-hierarkiske algoritmer

Ikke-hierarkiske algoritmer inddeler data i en mængde af adskilte klynger. De fleste algoritmer starter med en mængde af tilfældigt producerede klynger og dernæst flytter de ord fra den ene klynge til den anden indtil man har opnået en bestemt tærskel.

Den mest anvendte ikke-hierarkiske hårde algoritme er *K-means* (MacQueen 1967). I *K-means* bliver klyngerne defineret gennem centermassen af elementerne i hver klynge (centroid), hvor centermassen beregnes som middelværdien af elementerne i klyngen. Middelværdien af to vektorer måles ofte som deres euklidiske afstand. Som angivet i afsnit 2.1.1 angiver den euklidiske afstand hvor langt to vektorer ligger fra hinanden i vektorrummet:

$|\vec{x} - \vec{j}| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Målet med K-means algoritmen er at reducere forskellen mellem ord i den samme klynge og maksimere forskellen mellem ord i forskellige klynger. K-means-algoritme opererer på M som er mængden af alle objekter, mens k er et prædefineret heltal. K-means-algoritmens består af følgende trin:

1. udvælg k centroider, $c_1, c_2, \ldots, c_k$
2. alloker hver x i M til den klynge hvis centroid er tættest på x

3. beregn igen hver klynges centroid på baggrund af de elementer som klyngen indeholder

4. gå til trin 2 med mindre der er opnået et forudbestemt tærskel

EM-algoritmen (Expectation Maximation) beregner en blanding af probabilitetsdistributioner og er en blød algoritme. Ideen bagved EM-algoritmen er at forskellige uafhængige faktorer medvirker til generering af data, men at vi kun kan se den endelige blanding, uden at have oplysning om de enkelte faktorer.

EM-algoritmen er modelbaseret fordi den bruger forskellige modeller til at inddele data i klynger. I algoritmen optimeres afstanden mellem data og de anvendte modeller løbende. Enhver klynge repræsenteres med en parametrisk fordeling i form af en Gausskurve (et kontinuum) eller en diskret fordeling (Poisson fordeling). I EM-algoritmen modelleres data med en blanding af disse fordelinger og derfor kaldes EM-algoritmen for "blanding af Gausskurver"(*Gaussian mixture*). I EM-algoritmen bliver data inddelt i to dele:

1. data der kan observeres $\chi = \vec{x_i}$, hvor hvert element $\vec{x_i} = (x_{i1}, \ldots, x_{im})^T$ er vektoren som svarer til det i^{te} datapunkt

2. data der er skjult ,og derfor ikke kan observeres $\mathcal{Z} = \vec{z_i}$. z_{ij} i hvert $\vec{z_i} = (z_{i1}, \ldots, z_{ik})^T$ er lig 1 hvis objektet i er et medlem af gruppe j, 0 hvis dette ikke er tilfældet. Man kan gruppere data med EM-algoritmen hvis man kender typen af distributionen for de individuelle klynger. Man antager at enhver klynge er en Gausskurve. Man beregner løbende de mest sandsynlige værdier for dens distributionsparametre (gennemsnitsværdien og variansen). Samme objekt kan godt tilhøre forskellige klynger, dog er sandsynligheden for dets tilhørssted i hver klynge forskellig.

EM-algoritmen løser iterativt de to reciprokt afhængige udsagn kendt som *estimate expectation* (skøn af forventede data) og *maximize* (maksimering). Givet at Θ er modellernes parametre, siger det første udsagn (*estimate expetation*) følgende: hvis Θ's værdier kendes, er det muligt at beregne de forventede værdier af den skjulte modelstruktur.

Maximize-udsagnet siger følgende: hvis de forventede værdier af den skjulte modelstruktur er kendt, er det muligt at beregne den maksimale sandsynlighedsværdi (*maximum likelihood value*) for Θ.

EM-algoritmen bryder cirkulariteten i de to udsagn ved at initialisere Θ med en tilfældig værdi. EM-algoritmen består derefter i en iterativ serie af et E(xpectation)-trin efterfulgt af et M(aximation)-trin. EM-algoritmen er monoton, dvs. algoritmens resultater forbedres efter hver iteration. Der er ingen garanti for at EM-algoritmen finder den bedste gruppering.

Expectation- og Maximization-trinnene gentages så længe den logaritmiske sandsynlighedberegning kan forbedres op til en forudbestemt tærskel.

Der er mange mulige applikationer for EM-algoritmen, og selve K-means algoritmen kan fortolkes som en hård version af EM-algoritmen. Desuden kan man anvende andre modeller end Gausskurverne i EM-algoritme.

Kapitel 3

Eksperimenter med clustering

I dette kapitel beskrives nogle test af clusteringsteknikker på standarddokumenter fra patentdomænet (Jongejan et al. 2004) (afsnit 3.1, samt kørsel af clusteringsdemo fra et stort internationalt project Infomap (section 3.2).

3.1 Clustering afprøvet på patenttekster

Vi har testet clustering på dokumenter fra vores patentdomæne ved hjælp af CMU-Cambridge Statistical Language Modeling Toolkit (http://lib.stat.cmu.edu/) og Lnknet-systemet udviklet på MIT Lincoln Laboratory (http://www.ll.mit.edu/IST/lnknet/).

Standarddokumenterne fra patentdomænet er blevet konverteret fra WORD til tekstformat, tagget med morfosyntaktiske oplysninger og lemmatiseret som beskrevet i (Jongejan et al. 2004). Bigrams og trigramsmodeller for indholdsord fra patentdomænet blev uddraget, og der blev skabt sprogmodeller for disse med CMU-Cambridge Statistical Language Modeling Toolkit. Vi anvendte K-means-clustering og EM-clustering i Lnknet. Som lighedsparametre brugte vi bigrams og trigrams i teksterne for indholdsord. Resultaterne fra disse eksperimenter var klynger som både indeholdt enkelte, semantisk relaterede ord og ikke relaterede ord. Generelt disse resultater var dårligere end de resultater beskrevet i litteraturen for lignende data. Årsagen til dette er at vores testmateriale er ret begrænset størrelsesmæssigt, og at fælles kontekster for semantisk relaterede ord ikke har tilstrækkelig høj relativ frekvens.

For at forbedre resultaterne af clustering udnyttede vi det faktum at standarddokumenterne i patentdomænet indeholder en del lister som fx. “Albanien, Letland, Litauen” og alternationer som fx. “patentansøgning/oversættelse/...”. Derfor opmærkede vi automatisk ord i lister og/eller alternationer og tilføjede denne observation som en af clusteringsparametrene. Desuden initialiserede vi EM-

algoritmen med resultaterne opnået ved at anvende K-means-clustering på vores data. Denne test gav ret blandede resultater. Nogle af klyngerne indeholdt data der klart er semantisk relaterede, andre indeholdt ord som ikke intuitivt synes at være relaterede. Eksempler på gode klynger er følgende:

1. Albanien, Letland, Litauen, Slovenien, Rumænien, Makedonien
2. Gambia, Ghana, Kenya, Lesotho,Malawi, Mozambique,Sierra Leone, Sudan, Swaziland, Tanzania, Uganda,Zimbabwe
3. gebyr, afgift, årsafgift, årsgebyr, fornyelsesafgift, kravgebyr
4. patentansøgning, grundansøgning, ansøgning, oversættelse, patent
5. rapport, indleveringsrapport, besvarelse
6. konceptkopi, bilag, skrift, kopi
7. skrivelse, kvittering, faktura

I det følgende vil vi analysere data fra de ovenstående klynger. De første to klynger indeholder betegnelser af lande fra samme geografisek område, henholdsvis Østeuropa og Afrika. Landene i hver gruppe omfattes af samme patentlovgivning og patentbehandling i vores domæne.

Den tredje klynge indeholder substantiver der har med betalinger at gøre.

De sidste fire klynger er sværere at karakterisere. De fleste af objekterne i disse klynger er dokumenter, men det er sværere at karakterisere forskelle mellem data i de forskellige klynger fordi forskellige relationstyper holder mellem disse data. For at karakterisere disse forskelle har vi sammenlignet de automatisk opnåede klynger med de klasser som er blevet kodet manuelt i patentontologien (Pedersen et al. 2004). Klasserne i denne ontologi er lingvistisk motiverede i det de er blevet uddraget ud fra termer og andre relevante ord i Zacco A/S' korpus af standarddokumenter.

Sammenligningsresultaterne af de to typer data er følgende:

- Objekterne i en klynge svarer til instanser af en klasse i patentontologien: dette er tilfældet for alle ord i den første klynge som er instanser af klassen extentionland i ontologien.
- Objekter i en klynge er alle instanser eller underklasser af samme klasse i ontologien, men tilhører forskellige klassifikationsniveauer i ontologien: dette er tilfældet for objekterne i den tredje klynge. Alle ord i denne klynge optræder som underklasser af *Betaling*, dog er *afgift* overklasse for *årsafgift* og *fornyelsesafgift*, mens *gebyr* er overklasse for *årsgebyr* og *kravgebyr*.

- Objekter i samme klynge tilhører forskellige klasser i ontologien. Dette er tilfældet for objekterne i fx klynge 4, hvor *patentansøgning* og *grundansøgning* er underklasser af *ansøgning*. *Ansøgning* er en underklasse af *ansøgningsdokument* som er en underklasse af *dokument*. *Oversættelse* i klynge 4 er klasiificeret under f *dokument* i ontologien, mens *patent* er en underklasse af *convention*. På trods af disse forskelle er det dog klart at der er en vis semantisk relation mellem *patent* og *patentansøgning* og at denne relation ikke er en hyponymisk relation (eller IS-A relation).

- Objekterne i klyngerne er ikke fundet i ontologien: dette er tilfældet for ord som *skrift* og *skrivelse* som ikke blev genkendt som termer eller centrale ord i domænet, (Navarretta et al. 2004, Jongejan et al. 2004).

Konkluderende kan man sige at et stort antal af de automatisk uddragede klynger indeholder data som er semantisk relaterede, men at de semantiske relationer der holder mellem data i samme klynge ikke altid er de samme. Desuden kan relationerne mellem klyngerne heller ikke umiddelbart uddrages.

3.2 Afprøvning af clustering med Infomap-demo

I vores anden række af eksperimenter har vi afprøvet den internet-baserede demo af en bayesisk clusteringsalgoritme på Stanford University (http://infomap.stanford.edu/webdemo), implementeret under Infomap-projektet. Infomaps clustering er trænet på store engelske korpora opmærket med morfosyntaktiske oplysninger. Vi har afprøvet demoet med enkelte ord fra vores domæne oversat til engelsk. Demoet returnerer de ord som er mest relateret til disse i de corpora man vælger at køre demoet med. Vi har kørt demoet med clustering trænet på henholdsvis **British National Corpus** (BNC) og **Wall Street Journal** (WSJ). Resultaterne fra vores kørsler er givet i tabel 3.1.

Data i 3.1 viser hvor forskellige resultater der opnås ved at træne samme algoritme på forskellige typer korpora, i dette tilfælde et almensprogligt korpus og et finansaviskorpus som Wall Street Journal. Resultaterne af clustering for ord som *patent* og *document* adskiller sig ikke meget fra de resultater vi har opnået med de danske tekster. Dog er resultaterne fra Infomap generelt bedre, da det anvendte træningsmateriale er meget større end vores patentkorpus. Resultaterne fra Infomap viser også at de relationer der holder mellem ord i samme klynge, kan være forskellige, fx er den relation der holder mellem *copy* og *print* helt klart af en anden natur end relationen der holder mellem *copy* og *editor*. I disse tilfælde ville resultaterne fra clusteringsalgoritmer forbedres hvis algoritmerne kunne udnytte lingvistisk viden om fx ord repræsenterer animerede eller ikke animerede entiteter, eller valensrelationer.

Korp	nøgleord	relateret ord
BNC	**patent**	copyright registration registered statute infringement lawful register law
WSJ		patents infringement burroughs infringe infringed copyright court trademark genentech
BNC	**document**	draft revised handwritten circulated paragraph documentation printed copy
WSJ		confidential detailed reviewed internal matters disclosure information contents
BNC	**fee**	pay instalments monthly payment cancellation subscription sum salary
WSJ		commissions waived charging schwab's expense schwab deductibles payment extra
BNC	**patent application**	applicant complaint registration contravention registered registry lodged
WSJ		patents fda interferon avonex berlex generic biogen osteoporosis
BNC	**copy**	printed copied typed write letter edition print page
WSJ		printed page prints reporter edition handwritten editors writer
BNC	**slovenia**	croatia territories armenia baltic georgia yugoslavia hercegovina ukraine serbia
WSJ		NONE

Tabel 3.1: Resultater fra Infomap-clustering

Kapitel 4

Sammenfatning og perspektivering

I denne rapport har vi beskrevet nogle af de mest anvendte statistiske metoder til at gruppere data fra store mængder tekster ud fra forskellige lighedskriterier. Vi har især fokuseret på clusteringsmetoder til at gruppere semantisk relaterede ord ud fra den antagelse at ord der kan optræde i samme kontekst ligner hinanden semantisk.

I det sidste afsnit af rapporten har vi afprøvet EM-clusteringsteknikker på tekster fra vores patentdomæne og har sammenlignet nogle af de opnåede resultater med klasserne i en lingvistisk-baseret ontologi som er blevet manuelt opbygget ud fra de samme tekster.

Resultaterne fra clustering trænet på vores korpus var for de fleste ord dårligere end resultaterne fra samme algoritmetype i litteraturen. Dette skyldes hovedsagelig størrelsen af vores korpus som er meget mindre end lignende træningskorpora. Ved at inkludere den observation i clustering at ord som optræder i forskellige typer af lister ofte også er relaterede til hinanden, har vi opnået bedre resultater for nogle af domænets indholdsord.

Vi har sammenlignet data i de bedste klynger med den manuelle klassifikation af samme data i den domænespecifikke ontologi. Sammenligningen viste at de fleste ord i de automatisk uddragede klynger er semantisk relaterede, men at ordenes indbyrdes relation inden for samme klynge ikke er af samme type. Dette resultat bekræftes af kørslerne af clustering med Infomap-systemet fra Stanford Universitet på enkelte engelske ord, som er oversættelser af nogle af de samme ord vi har analyseret tidligere. Vi uddrog clusteringsresultater fra Infomap, hvor træningskorpora var henholdsvis **British National Corpus** og **Wall Street Journal**.

Generelt kan vi konkludere at clusteringsteknikker kan støtte ontologiopbyggere

med at foreslå en første grov klassifikation af semantisk relaterede ord i domæner beskrevet af store mængder tekster. De opnåede klynger i denne klassifikation er dog af varierende kvalitet og kræver videre manuel bearbejdning.

Vores test viser også at clusteringsteknikkernes resultater vil kunne forbedres hvis de kunne udnytte lingvistiske oplysninger som fx ordenes valens.

Litteratur

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L. & Roossin, P. S. (1990), 'A statistical approach to machine translation', *Computational Linguistics* **16**, 79–85.

Buitelar, P., Olejnik, D., Hutanu, M., Schutz, A., Declerck, T. & Sintek, M. (2004), Towards ontology engineering based on linguistic analysis, *in* 'Proceedings of LREC-2004', Lisboa, Portugal, pp. 7–10.

Church, K. W. (1988), A stochastic parts program and noun phrase parser for unrestricted text, *in* 'Proceedings of ANLP 2', pp. 136–143.

G.A.Miller & W.G.Charles (1991), 'Contextual correlates of semantic similarity', *Language and Cognitive Processes* pp. 1–28.

Jelinek, F. (1990), Self-organized language modeling for speech recognition, *in* A. Waibel & K.-F. Lee, eds, 'Reedings in Speech Recognition', Morgan Kaufmann, CA, pp. 450–506.

Jongejan, B., Pedersen, B. S. & Navarretta, C. (2004), Automatisk analyse af zaccos og ankiros materiale, VID-rapport 3, Center for Sprogteknologi.

MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *in* L. L. Cam & J. Neyman, eds, 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability', University of California Press, Berkeley California, pp. 281–297.

Manning, C. D. & Schütze, H. (1999), *Foundations of Statistical Natural Language Processing*, The MIT Press.

Markov, A. A. (1913), An example of statistical investigation in the text of 'eugene onyegin' illustrating couples of 'tests' in chain, *in* 'Proceedings of the Academy of Sciences', Vol. 7, St. Petersburg, pp. 153–162.

Navarretta, C., Pedersen, B. S. & Hansen, D. H. (2004), Human language technology elements in a knowledge organisation system -the vid project., *in* 'Proceedings of LREC-2004', Vol. 1, pp. 75–78.

Pedersen, B. S., Navarretta, C. & Henriksen, L. (2004), Building business ontologies with language technology techniques - the vid project, *in* 'Proceeding of ONTOLEX Workshop in conjunction with LREC 2004', pp. 30–35.