TEKNISKA HÖGSKOLAN
Avdelningen för datateknik

Generering av naturligt språk från emergenta representationer

Oskar Kohonen

Diplomarbete som inlämnats för granskning som lärdomsprov för avläggande av diplomingenjörsexamen

Esbo, 31.5.2005

Övervakare: Prof. Timo Honkela
Handledare: Prof. Timo Honkela

Tekniska Högskolan
Avdelningen för datateknik

Sammandrag av diplomarbetet

Författare	Datum
	31.5.2005
Oskar Kohonen	Sidoantal
	75
Rubrik	
Generering av naturligt språk från emergenta representationer	
Professur	Kod
Informationsteknik	**T-61**
Övervakare	
Prof. Timo Honkela	
Handledare	
Prof. Timo Honkela	

När man tillämpar ostyrd inlärning på stora mängder naturligt språk (t.ex. texter på svenska eller engelska), framträder ofta representationer som kategoriserar de analyserade språkenheterna (t.ex. ord, eller morfem) på ett sätt som återspeglar intuitiva kategoriseringar gjorda av människor. Eftersom representationerna både framträder direkt ur språkdata och dessutom verkar vara kognitivt relevanta, undersöker vi möjligheten att tillämpa dem på kognitiva uppgifter, specifikt språkgenerering. Vi undersöker specifikt emergenta strukturer som framträder genom tillämpning av den sjävorganiserande kartan (SOM) och analys av oberoende komponenter (ICA). Vi undersöker hur de emergenta representationerna kan utnyttjas inom ramen för olika lingvistiska teorier och gör en litteraturundersökning om språkgenereringsmetoder, samt evaluerar möjligheten att använda de emergenta representationerna som utgångspunkt för generering av texter med olika stilistiska och semantiska egenskaper. Vi studerar också förhållandet mellan ICA komponenter som estimerats för ord i ett korpus och samma ords roll i den kontextfria grammatik som genererar korpuset.

Nyckelord
analys av oberoende komponenter, sjävorganiserande kartor, statistisk behandling av naturligt språk, generering av naturligt språk, stilanalys

Helsinki University of Technology
Department of Computer Science and Engineering

Abstract of Master's Thesis

Author	Date
Oskar Kohonen	**31.5.2005**
	Pages **75**
Title of the thesis **Natural Language Generation using Emergent Representations**	
Professorship **Computer and Information Science**	Professorship Code **T-61**
Supervisor **Prof. Timo Honkela**	
Instructor **Prof. Timo Honkela**	

When unsupervised learning methods are applied to large corpora of natural language texts (in languages such as e.g. English or Swedish), such representations often emerge that categorize the analyzed language units (e.g. words or morphemes) in a way that bears similarity with intuitive human made categorizations. Since these representations both arise directly from data and seem cognitively relevant, we investigate their usefulness in performing cognitive tasks, specifically in natural language generation. We investigate the structures that emerge from the application of Self-Organizing Maps (SOM) and Independent Component Analysis (ICA). We evaluate how emergent representations can be used within the framework of different linguistic theories, perform a literature review of natural language generation methods and evaluate emergent representation for generating texts with different stylistic and semantic properties. We also study the relationship between independent components estimated for words in a corpus and the roles of those words in the context-free grammar that generates the corpus.

Keywords
independent component analysis, self-organizing maps, statistical natural language processing, natural language generation, style analysis

Förord

Detta arbete har gjorts på laboratoriet för informationsteknik vid Tekniska Högskolan. Jag vill tacka min handledare, prof Timo Honkela. Jag vill också tacka Jaakko Väyrynen för hjälp med formatteringen av texten, samt Tiina Lindh-Knuutila och Matti Pöllä i cog-forskningsgruppen för att de gjort tiden då detta arbete skrevs betydligt roligare än den annars varit. Tack även till Mathias Creutz för feedback och korrigeringar.

Sist men inte minst vill jag tacka min fru, Marie, för att hon stått ut med mig då jag bara läst en massa konstiga böcker i all oändlighet. Tack för ditt stöd.

Innehåll

1 Inledning

1.1 Bakgrund

Människor talar. Somliga mindre, andra kanske för mycket. Om någon människa inte talar alls menar man vanligen att det är något som inte riktigt är som det ska vara. Med andra ord anses det ytterst naturligt för människor att kommunicera genom att tala. När vi kommer upp i sjuårsåldern får vi lära oss läsa. Det vi läser är samma språk som vi talar men det består inte av ljud utan av symboler på ett papper. Först är detta svårt för oss, det tar flera år innan det känns lika naturligt att läsa som att lyssna på när andra talar. För att kunna läsa måste man lära sig vilka bokstäver som motsvarar vilka ljud. Vanligen finns det mindre än 30 bokstäver. Sedan måste man lära sig stava också, men med 10 000 ord i sin vokabulär börjar man redan kunna stava det mesta.

Men om det är svårt att lära sig läsa, hur svårt är det då att lära sig att prata och förstå från första början? Vi minns inte hur det kändes att försöka göra sig förstådd när vi började tala i 1,5-2 års åldern. Därför vet vi inte heller hur svårt det kändes att lära sig tala.

För att försöka förstå hur svårt det är att göra något en människa gör, kan man försöka göra en maskin som gör samma sak. Som den kända fysikern Richard Feynman sade: “Det jag inte kan skapa, förstår jag inte”. Hur svårt är det då att göra en talande maskin? Att göra en maskin som räknar ihop 2 + 4 klarar vi utmärkt av, räknemaskiner finns överallt. Försöker vi göra en maskin som läser en text högt blir det betydligt svårare. De s.k. talsyntetisatorer som finns producerar tal som vi nog förstår men som inte låter naturligt. Försöker vi göra en maskin som skriver ner det som en människa säger blir det ännu svårare. Sådana finns, men de gör många fel.

Hur svårt är det då att göra en maskin som själv producerar språk? Antingen talar, eller skriver text? De flesta har sett science fiction-filmer med datorer som berättar hur de mår och som man kan diskutera med som med en människa. Sådana ser man inte idag trots att de garanterat kunde konkurrera med robothundar på marknaden för maskinella husdjur. Orsaken är att det inte i dagens läge är möjligt att göra en sådan maskin särskilt väl. För att förstå varför, måste man bekanta sig närmare med vad språket egentligen är.

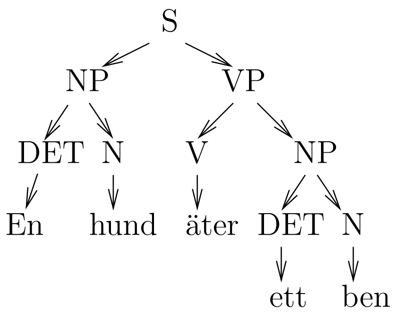
1.1.1 Vad är språk?

I första hand är språket det man talar. Det går inte att undvika att lära sig att tala i ett samhälle med andra människor. Däremot för att lära sig det skrivna språket måste man gå i skola och bli undervisad, det kommer inte naturligt som följd av att man är människa.

Det skrivna språket är en genialisk uppfinning som tillåter en människa att spara sina uttalanden för senare läsning eller för att andra ska kunna bekanta sig med dem. För att göra läsandet enklare att lära sig strävar man efter att skriftspråket ska likna det talade språket. Därför har skriftspråket många av de egenskaper som talat språk har. Detta kan leda till en illusion att talat och skrivet språk är “samma sak”.

Men det finns också vissa tydliga skillnader. Skrivet språk är i regel mer genomtänkt och mindre spontant. Detta beror mest på att den som läser inte kan höra med hurudant röstläge författaren skriver något, om han är ironisk, lättsam, glad eller betungad. Denna information som finns i det talade språket, men inte i det skrivna. Därför måste det skrivna språket att vara tydligare. Förutom detta har läsaren heller inte någon möjlighet att avbryta författaren och fråga “hur menar du nu riktigt?”, som han kunde i en muntlig diskussion. Allt läsaren kan göra är att läsa stycket en gång till och försöka förstå. All information som läsaren ska uppfatta måste alltså finnas i texten, medan det i ett muntligt samtal kan överföras också via gester och tonfall m.m.

När man forskar i språket bör man beakta skillnaderna mellan de olika språkformerna, eftersom



Figur 1: Syntaxträdet för meningen “En hund äter ett ben”

risken finns att man kommer till olika resultat om man bara betraktar någondera. Hur allvarlig skillnaden är beror på exakt *vad* man studerar angående språket. Vår ultimata målsättning är att hitta *regelbunden struktur* i språket.

Vi kommer främst att behandla skrivet språk. Inte för att det på något sätt skulle vara bättre eller “högre stående” som ibland påstås, utan av praktiska skäl. Det skrivna språket är lättare att processera maskinellt, och det är lättare att hitta stora mängder sparad text (t.ex. från WWW), än stora mängder inspelat tal. Dessutom verkar det rimligt att anta att resultat om strukturen i skrivet språk kan ge oss kunskap om strukturen i talat språk. Vi antar att likheten mellan de två är så stora att resultat från den ena kan ge hypoteser om den andra.

Genom att göra valet att hålla sig till skrivet språk skyddar vi oss från många komplikationer som finns i det talade språket, som t.ex. betoningar, pauser, olika långa uttal o.s.v. Sådana gör det svårare att hitta regelbunden struktur i språket.

1.1.2 Språkvetenskapens historia

Från ca 1920 till 1960 dominerades språkvetenskapen av något som kallas *empirism* (eng. empiricism). Man antar att människans språkliga förmåga är en följd av att människans mentala mekanismer tillämpas på språk som hon hör. Därför studerar man språket som det används, istället för att definiera något idealiskt språk. Under denna tid studerade Zellig Harris metoder som skulle upptäcka struktur i språket [Harris, 1951].

På 1960-talet blev *rationalismen* dominerande. Rationalister menar att det mesta av människans mentala förmåga finns genetiskt i människan vid födseln. Som följd började man studera hurudana apparater som kunde producera korrekta meningar ([Manning och Schütze, 1999] s. 4-7). Noam Chomsky formulerade sin mycket inflytelserika språkvetenskapliga teori som baserar sig på att studera hurudana formella grammatiker som kan producera enbart korrekta meningar [Chomsky, 1957]. Denna koppling till matematiken gör att Chomskys teorier är möjliga att förverkliga automatiskt som datorprogram. Till skillnad från Harris’ arbete som beskrev strukturen i språket, utgjorde Chomskys grammatiker regler som *genererar* språket. Argumentet som motiverar ändringen är, enligt Chomsky, något som kunde översättas till svenska ungefär som “brist på stimuli” (eng. “poverty of the stimulus”). Chomsky menar, att ett barn uppfattar en så liten del av de meningar som kan sägas att hon omöjligt kan lära sig språket från dem. Därför måste språkets maskineri vara medfött [Chomsky, 1986].

Chomskys syn på ett språks syntax, d.v.s. meningsstruktur, känd som den *generativa grammatiken*, har visat sig mycket framgångsrik och tillämpats förutom inom språkvetenskapen också inom datatekniken för t.ex. programmeringsspråk. Språkets syntax beskrivs med hjälp av en matematisk formalism där olika symboler *genererar* andra symboler. En enkel grammatik är t.ex.:

S ::= NP VP .
NP ::= DET N
VP ::= V NP
DET ::= en | ett
N ::= hund | ben | djur | växt
V ::= äter

Den kan generera meningar som:
"En hund äter ett ben"
"Ett djur äter en växt"

Hur ska man läsa grammatiken? Det finns två sorters symboler: *Nonterminalsymboler* och *Terminalsymboler*. Terminalsymbolerna är orden som finns i det språket, nonterminalsymbolerna däremot är "variabler", motsvarande kategorier av ord, som byts ut enligt reglerna i grammatiken tills bara terminalsymboler återstår. Man börjar alltid med startsymbolen S och tillämpar någon av grammatikens regler på någon av nonterminalsymbolerna. Så här härleder man meningen "en hund äter ett ben" från grammatiken.

S → NP VP
S → DET N VP
S → en N VP
S → en hund VP
S → en hund V NP
S → en hund äter NP
S → en hund äter DET N
S → en hund äter ett N
S → en hund äter ett ben

Man kan också framställa det grafiskt som ett syntaxträd (fig 1). Förutom denna sorts regler för frasstruktur finns även transformationsregler som t.ex. transformerar mellan aktiv och passiv form. Dessa regler är också formulerade som symbolmanipulation. Man kan summera det hela med att säga att språket beskrivs av *symboler* som transformeras till annorlunda symboler. En sådan grammatik kommer vi att kalla *symbolisk grammatik*.

1.1.3 Vad är intelligens?

Parallellt med utvecklingen inom språkvetenskapen skedde också motsvarande utveckling inom datatekniken. Under 1950-talet framträdde forskningsområdet för *artificiell intelligens*. *Artificiell* betyder konstgjord. I detta fall menas specifikt något man själv kan tillverka. Artificiell intelligens studerar automatiska system som utför intelligent beteende [Luger och Stubblefield, 1994].

Men vad är *intelligens*? Det är en betydligt svårare fråga att svara på. T.ex. blommor växer mot solen, är det intelligent beteende eller inte? Brevduvor hittar hem, hästar lär sig lyda. Det verkar ju kräva intelligens, eller hur? Men flyger brevduvan hem för att den är intelligent eller bara för att den bara fungerar så? Kan duvan välja att göra något annat? Och människan då? Människan är *medveten* om sig själv och sina beslut, är det därför hon är intelligent? Den artificiella intelligensen kan inte besvara dessa frågor entydigt, och området studerar problem som anses kräva intelligens att lösa. Sätten att lösa dessa problem är främst den matematiska logiken.

Logik är en matematisk formalism där man räknar med två olika värden: Sant och falskt. Gottlob Frege utvecklade redan på 1800-talet en formalism som kunde användas för att resonera med dessa sanningsvärden: *första gradens predikatkalkyl* [Frege, 1879]. Påståenden kan i predikatlogiken uttryckas med symboler. Ofta antar man att symbolerna hänvisar till något som finns i den yttre världen. Denna referensteori är grunden för mycket av formell semantik, d.v.s. läran om meningars betydelser [Tarski, 1944].

Fördelen med att använda symboler är att man inte behöver veta vad symbolerna betyder för att kunna härleda någonting. Det enda som spelar någon roll är hur symbolerna förhåller sig till varanda. Vi kan t.ex. uttrycka Aristoteles härledning:

Alla människor är dödliga. Sokrates är en människa.

Alltså: Sokrates är dödlig.

Uttryckt i predikatkalkyl:

$\forall x$Människa(x) $\Rightarrow$ Dödlig(x)) (1)
Människa(Sokrates) (2)

De logiska formlerna ska läsas
(1) För vilken som helst symbol x gäller att om vi vet att x är en människa följer att x också är dödlig.
(2) Sokrates är en människa.

Då finns det allmänna regler att härleda med som t.ex. Aristoteles' modus-ponens $\frac{F,F=>G}{G}$. Regeln bör läsas: Om vi vet att påståendet F är sant och vi vet att ur F följer G, så kan vi dra slutsatsen att G också är sant. Vi tillämpar det på de två påståendena ovan, vi börjar med att skriva om (1) genom att ersätta x med Sokrates, vilket vi kan göra för regeln gäller alla x:

Människa(Sokrates) $\Rightarrow$ Dödlig(Sokrates)

Vi sätter in detta i modus-ponens regeln tillsammans med (2):

Människa(Sokrates), Människa(Sokrates) $\Rightarrow$ Dödlig(Sokrates)

Då får vi:

Dödlig(Sokrates)

Med andra ord kommer vi fram till att Sokrates är dödlig. Eftersom "resonerandet" görs mekaniskt med regler, kan man utmärkt låta en dator göra det. På det här sättet har vi alltså gjort ett automatiskt "förnuft". Men allt som räknar med logiska formler är inte intelligent. Om det var så skulle t.ex. trafikljus kunna anses intelligenta, vi behöver fortfarande ett bättre sätt att definiera intelligens.

Alan Turing, en av datateknikens pionjärer definierade intelligensen på ett sätt som är typiskt matematiker. Han hittade på ett sätt att testa om något var intelligent eller inte. Turing utgick från att människan är intelligent. För det s.k. Turing-testet antar man att två parter kan samtala med varann genom att skriva text. T.ex. så att två personer sitter i var sitt rum och utan att se den andre kommunicerar de genom att skriva varandra meddelanden. Om man kan ersätta den ene personen med en maskin, så att människor som kommunicerar med maskinen inte vet om att de skriver till en maskin, och inte en människa, då anses maskinen intelligent [Turing, 1950].

Detta test mäter givetvis bara språklig intelligens, och vad intelligens egentligen är debatteras fortfarande flitigt. Inga slutgiltiga svar på den frågan verkar heller vara inom synhåll.

Turing-testet har den fördelen att dess språkliga inriktining har lett till en hel del intressant forskning. Ett exempel på sådan forskning är ett "pratsamt" dataprogram, Joseph Weizenbaums

ELIZA. ELIZA är en automatisk psykiater som svarar på ens frågor genom att använda logiska regler för hur svaren ska ges [Weizenbaum, 1966].

Exempelkonversation med en ELIZA (från http://www-ai.ijs.si/eliza-cgi-bin/eliza_script): You: I have a problem writing Eliza: Can you elaborate on that? You: I don't write fast enough Eliza: Why do you not write fast enough? You: I'm too slow Eliza: Does it please you to believe I am too slow? You: No, it's not you Eliza: You are being a bit negative. You: Yes I know Eliza: You are sure? Man kan se att ELIZA är ganska bra, men nog gör sådana fel en människa aldrig skulle göra.

Trots att det är svårt att exakt säga vad intelligens är, verkar det som om man för att kunna behandla språk behöver intelligens, åtminstone någon del av det vi kallar intelligens.

Forskningen inom artificiell intelligens har kombinerat symbolisk logik med Chomskys generativa grammatik. Ibland har detta lett till stora succéer, t.ex. i fallet av programmeringsspråk, alltså språk som utvecklats för att programmera datorer. Men i fallet av *naturligt språk*, sådant som människor talar, har framgångarna på detta område inte varit lika strålande. Ett dataprogram förstår fortsättningsvis väldigt lite av naturligt språk. De grammatiker som man konstruerar för att tolka naturligt språk är mycket komplicerade eftersom naturligt språk är så mångtydigt och har så otroligt många former.

Men kanske man kan göra en maskin som lär sig grammatiken från exempel, såsom människan lär sig språket. Hur lär man sig en formell grammatik?

1.1.4 Inlärning av språk

Enligt Chomskys teori har varje barn redan medfött förmågan att förstå och tala språk, i allmänhet, d.v.s. alla har en språkinlärningsapparat. Denna apparat känner till språkets struktur i allmänhet och hittar de parametrar som gäller specifikt för den grammatik som barnet hör.

Men hur hittar barnet dessa parametrar? Grammatiken består av helt abstrakta symboler som subjekt och predikat, verb och substantiv och dylikt. Hur kan man hitta sådana bara genom att höra språk? Stephen Pinker föreslår att barnet använder semantiken som utgångspunkt [Pinker, 1984]. Semantiken är ordens och frasernas betydelser. Pinker föreslår att barnet märker att föräldrarna använder vissa ord när de talar om saker och andra ord när de talar om händelser, eller något man gör. Dessa grupper av ord blir sedan grunden för substantiv- och verbsymbolerna i grammatiken. Senare fylls dessa på genom att iaktta vilka andra ord som beter sig lika i språket, alltså förekommer i liknande sammanhang. På det sättet kommer man fram till det vuxna stadiet där kategorierna är abstrakta och inte har så mycket med ordens betydelse att göra. T.ex. "händelse" och "äpple" är båda substantiv, men man kan inte direkt se något stort samband mellan deras betydelser. Att de är substantiv märker man på att man kan lägga en obestämd artikel före "en händelse" och "ett äpple". Kategorin substantiv är alltså mer bestämd av ordens sammanhang och roller i grammatiken än deras betydelse.

Nu har vi alltså en teori både för hur språket byggs upp med hjälp av logiska regler, och hur man kan göra logisk slutledning, så då har vi både maskinellt "språk" och "förnuft"! Då borde det bara vara att tillämpa dessa principer och vi har en intelligent maskin som kan lära sig språk.

Men det visar sig inte riktigt vara så enkelt. Chomsky klassificerade språken i en hierarki enligt hur komplicerad grammatik språket har [Chomsky, 1956]. Denna klassificering är baserad på formella egenskaper hos grammatikerna av den sorten vi presenterade tidigare. Mark Gold visade [Gold, 1967] att det inte går att lära sig grammatiken för ett kontextberoende språk (eng. context-sensitive) om man inte vet mera än bara en ändlig mängd exempelmeningar ur språket och saknar exempel på meningar som är felaktiga. Beviset är formellt till sin natur, men man kan uttrycka det som så att det är omöjligt att lära sig att det är fel att säga "ätade" (borde vara "åt") om man inte får veta att "ätade" är fel. Men det får man ju inte, för exempelmeningarna kan inte innehålla alla olika former, för språket är ju oändligt. Då går det så att "Ätade" inte finns bland exempelmeningarna, men också många andra, korrekta men ovanliga, former som "båtade" saknas. Eftersom den engelska grammatiken anses vara kontextberoende, betyder det att den inte går att lära sig, om man inte vet också vilka meningar som är fel.

Men vi vet att barn lär sig att "prata rätt", och enligt Gold går det till utan att barnens föräldrar korrigerar deras felaktiga meningar. T.o.m. om föräldrarna försöker, så påverkar det inte barnens språkliga beteende. Då måste det vara så att barnet har mer kunskap om språket än bara att grammatiken är kontextberoende. Detta, beroende på hur man ser på saken, antingen bekräftar Chomskys tanke om en språkinlärningsapparat, eller är ett argument mot tanken att grammatiken verkligen skulle vara en kontextberoende symbolisk grammatik.

Men enligt Michael Tomasello empiriska undersökningar verkar det inte alls vara så att barn använder sig av abstrakta kategorier, inte semantiska och inte syntaktiska. Istället tycks varje ord utvecklas ensamt, utan att påverkas av andra ord. När han studerade sin dotters språkliga utveckling var det inte så att alla verb på en gång fick sina olika former, utan varje verb tycktes utveckla sina former, vart och ett för sig. Ytterligare studier, med 12 engelsktalande barn på 2-3 år, visade att barnen använde praktiskt taget alla verb i enbart en enda meningskonstruktion. Varje verb hade alltså en egen mening i vilken verbet användes, och utanför den meningen användes verbet inte alls. Motsvarande gällde när barnen började använda artiklarna "a" och "the". De använde dem för olika substantiv. Det fanns så gott som inga substantiv för vilka barnen använde både "a" och "the" vid den här åldern. Det tyder på att barnen inte använder sig av några abstrakta kategorier för substantiv, för annars skulle de upptäcka att "a" och "the" passar framför alla substantiv. Motsvarande resultat har också bekräftats för italienska, brasiliansk portugisiska och hebreiska så det gäller inte enbart engelsktalande [Tomasello, 2000].

Elizabeth Bates, en annan språkinlärnings- och hjärnforskare angriper Chomskys teori om en språkinlärningsapparat i hjärnan. Enligt hennes resultat är visserligen hjärnans struktur ganska lika hos vuxna individer, men inte hos barn. Om en vuxen människa får en skada på det område i hjärnan där mycket aktivitet observerats under språkliga uppgifter, blir hon oförmögen att tala (får en eller annan form av *afasi*). Om det däremot händer ett barn vid tillräckligt ung ålder, lär sig barnet ändå språket, bara några få procent sämre än alldeles friska barn [Bates, 1999]. Detta beror på att språkets funktionalitet kan flytta sig till andra delar av hjärnan vid ung ålder, men inte längre vid vuxen ålder. Då kan det inte finnas någon speciell del av hjärnan som är medfödd för språket, utan språket måste använda sig av hjärnans generella inlärningsmekanismer.

Ytterligare ett argument mot Chomskys språkteorier presenteras av David M. Powers som analyserar Chomskys axiom angående språket, och kommer fram till att de är sådana till sin natur att de gör det väldigt svårt att skapa mekaniska inlärningsmetoder för språket [Powers, 1996]. Men när nu språket ändå går att lära sig, varför skulle det då vara sådant till sin natur att det inte skulle gå att lära sig?

Dessa resultat talar emot teorin att språklig och grammatikalisk kunskap skulle basera sig på medfödd förmåga att hantera symboler. Men om man förkastar den tanken vad har man då för alternativ?

Följden av tanken att språkförmågan inte är medfödd är att den måste vara inlärd med allmänna inlärningsmekanismer. Detta betyder att *empirismen* gör comeback, vilket den gjort på senaste tiden. Då blir en intressant fråga hur man egentligen lär sig språk och vilka källor man använder. Enligt Chomsky är språkförmåga något som en individ har, men man kan också betrakta det som något ett samhälle besitter. Ett barn som föds in i samhället lär sig det språk som dess föräldrar, och senare dess kompisar, talar. De har i sin tur lärt sig språket från andra som lärt sig från andra o.s.v. Vi vet också att den svenska vi talar idag inte är densamma som talades för 100 år sedan, eftersom språket ändras både efter kulturen, men också åtminstone till synes slumpmässigt. Betyder det att språket med tiden kan ändras hur som helst? Nej, för språkets form begränsas av det faktum att människor måste lyckas lära sig det för att kunna föra det vidare till sina barn [Zuidema, 2003]. Därför anpassar sig språket inte bara efter kulturen utan också enligt människans inlärningsmekanismer. Språkets form är alltså begränsad, men inte enbart av teoretiska modeller utan av människans egna egenskaper och lärande.

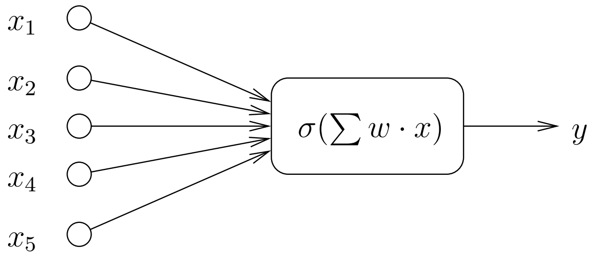
Vi har alltså orsak att misstänka att den generativa språkvetenskapen inte är korrekt, åtminstone inte i detalj. Människans inlärningsmetoder är centrala på grund av de sociala mekanismerna som för språket vidare. Vi betraktar därför språkforskning kombinerat med forskning i maskinella inlärningsmetoder som intressant och är öppna för andra tankar om språket än de generativa teorierna. Alternativet till symboliska grammatiker är probabilistiska grammatiker, d.v.s. strukturen beskrivs med hjälp av sannolikheter att en viss mening förekommer. Med empirismens återkomst har också dessa statistiska språkprocesseringsmetoder blivit mer intressanta. Datorernas stora framsteg sedan 60-talet, gör statistiska metoder och automatisk analys tillgängliga för en bredare publik än någonsin förut. Vi bör därför noga observera kombinationen av maskininlärning och språkvetenskap.

1.2 Maskiner som lär sig

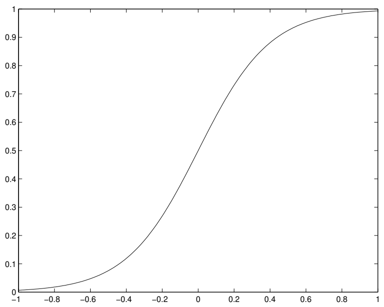
Vad betyder det att lära sig? För en människa betyder det vanligen att man först inte kunde göra något, men sedan lärde man sig och efteråt kan man göra det man tidigare inte kunde. Det är alltså frågan om att anpassa sitt beteende efter sina observationer. Men vad menar vi med att en maskin lär sig? En dator lär sig vanligen inte att gå eller utföra något uppdrag, enligt exempel. Hur kan vi då jämföra lärandet? En tanke är att lärandet händer i människans hjärna och datorn kan härma det som händer i människans hjärna. De symboliska systemen för artificiell intelligens, som nämndes tidigare, härmar den matematisk-logiska slutledningsprocessen. Om man däremot försöker härma sättet som hjärnan fungerar, kommer man in på det som kallas för konnektionism, eller neuralnät. Neuralnäten är kopplade till omvärlden genom inputs och outputs. För neuralnät definieras inlärning som modifikationen av nätverkets parametrar som följd av dess interaktion med omvärlden (från [Haykin, 1999] s.50). Denna definition betyder i praktiken att neuralnätet ändras så att dess prestanda i en viss uppgift förbättras som följd av inlärandet. Detta är rätt nära en intuitiv uppfattning om inlärning.

Konnektionismen strävar till att efterlikna delar av hjärnans funktion med konstgjorda nervceller. Människans hjärna är ett mycket komplicerat system. Den består av ca 100 miljarder nervceller, och är kapabel att kontrollera kroppens mycket komplicerade funktioner, t.ex. att springa, tala, läsa m.m. Detta komplicerade system består av små och, relativt sett, okomplicerade delar, *nervceller* eller *neuroner* som de också kallas. Konstgjorda neuralnät består av konstgjorda nervceller som kopplats samman till ett nät.

För att skapa, eller åtminstone simulera, konstgjorda nervceller måste man ha en *modell* av vad som pågår i de biologiska nervcellerna. Vi vet att i dem sker komplicerade elektriska och kemiska processer. Därför använder man sig vanligen av en kraftigt förenklad modell när man simulerar dem.



Figur 2: Modellen av en nervcell i MLP-nätet



Figur 3: Sigmoidfunktionen som ofta används som aktivationsfunktion

Det finns givetvis flera olika sorters modeller, men eftersom de liknar varann kan beskrivningen av en modell hjälpa att förstå också de andra. Här är beskrivningen av modellen som används i MLP-nät (Multilayer Perceptron).

Neuronerna består av en kärna, en eller flera inputs och en eller flera outputs (se figur 2). Inputarna har olika *vikter* efter hur mycket en viss input betonas och i kärnan finns en *aktivationsfunktion* som räknar ut outputvärdet på basen av inputvärdet. På bilden ser man en neuron med fem inputs, x_1, x_2, x_3, x_4 och x_5. Åt aktivationsfunktionen ger man en vägd summa av input och vikter:

$$v = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 + w_5 * x_5 \tag{1}$$

Aktivationsfunktionen är vanligen icke-linjär för att neuralnätet ska kunna räkna funktioner som inte är linjära och sådan att den "plattar till" den input den får. En vanlig aktivationsfunktion är *sigmoidfunktionen* (se figur 3)

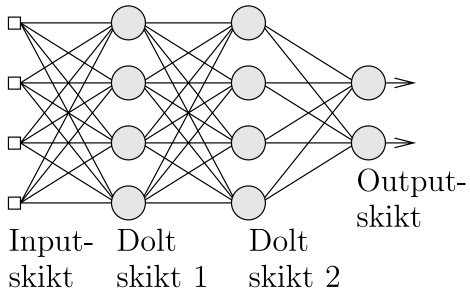
$$P(v) = \frac{1}{1 + e{-}av} \tag{2}$$

Ett matematiskt praktiskt sätt att betrakta neuralnätet är att se x och w som vektorer. Vektorers inre produkt är definierad:

$$\mathbf{w} \cdot \mathbf{x} = w_1 * x_1 + w_2 * x_2 + ... + w_{n-1} * x_{n-1} + w_n * x_n \tag{3}$$

Då kan vi skriva outputfunktionen för en neuron med n inputs med hjälp av inre produkten:

$$y = \sigma(\sum \mathbf{w} \cdot \mathbf{x}). \tag{4}$$



Figur 4: Ett MLP-nät med tre skikt, två dolda skikt och ett outputskikt. De grå cirklarna är neuroner som i figur 2 och linjerna mellan dem är kopplingar mellan neuronerna.

Tanken är den att signalen som kommer in i nätets input motsvarar nervsignaler t.ex. från sinnena. I det biologiska nervsystemet skjuter nervcellerna upprepade pulser, i MLP motsvarar siffervärdet på en viss input frekvensen med vilken nervcellen pulserar. Denna förenkling betyder att man lämnar bort all tidsberoende information från simulationen och bara tar i betraktande nätets statiska egenskaper. Det finns andra neuralnätsarkitekturer som också försöker ta tidsperspektivet i beaktande (se t.ex. [Elman, 1990, Maass och (eds.), 1998]).

En viktig sak att notera är att till skillnad från de symboliska metoderna, som räknade med värdena sant och falskt, räknar neuralnät med vektorer av reella tal. Därför säger man att de är *vektorrymdsmodeller*.

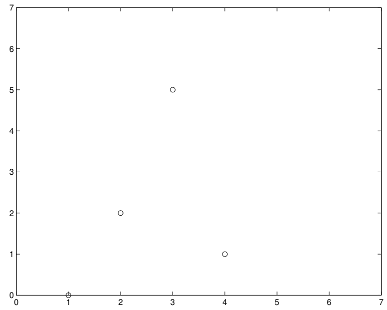
Men bara enskilda neuroner räcker inte, de måste kopplas ihop också. Därför organiserar man neuronerna i flera skikt (se figur 4), så att neuronerna är kopplade med en neurons output i ett skikt till input för neuronerna i nästa skikt. Först kommer det s.k. inputskiktet där man matar in den inputvektor man vill räkna med, sedan kommer ett varierande antal dolda skikt. De kallas så därför att deras aktivitet inte syns utifrån, utan de är bara mellan inputskiktet och det slutliga outputskiktet som ger ut resultat som nätet räknat. De dolda skikten finns med för de tillför nätet förmåga att räkna ut mer komplicerade funktioner.

Man kan se att MLP-nätet får in vektorer och ut kommer vektorer. De motsvarar alltså en vektorvärd funktion av en vektor i matematiken med input och output. När MLP-nätet lär sig så anpassas dess vikter så att funktionen nätet räknar anpassar sig efter den inlärningsdata man använder.

Hur får man den att göra det? MLP lär sig genom s.k. *styrd inlärning* (eng. supervised learning). Det betyder att man lär nätet genom att mata en mängd data för vilken man vet det rätta svaret. Man ser vad nätet räknar ut, jämför med det rätta svaret och en *inlärningsalgoritm* ändrar vikterna i nätet på ett sådant sätt att felet blir mindre för varje exempel. När man gör detta tillräckligt länge lär sig nätet att räkna ut den önskade funktionen.

Men vad är det för idé med att lära sig något som man redan visste det rätta svaret för? Jo nätet kan *generalisera*. Om vi vet att $f(3) = 9$ och $f(5) = 25$, men inte vet vad $f(4)$ är så kan neuralnätet ge en gissning på vad $f(4)$ kunde vara. Man lär sig alltså en kontinuerlig funktion från ett ändlig sampel.

Vi kan tänka oss som exempelproblem att artklassificera blommor. Vi kan anställa en biolog som kan säga vilken art en blomma är, då han ser den. Men klassificeringen är ingen enkel sak. Biologen kan (antagligen) inte sätta sig ner och skriva ett dataprogram som skulle ta emot vissa mått på blomman och på basen av dem göra samma klassificering som han. Hans förmåga är inte tillgänglig för honom så att han formulera *hur* han gör. Då kan det vara nyttigt att låta biologen klassificera några blommor och mäta vissa parametrar, t.ex. kronbladens storlek, antal och form,



Figur 5: Punkterna (1,0), (3,5), (2,2) och (4,1) i en tvådimensionell vektorrymd

från samma blommor. Sedan låter man neuralnätet lära sig hur klassifikationen kan räknas ut från parametrarna. Då utgör neuralnätet ett program som räknar ut klassificeringen, inte fullkomligt, men med någon noggranhet. Man kan också granska vilka parametrar neuralnätet använder och lära sig vilka parametrar som kunde vara bra tecken för en viss klass.

MLP-näten är uppenbart inspirerade av hjärnans funktion. Senare har man också löst liknande problem med maskininlärningsmetoder som är baserade på enbart statistiska teorier o.d. Numera är det vanligare att man baserar sina metoder på matematiska kriterier än på likhet med biologiska system (t.ex. SVM [Vapnik, 1998]).

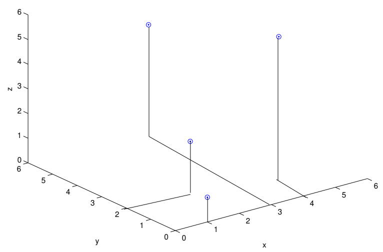
Styrd inlärning är inte heller den enda formen av maskininlärning. Det finns också något som kallas *ostyrd inlärning* (eng. unsupervised learning). Det betyder att man lär sig från data, utan att ha några rätta svar till hands. För ostyrd inlärning behöver man en modell som beskriver hurudan struktur i datan man är intresserad av, och hurudan struktur som är ointressant. Modellen är ofta sådan att saker som liknar varandra före inlärningen (enligt något matematiskt kriterium) ska likna varann också efter inlärningen, men ytliga skillnader ska falla bort så att det är lättare att märka de verkliga likheterna och skillnaderna. Andra tänkbara modeller är sådana att man ska kunna förvara mer data i mindre utrymme genom att hitta struktur i datan. Generellt försöker ostyrd inlärning hitta struktur i data, inte lära sig en funktion från input till output. I detta arbete är vårt fokus vid ostyrd inlärning.

Trots olikheter i ursprung och funktion finns en fundamental likhet i så gott som all maskininlärning: Man representerar data som vektorer och vektorrymder, inte symboler eller annat sådant.

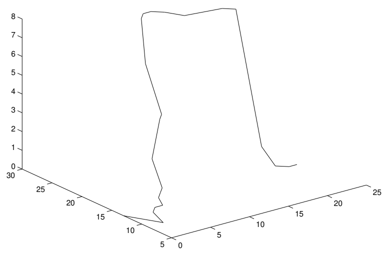
1.2.1 Vektorrymdsmodeller

Ordet *vektorrymdsmodeller* låter ungefär som individer som fotograferas för reklambilder i sämre science-fictionlitteratur. I praktiken rör det sig om något mycket mer vardagsnära. Det enklaste exemplet är xy-planet i matematiken, en tvådimensionell vektorrymd. Alla punkter i xy-planet kan utttyckas med hjälp av två koordinater: x- och y-koordinaten, se t.ex. figur 5 med punkterna (1,0), (3,5), (2,2) och (4,1). Samma idé kan utvidgas till rummet med xyz-koordinater, som i 6 med punkterna (1,0,1), (3,5,5), (2,2,3) och (4,1,6).

Men koordinater är inte det enda vi kan beskriva med vektorer. Om vi har en maskin som har en klocka, en termometer, en vindmätare kan vi också skriva det som en vektor: (12, 25, 5), betyder 25℃ varmt, 5 m/s vind, klockan 12. Då kan vi visualisera hur temperaturen varierar som funktion



Figur 6: Punkterna (1,0,1), (3,5,5), (2,2,3) och (4,1,6) i en tredimensionell vektorrymd. Linjerna visar förhållandet till axlarna.



Figur 7: Tiden, temperaturen och vindhastigheten

av tiden, se figur 7 för en (påhittad) dag. Vi ser hur både temperaturen och vinden ökar under dagen och sedan igen lägger sig mot natten. Men tänk om vår maskin dessutom mätte luftfuktigheten. Då kunde den på samma sätt för varje tidpunkt mäta en vektor, t.ex. (12, 25, 5, 60), där 60 betyder 60% luftfuktighet. Men då kan vi inte längre rita en skojig graf med fyra dimensioner, det går inte att visualisera. Det trevliga med vektorer är att samma matematik som används för att räkna med två- eller tredimensionella vektorer fungerar också på vektorer av högre dimension.

I maskininlärning använder man sig av vektorrymdsmodeller som input. Det leder till att man kan lägga till inputs och ta bort inputs som det passar en utan att man behöver ändra på den matematiska formel man använder. I praktiken blir det dock svårare att lösa problem med flera dimensioner eftersom man måste räkna mer och det finns flera frihetsgrader.

All data man vill inkludera i sin analys måste alltså kodas som vektorer. Det gör att t.ex. text, som inte är i vektorform, måste representeras på något annat sätt än direkt som text. Ofta använder man t.ex. frekvenserna på ord omkring ett visst ord för att representera det ordet som vektor. Denna idé härstammar från den brittiska språkvetaren J.R.Firth som på 50-talet sade: "du känner ett ord på dess sällskap" (eng "You shall know a word by the company it keeps") ([Manning och Schütze, 1999] s. 4-7).

Valet av vektorrepresentation kan ofta vara en avgörande del av problemet när man försöker tillämpa maskininläning på något som inte naturligt är i vektorform. En bra representation leder till bra resultat, medan en sämre representation inte leder till ens vettiga resultat.

1.2.2 Neuralnätens historia

För att få en överblick av några centrala idéer i maskininlärning och deras utveckling finns här samlat några punkter ur neuralnätens historia (enligt [Haykin, 1999]). De moderna neuralnätens historia börjar på 1940-talet med Warren McCulloch och Walter Pitts. McCulloch var psykiatriker och neuroanatomiker. Han funderade i 20 år på hur information representeras i nervsystemet. Pitts i sin tur var en hejare på matematik. De båda deltog i en grupp vid University of Chicago som sedan 1938 sysslade med att göra modeller av nervsystemet. År 1943 gav de ut sin artikel om hur nervceller kunde arrangeras för att räkna olika logiska funktioner. Dessa neuroner var mycket lika de teoretiska maskiner som de samtida datateknikens pionjärer Alan Turing och John von Neumann utvecklade. Båda gruppernas maskiner använde sig av input och output som var binära, antingen 0 eller 1. Båda parterna var också inspirerade av varandras arbete. McCullogh och Pitts visade att med rätta kopplingar och rätta vikter kunde deras neuroner räkna alla de funktioner som Turings maskin (den s.k. *Turing-maskinen*) kunde. Det betyder att neuralnäten kan räkna ut allt som en dator teoretiskt kan räkna ut.

År 1949 gav Donald Hebb ut sin bok *The Organization of Behavior* [Hebb, 1949], där han som den förste föreslår en regel för inlärning i nervsystemet genom ändring av synaptiska vikter. Hebbs regel säger att styrkan i en koppling mellan två nerver blir starkare av fortsatt aktivering. D.v.s. om två nervceller ofta aktiveras samtidigt så kopplas de ännu starkare samman. Hebbs bok blev mycket inflytelserik bland psykologer, men kom inte till kännedom bland ingenjörer. De första försöken att simulera Hebbs inlärningsregel på dator gjordes 1956 och de visade att regeln nog fungerade, men bara efter viss modifikation.

1948 utkom Norbert Wieners verk *Cybernetics* [Wiener, 1948] som blev grunden för den moderna cybernetiken. Ofta associeras ordet cybernetik, delvis felaktigt, med robotik, eller cyborgs (varelser som är blandning av människa och maskin i science-fiction litteratur). Det som Wiener påbörjade är vetenskapen som studerar kommunikation och kontroll bland levande varelser, alltså hur levande varelser beter sig och kommunicerar. Perspektivet som tas är att jämföra det levande systemets funktion med ett konstgjort system. Wieners bok behandlar dessa punkter mot bakgrunden av de samtida framstegen inom kontroll- och kommunikationsteori samt statistisk signalbehandling.

Dessa framsteg och tankegångar gjorde att man var mycket optimistisk och intresserad av hur man kan skapa en konstgjord hjärna för att uppnå artificiell intelligens. Senare har forskningen för artificiell intelligens inriktat sig på symboliska metoder, och neuralnäten skiljt sig till en egen inriktning. Men i detta skede betraktades neuralnäten som en del av artificiell intelligens vilket illustreras bl.a. av Marvin Minskys doktorsavhandling om neuralnät, samt hans artikel "Steps Toward Artificial Intelligence" [Minsky, 1961] som innehöll till en stor del innehöll material som numera skulle klassificeras under neuralnät.

Ungefär 15 år efter McCulloch och Pitts artikel utvecklade Frank Rosenblatt en inlärningsalgoritm för McCullochs och Pitts' neuralnät som härefter kallas *perceptronen* [Rosenblatt, 1958]. Perceptronen liknar ett MLP-nät, men är mer begränsat. Den viktigaste skillnaden är att perceptroner bara har ett skikt, jämfört med MLP-nätets (Multilayer Perceptron) flera dolda skikt. Nu hade man en inlärningsalgoritm, så plötsligt verkade allt allting möjligt för neuralnäten!

Men så småningom började det visa sig att de antaganden man gjort inte alltid var giltiga. Perceptronerna var inte lika användbara som väntat. John von Neumann, som ibland kallas den digitala datorns fader för sina insatser i datateknikens utveckling, hade under sina sista år arbetat på ett manuskript som efter hans död 1957 gavs ut i bokform under namnet *The Computer and the Brain* 1958 [von Neumann, 1958]. Von Neumann påpekade fundamentala skillnader mellan datorn och hjärnan. Bland annat händer allt parallellt i hjärnan, medan det händer en sak i taget i datorn. Också kemiska aspekter av hjärnans funktion skiljer den rent mekaniskt från datorn.

År 1969 skrev Minsky och Papert en artikel där de visade att det nog var lite för mycket luft i neuralnätsballongen [Minsky och Papert, 1969]. Bland annat visade de matematiskt att det inte var möjligt för en perceptron att räkna ut den logiska funktionen XOR. Det beror på att Rosenblatts perceptron bara har ett skikt neuroner. Man hade försökt utvidga detta till flera skikt, men inte lyckats hitta någon inlärningsregel. Minsky och Papert menade att det inte var så sannolikt att perceptroner i flera skikt skulle lösa problemet, eftersom det i ett sådant nät är mycket svårt att vet *vilken* neuron man ska tacka eller skylla på när det går bra eller dåligt, och därför skulle det vara mycket svårt, om inte omöjligt, att hitta en inlärningsregel som fungerade.

Som resultat stannade neuralnätens utveckling upp under 70-talet. Den artificiella intelligensen som förr varit ett enhetligt fält med både symboliska system och neuralnät utvecklades mer åt det symboliska hållet. Utvecklingen av neuralnät saktade in och forskningen minskade. Trots detta utvecklades, under denna tid, den självorganiserande kartan, ett neuralnät som kan användas både för visualisering av stora datamängder och klustring av liknande datapunkter. Vi kommer att använda den självorganiserande kartan i några av våra experiment.

Efter 70-talets pessimism började nya framsteg göras under 80-talet 1986 lyckades Rumelhart och McClelland med det som Minsky och Papert hade försökt visa osannolikt, om inte omöjligt: De upptäckte en inlärningsregel för MLP-näten [Rumelhart och McClelland, 1986]. Ett MLP-nät är alltså en perceptron i många skikt . Denna nya neuralnätsarkitektur kunde bl.a. lösa det för perceptronen omöjliga XOR problemet. Senare upptäcktes att inlärningsregeln, kallad *backpropagation*, utvecklats ursprungligen på redan 70-talet av Paul Werbos [Werbos, 1974] i hans doktorsavhandling, men då hade den inte väckt någon större reaktion. Efter att Rumelhart och McClelland återupptäckt backpropagation fick det på nytt fart på neuralnätsforskningen. Man började allt mer upptäcka sambandet mellan neuralnät och statistiska optimeringsproblem i matematiken. Man började tillämpa Claude Shannons *informationsteori* [Shannon och Weaver, 1949] även inom neuralnätsforskningen. Det ledde till att man började intressera sig för något som kallas *Blind Source Separation*, att skilja på olika signaler som blandats ihop till en enda signal. Under 90-talet fördjupades förståelsen av sambandet mellan neuralnät och statistik genom Vladimir Vapniks Statistical Learning Theory [Vapnik, 1998].

1.2.3 Den självorganiserande kartan

Redan för ca 100 år sedan visste man att människans hjärnas olika delar utförde olika uppgifter. Man kunde veta det genom att observera hur olika sorters skador på hjärnan påverkade människors beteende. T.ex. om en tumör eller blodpropp skadade en viss del av hjärnan på en person, så kunde den personen inte tala tydligt, och om en annan del skadades kunde personen inte använda högra handen o.s.v. Hjärnan är alltså *modulär*.

Senare utvecklade man metoder för att undersöka aktiviteten i hjärnan. T.ex. *PET* (Positron Emission Tomography) låter forskaren se hur blodflödet och ämnesomsättningen i hjärnan varierar, och *MEG* (Magnetoencephalography) visar hjärnans elektriska aktivitet. När man undersökt hjärnan med dessa upptäckte man att det fanns en ännu mer specifik struktur inom de olika delarna av hjärnan. T.ex. i syncentret är cellerna organiserade så att de celler som aktiveras av en viss färg är nära varandra, och de celler som känner igen linjer i olika riktningar är nära varandra. Känseln har en liknande struktur där känseln på de olika delarna av kroppen finns organiserat efter kroppens form. Läpparna på ett ställe, kinderna i närheten, precis som halsen, men fötterna är längre borta. Motsvarande struktur går också att hitta för de motoriska delarna av hjärnan. Denna struktur där platsen i hjärnan motsvarar någon viss del eller enhet kallas för *tonotopisk karta*.

Inspirerade av de tonotopiska kartorna i hjärnan har konstgjorda tonotopiska kartor utvecklats. De är gjorda som neuralnät och räknar därför med vektorer.

Den mest kända variationen av sådana konstgjorda tonotopiska kartor är den självorganiserande kartan, oftast kallad *Self-Organizing Map* eller förkortat *SOM*. Den är utvecklad på Neuralnätsforskningscentret vid Tekniska Högskolan i Finland under början av 80-talet [Kohonen, 1982, Kohonen, 1990, Kohonen, 2001].

Varför kallas kartorna sjävorganiserande? De tonotopiska kartorna är ordnade på ett sätt där det *spatiella* är viktigt. Det som liknar varann är nära varann i rummet. Samma idé använder den självorganiserande kartan. Man ger data i vektorform och den självorganiserande kartan skapar en (vanligen) tvådimensionell karta av hur mycket de olika vektorerna liknar varandra. De som liknar varandra mycket är nära varandra på kartan, och de som inte liknar varandra särskilt mycket är längre från varandra på kartan.

SOM fungerar på följande sätt. Kartan består av ett nät noder, mer som ett fisknät, än som ett nätverk av datorer. Varje knut motsvarar en nod och varje nod sitter fastbunden med andra noder bredvid sig. De noder som är fastbundna nära kallas för nodens grannar.

Varje nod representeras av en *basvektor* som placeras *slumpmässigt* någonstans i samma rymd som inputdatan. Om inputdatan är tiodimensionell så placeras noderna i en tiodimensionell rymd. Man kan tänka sig att tråden i fiskenätet består av gummiband så att avståndet mellan noderna kan variera. Nu när alla noder i nätet är någonstans slumpmässigt i rymden är nätet ett enda trassel.

Sedan påbörjas inlärningsfasen. Man går igenom varje inputvektor en i gången, från den första till den sista. När man visar en vektor åt nätet, räknar man ut vilken nod i nätet som är närmast inputvektorn. Denna närmaste nod kallas för *vinnaren*. Om x är inputvektorn och m_c är vinnaren gäller:

$$\|x - m_c\| = \min()i\|x - m_i\| \tag{5}$$

Vinnaren flyttas lite närmare inputvektorn. Vinnarens grannar och grannarnas grannar, grannarnas grannars grannar o.s.v. till en viss gräns bildar ett grannskap. Alla i grannskapet flyttas närmare inputvektorn, övriga hålls på plats. Grannskapet N_c är stort i början och minskas mot slutet av inlärningen. Vid tidpunkten t flyttas noden m_i mot inputen x enligt följande:

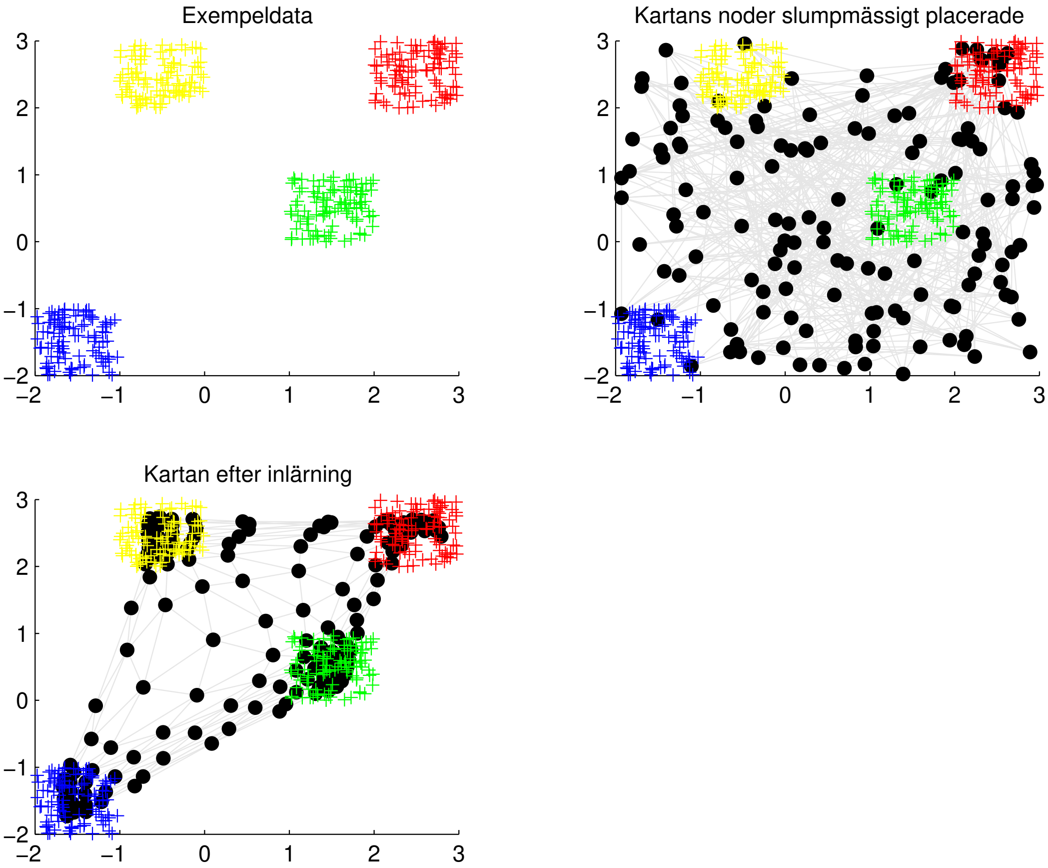
$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)(x(t) - m_i(t)) & \text{om i} \in N_c(t) \\ m_i(t) & \text{om i} \notin N_c(t) \end{cases} \tag{6}$$

Funktionen $\alpha(t)$ styr hur snabbt adaptionen sker $0 < \alpha(t) < 1$. I början är adaptionen snabb och avtar mot slutet av inlärningen. Samma gäller grannskapet, i början är det stort, mot slutet litet. Alternativt till det strikta grannskapet i ekvation 6 kan man använda en kontinuerlig funktion så att varje nod tillhör grannskapet till en del, de som är närmast tillhör mer, och flyttas därför mer, de noder som är längre bort tillhör grannskapet mindre och flyttas därför mindre.

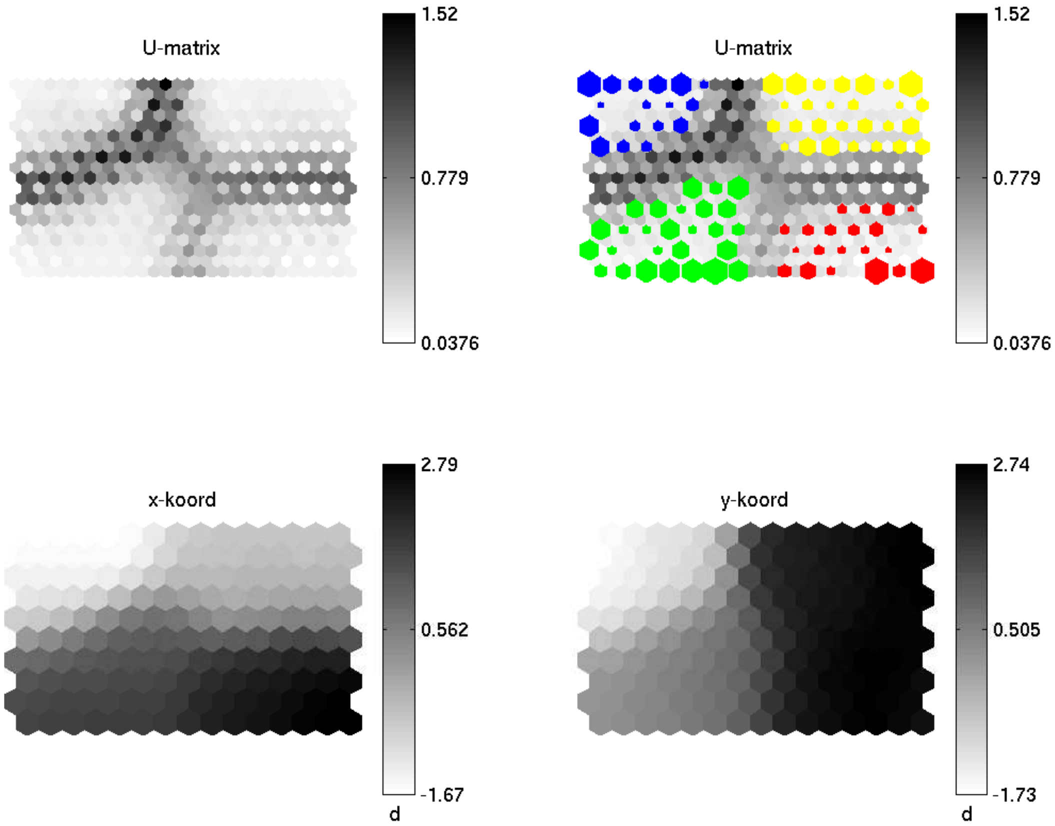
Man upprepar proceduren att söka vinnaren och adaptera enligt ekvation 6 för alla inputvektorer. Efter det upprepar man från början av inputdatan tills nätet hålls stabilt på plats i rymden. Det märkliga är att genom att följa denna procedur så kommer nätet att lägga sig vackert och organiserat i rymden efter var det finns input, trots att det startade som totalt trassel. För en mer visuell presentation se figur 8.

Det färdigt inlärda nätet kan användas som en tonotopisk karta. Man kan mata nya vektorer åt det och se vilken nod som vinner för den vektorn. Vektorer som liknar varann kommer att aktivera noder nära varann på kartan. Så kan kartan användas till både analys av en mängd vektorer och klassificering av nya vektorer. Se figur 9 för ett exempel på en sådan karta.

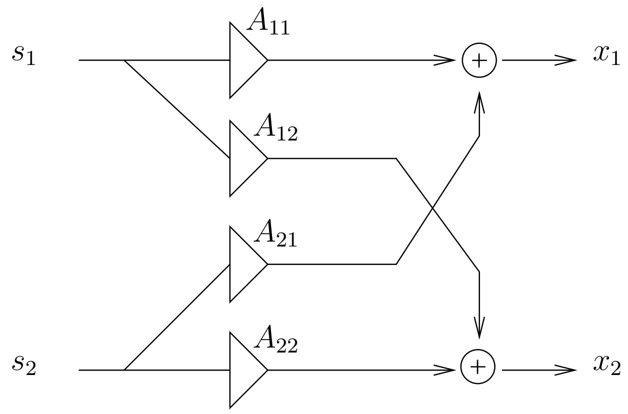
Kartan kan också användas till data-analys genom att man betraktar hurudana värden nodernas basvektorer m_i har. Då ser man hur värden för de olika dimensionerna i vektorerna varierar över kartan. På det viset kan man hitta korrelationer mellan dimensionerna, och andra statistiskt intressanta förhållanden, utan att enskilt behöva jämföra varje dimension en och en.



Figur 8: Uppe till vänster är de exempelvektorer vi använt som inputdata. Den är organiserad som fyra olika kluster, varje kluster utmärkt med var sin färg. Uppe till höger är kartan efter att dess noder placerats slumpmässigt i planet med inputdatan. Som man kan se är det ingen större ordning att finna. Nere ser vi kartan efter att den lärts med inputdatan. Kartan har organiserat sig så att dess flesta noder finns i, eller nära ett av klustren. I det tomma området i mitten finns färre noder, eftersom där inte finns datapunkter.



Figur 9: Kartan i figur 8 som den brukar visualiseras från “kartans yta”. Detta är det vanliga sättet, eftersom det kan tillämpas för data som är mångdimensionellt och inte kan visualiseras direkt. Alla bilder visar olika aspekter av samma karta, och det som länkar dem samman är positionen på kartan, samma position motsvarar samma nod på kartan, och färgen visar värdet på någon storhet i den noden, olika för de olika bilderna. Bilden uppe till vänster: Den s.k. U-matrisen som visar hur stort avståndet är mellan de olika noderna. De mörkare områdena motsvarar stort avstånd, och kan betraktas som bergskedjor på en vanlig geografisk karta. Bergen skiljer närliggande dalar från varann. Dessa dalar motsvarar data som är sinsemellan liknande och därför drar till sig många av kartans noder. Bilden uppe till höger är samma karta, men de olika datapunkternas plats har visualiserats med samma färger som i figur 8. Notera hur varje färg ligger i var sitt område, och att mörkare områden på U-matrisen ligger emellan dem. På detta sätt visar kartan inputdatans struktur: Det finns fyra skiljda grupper av datapunkter som inom gruppen liknar varann, men sinsemellan är olika. De två bilderna nere visar vad x- och y-koordinaternas värden är på de olika delarna av kartan. T.ex. På de röda datapunkternas område är både x- och y-koordinaten stor och därför är dessa figurer mörka där. På det gröna området är x-koordinaten ganska stor och y-koordinaten mitt emellan.



Figur 10: ICA-modellen för två källor: s_1 och s_2 som förstärks olika mycket av koefficienterna A_{ij} och adderas ihop för att utgöra de observerade signalerna x_1 och x_2. Med matriser kan det uttryckas:
$X = As = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$

1.2.4 ICA

Independent Component Analysis, eller ICA [Hyvärinen m.fl., 2001] är en metod att lösa Blind Source Separation-problem. Man brukar beskriva Blind Source Separation med det s.k. cocktail party-problemet: Tänk dig att du är på ett cocktail party och hör samtidigt flera personer tala på olika sidor av dig. Det ljud som kommer till dina öron är en blandning av flera olika samtal. Ändå klarar du av att separera signalerna tillräckligt mycket för att kunna lyssna på bara ett av samtalen åt gången. Människorna som talar motsvarar källorna i Blind Source Separation. Signalen som kommer till öronen är en summa av de olika källorna. Eftersom dina öron inte båda är helt på samma ställe blir blandningarna lite olika i olika öron, och därför är det möjligt att ana hur de olika signalerna blandas. ICA är ett sätt att hitta de ursprungliga källorna från de blandade signalerna. För att kunna se hur det kan gå till behöver vi uttrycka problemet matematiskt.

ICA-modellen Presentationen av ICA baserar sig på [Hyvärinen och Oja, 2000], och för fördjupad insikt kan den artikeln rekommenderas, som både utmärkt och lätt att förstå.

Matematiskt kan vi beskriva cocktail party problemet så här: X är en matris med de blandade signalerna, A är en matris av vikterna som avgör hur mycket varje källa hörs i de olika blandningarna, och slutligen s är en matris med de ursprungliga källorna. För dessa gäller sambandet:

$$X = As \tag{7}$$

D.v.s. X är en blandning av signalerna s betonade av viktmatrisen A. ICA räknar ut både A och s från en känd matris X.

Men ekvation 7 har ju oändligt många möjliga lösningar, eftersom det finns fler okända än ekvationer! Hur hittar vi de ursprungliga källsignalerna s och deras koefficienter A? Man kan inte lösa problemet utan att introducera ytterligare begränsningar. Men en hurudan begränsning ska man använda för att komma så nära de urpsrungliga källsignalerna s som möjligt? Eftersom vi tänker oss att signalerna skapas av olika källor är det förnuftigt att anta att signalerna s är *statistiskt oberoende*. Det betyder att om jag vet värdet för s_1 så ger det mig ingen information om värdet hos s_2. Om vi tänker oss in i cocktail partyt så betyder detta antagande att det människorna säger i en grupp som samtalar till höger om oss inte påverkar samtalet bakom oss på något sätt. Detta antagande är rätt rimligt. Det är bara om någon talar väldigt högljutt som den påverkar någon som deltar i ett annat samtal. Ofta betraktar vi processer där källorna är någon mekanisk process, och inte människor i vilket fall antagandet är ännu mer giltigt.

För att hitta de statistiskt oberoende signalerna måste vi också betrakta signalerna X probabilistiskt. Statistiskt oberoende kan uttryckas med hjälp av sannolikhetsfördelningar. Om m och n är statistiskt oberoende gäller:

$$p(m, n) = p(m)p(n) \tag{8}$$

Man kan alltså betrakta m och n var för sig, utan att behöva fundera på den andras värde.

ICA hittar oberoende signaler genom att estimera en matris W som projiserar X så att projektionerna WX är så oberoende varandra som möjligt. För den W som leder till största oberoende hos WX är $WX = s$ (och W är alltså pseudoinvers till A).

$$X = As \equiv WX = WAs = Is = s \tag{9}$$

Nu måste man hitta den W som projiserar datan så att man får maximalt oberoende signaler. Innan vi går in på hur det går till bör några egenskaper hos modellen göras klara:

1. De estimerade signalernas förtecken är godtyckliga
   Eftersom både A och s estimeras samtidigt kan man inte säga om båda koefficienterna i A ska vara positiva eller negativa i förhållande till värdena i s. Detta leder ofta till att man hittar de ursprungliga källorna s men att de har omvänt förtecken. Detta kan korrigeras med att ändra förtecken både i s och i A, ifall man på något sätt kan avgöra vad de korrekta förtecknen borde vara.

2. De estimerade signalernas styrka (energi) kan inte avgöras
   Vilken som helst faktor som multiplicerar A kan korrigeras med motsvarande inverterande faktor i s. Därför normaliseras de oberoende signalerna så att har en varians $var(s) = 1$. Förtecknens godtycklighet är egentligen ett specialfall av denna egenskap.

3. Vi kan inte hitta någon ordning för de oberoende signalerna
   Om vi ändrar ordning på de oberoende signalerna s vi estimerat med ICA, är dessa precis lika oberoende i alla fall. Eftersom vi inte vet signalernas relativa styrkor kan vi inte heller ordna signalerna efter dem.

Estimering av ICA. Vi vill estimera en matris W som ska maximera oberoendet mellan signalerna WX (eller minimera beroendet). Men hur gör vi det, att vara statistiskt oberoende är inte en funktion som man kan maximera, utan en egenskap som finns eller inte finns. Därför behöver vi en funktion vi kan optimera, och som ger oss ett bra mått för oberoende.

I statistiken finns ett resultat att summan av oberoende stokastiska variabler har en fördelning som är närmare en normalfördelning än vad de ursprungliga variablerna har (eng. Central Limit Theorem). Då kan vi som vill hitta oberoende stokastiska variabler söka sådana projektioner WX vars fördelningar är så långt från en normalfördelning som möjligt.

Men tänk om de ursprungliga s är normalfördelade! Det visar sig att man inte kan hitta oberoende komponenter för en normalfördelning, eftersom den är fullständigt symmetrisk. Det som ICA egentligen gör är att hitta en *rotationsmatris* A som roterar källsignalerna så att de blir datan X. Om distributionan är alldeles symmetrisk, som normalfördelningen, är alla rotationer lika oberoende. Vill vi hitta oberoende signaler måste de alltså vara av annan fördelning än normalfördelning. För normalfördelningar hittar vi bara en av de oändligt många möjliga rotationerna.

För andra fördelningar av s kan vi hitta de oberoende komponenterna. Men för optimeringen måste vi kunna avgöra hur nära en normalfördelning en variabels fördelning är. Därför behöver vi en funktion som estimerar det. Man kan använda olika funktioner för detta. En vanlig estimator är *kurtosen*.

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{10}$$

Kurtosen har problemet att den är väldigt känslig för enskilda avvikkande sampel. Det finns ett noggrannare mått av hur nära man är normalfördelningen: *Negentropi*. *Entropi* är ett mått ur informationsteorin som mäter hur mycket information en observation av en variabels värde ger. Ju svårare det är att förutspå en variabels värde, d.v.s. ju mer "slumpmässig" den är, desto högre är dess entropi. Normalfördelade variabler har den allra största entropin av alla fördelningar. Vi betecknar Entropin med $H(y)$. Negentropin är då:

$$negentropi(y) = H(y_{gauss}) - H(y) \tag{11}$$

Alltså skillnaden mellan variabelns entropi och entropin hos en normalfördelad variabel. Eftersom en normalfördelad variabel har den största entropin av alla fördelningar är negentropin alltid positiv.

Problemet med negentropin är att för att kunna räkna ut den från definitionen krävs att man estimerar sannolikhetsfördelningen för variablerna, och detta är mycket tidskrävande och svårt. Därför brukar man approximera negentropin från variablernas statistiker. Vanligen används en approximation av formen:

$$negentropi(y) \approx \sum_{i=1}^{p} k_i[E\{G_i(y)\} - E\{G_i(t)\}]^2 \tag{12}$$

Där G_i är icke-kvadratiska funktioner och t är en normalfördelad variabel med väntevärdet 0 och variansen 1. Variabeln y antas ha väntevärdet 0 och variansen 1. Om man väljer $G(y)$ väl får man estimat som är mycket mer robusta än kurtosen. Följande funktioner har visat sig mycket bra och användbara:

$$G_1(u) = \frac{1}{a_1} log(cosh\ a_1 u),\ G_2(u) = -exp(-u^2/2) \tag{13}$$

Beväpnad med dessa estimatorer löser FastICA-algoritmen de oberoende komponenterna ur X. Optimeringen i FastICA baserar sig på Newtons iterationsalgoritm. Algoritmen behandlas ytligt i [Hyvärinen och Oja, 2000] och djupgående i [Hyvärinen och Oja, 1997]. I detta arbete kommer vi uteslutande att estimera oberoende komponenter med FastICA-algoritmen som finns förverkligad för Matlab-programmeringsomgivningen [MathWorks, 2005].

Preprocessering. För att förenkla ICA-estimeringen kan man utföra vissa preprocesseringsoperationer. Dessa är inte nödvändiga, men kan ofta göra estimeringsalgoritmen enklare och snabbare.

En enkel operation som används är centrering av X. Man subtraherar väntevärdet från X så att matrisen får väntevärdet 0. Efter estimeringen av A kan man lägga till det ursprungliga väntevärdet till s och så få de resultat som motsvarar den ocentrerade X.

Den andra viktiga preprocesseringen som görs kallas på engelska *whitening* och innebär att komponenterna i X görs okorrelerade och att deras varians normaliseras till 1. För X gäller då:

$$E\{XX^T\} = I \tag{14}$$

Denna transform kan utföras t.ex. genom en dekomposition av kovariansmatrisens egenvärden. E är den ortogonala matrisen av kovariansmatrisens egenvektorer och D är en diagonalmatris med egenvärden.

$$E\{XX^T\} = EDE^T \tag{15}$$

Då kan man göra whitening-transformen genom en ny matris $D^{-1/2}$ där man upphöjt alla värden i diagonalmatrisen D med -1/2. Då blir den vita datan X_{vit}:

$$X_{vit} = ED^{-1/2}E^T x \tag{16}$$

Ofta är X en matris av mycket hög dimension, och då kan det vara fördelaktigt att sänka den dimensionen. Tekniken som kallas PCA (Principal Component Analysis) går ut på att man lämnar bort de rader i E och D som motsvarar de minsta egenvärdena, alltså de rader där värdena på diagonalen i D är minst. På detta sätt bevaras största delen av informationen i X intakt, men ICA-problemet blir enklare att lösa. PCA minskar dimensionen på ett sätt som bevarar de generella dragen i datan, men slumpmässigt brus och annat sådant faller bort. Om brus i datan är ett problem kan dimensionsminskning med PCA hjälpa. Ibland vill man inte ha så många signaler s som resultat, t.ex. då man strävar efter en överblick och visualisering av datan, och då innebär fler signaler mer att tolka, vilket inte alltid är vad man vill. En annan orsak att minska dimensionen kan vara att man vill ha resultat snabbt, och inte har tid att estimera så många komponenter. I fall som dessa är PCA en nyttig preprocessering.

1.3 Emergens

Ett begrepp som ger djupare insikt i ostyrd inlärning är *emergens*. Om man tillämpar t.ex. SOM på komplicerad data får man fram mycket intressanta, s.k. emergenta strukturer. Vad är då emergens?

Ett sätt att förstå vad emergens är, är att tänka på en myrstack. Myrorna är många och samarbetar på ett imponerande sätt. Därför börjar man lätt tänka att myrorna har en chef som bestämmer och koordinerar. Denna chef är säkert drottningen, tänker man. Men när man observerar myrorna kommer man fram till att de inte alls har någon chef. Alla myror följer vissa, relativt enkla, regler. När flera myror följer dessa regler resulterar det i att myrkolonin får ett avancerat och komplicerat beteende. Man kan säga att intelligensen hos myrkolonin är större än summan av de enskilda myrornas intelligens. En enskild myra klarar sig inte särskilt länge, men myrkolonin är kapabel till mycket avancerade saker, som att samarbeta med att föra mat till stacken när de hittar ett bra byte, t.ex. ett dött djur. Myrorna kan också begrava sina döda på ett ställe som är långt från stacken och samla sina sopor på en avstjälpningsplats, som är både långt från stacken och gravgården. (se [Johnson, 2001]).

Att enkla delar bildar ett system som är mer komplicerat än summan av delarnas komplexitet är en av de mest centrala delarna av det som kallas *emergens*.

1.3.1 Definition

Men det är inte bara det att enkla delar bildar något komplicerat, för om man plockar isär vad som helst som är komplicerat, t.ex. en bil får man till sist enkla delar. Något är emergent när det är *överraskande* att delarna fungerar ihop som de gör. David Chalmers [Chalmers, 2002], filosof och medvetenhetsforskare, säger att ett beteende på en högre nivå (t.ex. i en myrkoloni) är *svagt emergent* i förhållande till en lägre nivå (alltså myran) om den högre nivåns funktion är *oväntad*, med tanke på de regler som styr den lägre nivån.

I myrornas fall är det alltså så att beteendet som myrkolonin visar är *svagt emergent* eftersom det med tanke på de regler som styr en enskild myra, är oväntat att myrorna skulle fungera som de gör tillsammans i en koloni. Ett annat exempel på svagt emergenta system är t.ex. de självorganiserande kartorna (1.2.3). Reglerna för de enskilda noderna i nätet är välkända och relativt enkla, men det är överraskande att kartan i helhet organiserar sig och hittar sådana strukturer den gör, trots att beteendet kan härledas från reglerna som styr kartan.

Enligt Chalmers definition finns också *stark emergens*. Det är något där den högre nivåns funktion är *omöjlig att härleda* från den lägre nivåns regler, trots att det finns en uppenbar koppling mellan de båda nivåerna. Enligt Chalmers finns bara ett starkt emergent system: människans medvetande. Detta är ett mycket svårt filosofiskt territorium, men kan tjäna som jämförelse med den svaga emergensen. Människans medvetande är för tillfället ett stort mysterium som är svårt att förstå med dagens kunskap om hjärnans funktion. Ändå finns det en koppling mellan det fysiska som kan mätas från hjärnan och det en människa upplever att händer i hennes medvetande. De är alltså sammankopplade, men hur är ett stort mysterium. En tydlig släktskap med den svaga emergensen kan observeras och därför kallas båda fenomenen emergens. I detta arbete hänvisar ordet emergens härefter alltid till svag emergens.

1.3.2 Samband med inlärning och språk

Inlärningen på en självorganiserande karta resulterar i en emergent struktur. Med andra ord har kartan en helhetsstruktur som är mycket mer intressant än man skulle kunna tro när man hör hur kartan fungerar. De enkla principerna leder alltså till överraskande komplexitet. Det intressanta med det här fenomenet är att i naturen finns det en hel del på liknande sätt komplicerade system man gärna skulle kunna härma och analysera, men inte vet hur.

Inte minst på språkinlärningens område finns det gott om exempel av väldigt komplicerade fenomen. Hur lär man sig överhuvudtaget? T.ex. råder inget större tvivel om att det i språket finns kategorier för ord. De kanske inte är exakt som den generativa grammatiken säger, men det betyder inte att de inte skulle finnas alls. Men varifrån kommer de då? Honkela m.fl. tog de 150 vanligaste orden i bröderna Grimms sagor och räknade ut en vektor som beskrev hurudana kontexter de olika orden förekom i. Vilka ord som kom före och efter ordet själv, och hur många gånger. Det visade sig att när man matade dessa vektorer åt en självorganiserande karta så lär den sig en struktur där bl.a. verb, substantiv, adjektiv och prepositioner har var sin del på kartan [Honkela m.fl., 1995]. Det betyder dels att dessa kategorier utmärker sig så mycket i språket att de kan observeras i en sådan analys. Men det kan också betyda att dessa klasser är emergenta resultat av människans inlärning av språk. Kartans "erfarenhet" av kontexter leder till att klasserna framträder. Det betyder att det är möjligt att lära sig kategorier genom att enbart iaktta språket. Om den relativt enkla självorganiserande kartan hittar dessa kategorier, varför skulle då inte människans inlärningsmekanismer kunna göra detsamma?

Därför kan man misstänka att många av språkets fenomen på samma sätt kunde vara emergenta i förhållande till människans inlärningsmekanismer. Som vi insåg påverkas språkets utveckling av inlärningen, och begränsas av hurudana strukturer människans sinne föredrar och har lätt att lära sig. Då blir emergensen ett fenomen som vi vill inkludera i våra system för att behandla naturligt språk, eftersom emergensen skulle göra det möjligt att närma sig samma komplexitet som man kan observera i språket. Den kan vara ett verktyg för att bearbeta det paradoxala i att språket å ena sidan är enkelt och regelbundet, å andra sidan oregelbundet, flertydigt och oerhört komplicerat.

Om man tar den självorganiserande kartan som ett exempel av en emergent representation, d.v.s. en representation av data vars förhållande till datan är emergent (se [Ultsch, 1994] för det perspektivet på SOM). Intuitivt borde en sådan representationer vara väldigt nyttiga för många språkprocesseringsuppgifter, eftersom de innehåller struktur som hittats i riktiga texter och som dessutom motsvarar intuitiva uppfattningar om vad som borde finnas i språket (t.ex. ordklasser). Det är lätt att tänka att dessa representationer t.o.m. kan vara något som liknar dem som människans hjärna använder i sin språkprocessering. Men hur ska man använda sådana representationer i ett större system? Det är inte alltid så lätt att säga exakt vilken viktig storhet som t.ex. SOMs basvektorer motsvarar. Det känns som om dessa representationers potential inte utnyttjas till fullo så länge de bara används för att visualisera något för mänskliga observatörer. De borde gå att tillämpa på språkprocessering. Men hur det ska gå till är inte alltid så självklart. Emergens handlade ju om att något var överraskande. Överraskande är inte en egenskap som man lätt kan bygga komplicerade modeller av. Det är lättare att hantera saker som "i medeltal", "optimal" och "kategori" m.m. För att kunna nyttja emergensen måste det finnas rum för den i grundteorierna.

Innan vi kan ge oss på problemen med de emergenta representationerna behöver vi därför någon slags grundteori för språk, istället för den generativa grammatiken, och en annan teori för intelligens, istället för den symboliska paradigmen som artificiell intelligens utgår ifrån. Vi kommer att söka sådana i följande kapitel.

1.4 Kognitiv vetenskap - Idéer om språk och intelligens

Den kognitiva vetenskapen forskar i sinnet och intelligensen som ett empiriskt fenomen. Det betyder att man undersöker hur människan på riktigt löser problem, använder språk, upplever saker o.s.v. Man strävar alltså inte efter ideal t.ex. hur problem borde lösas, hur språk borde användas etc., utan studerar hur människan gör i praktiken.

Forskningen är mycket tvärvetenskaplig, innehållande bl.a. språkvetenskap, psykologi, neurologi, artificiell intelligens. Genom den kognitiva vetenskapen har det uppkommit en hel del värdefull forskning om språkinlärning och språkkunskap. Vi ska hastigt svepa igenom några viktiga resultat som påverkar vår syn på språk och intelligent resonerande. Genomgången är inte på något sätt komplett, men ger en inblick i intressanta frågeställningar som är relevanta också för hur man konstruerar datamodeller av språket. Kapitlet baserar sig på [Lakoff och Johnson, 1999] och [Gärdenfors, 2000].

1.4.1 Hur beskriver man betydelse?

Hur ska man beskriva vad ett ord betyder? Vi kan ta ett exempel för att göra det hela mer konkret: Ordet 'diskborste' vad betyder det egentligen? Man kan säkert titta i ett lexikon och hitta någon definition i stil med: "En borste som används vid diskning av servis". Vad betyder 'borste' då i såfall? "Sak som används för att ta bort orenheter, m.h.a. grövre strån, organiserade nära varandra"? Men vad betyder då 'orenheter' eller 'strån'?

Den traditionella lösningen, enligt referensteorierna som nämndes i kapitel 1.1.3, är att säga att varje menings betydelse kan beskrivas med påståenden som är sanna i världen. T.ex. "Bilen är blå" kan översättas till logiska formler som säger att påståendet "Bilen B är blå" är sant, där B är en viss bil som finns i världen och blå är en färg som bilar i världen kan ha, och bilen B specifikt har.

Men betyder 'blå' något i sig självt? Vad innebär det att en bil är blå, rent fysiskt? Vi vet att blå är en färg som bilen har. Men vad är färg? Vi vet att synligt ljus är elektromagnetisk strålning som har olika våglängder. I våra ögon finns tappar och stavar, nervceller som reagerar på ljusstrålningen. Det finns tre sorters tappar som reagerar på olika våglängder motsvarande färgerna rött, blått och grönt. Ljuset innehåller en blandning av dessa våglängder, och denna blandning ger oss den slutgiltiga färgen. Bilens blåa färg kommer från att dess yta har vissa egenskaper som reflekterar blå färg. Men om man lyser på den "blå bilen" med enbart grönt ljus kommer den inte längre att se blå ut. Är det då verkligen så att bilen är blå? I [Hardin, 1988] (s. 2) finns en lista av 15 olika fysiska källor för en ljusstrålning, allt från vibrationer till förändringar i molekylers elektronkonfigurationer. Två processer som strålar på samma frekvens behöver därför inte fysiskt likna varann. Man kan m.a.o. säga att bilen har en sådan reflektionsegenskap att den kan se blå ut i våra ögon, vid rätt ljusförhållanden.

Nyckelorden i föregående mening är "i våra ögon". Begreppet blå är beroende av hur våra ögon är konstruerade. Skulle vi inte ha tappar i ögonen för denna specifika frekvens som motsvarar blå skulle inte begreppet blå kunna finnas. Dessutom kan man inte påstå att det verkligen finns något i den fysiska världen som skulle vara blått i sig själv, d.v.s. om ingen människa skulle kalla det blått. Ordet 'blå' får alltså sin betydelse ur både ett fenomen i den yttre världen, ljusstrålningen, och ur människans sätt att uppfatta ljusstrålning. (För en djupgående beskrivning av färgsyn, se [Hardin, 1988].)

Detta betyder att ordet "blå" inte är en egenskap som bilen B besitter. Färgen blå finns inte i den yttre världen, där finns bara reflexioner, strålning och en mängd andra fenomen.

Färgen blå finns i människans uppfattning av världen. Betydelsen av ett ord är alltså beroende av människans kropp och sinnen. Detta kallar George Lakoff och Mark Johnson för att sinnet är förkroppsligat (eng. embodied mind) [Lakoff och Johnson, 1999].

Detta med färg är inte alls det enda exemplet på begrepp som är beroende av kroppen. T.ex. Beskrivs positioner i en rymd på samma sätt i språk över hela jorden. Dessa betydelser tycks uppkomma ur människans motoriska och visuella förmåga. Som tankeövning kan man fundera på vad ordet 'bil' betyder.

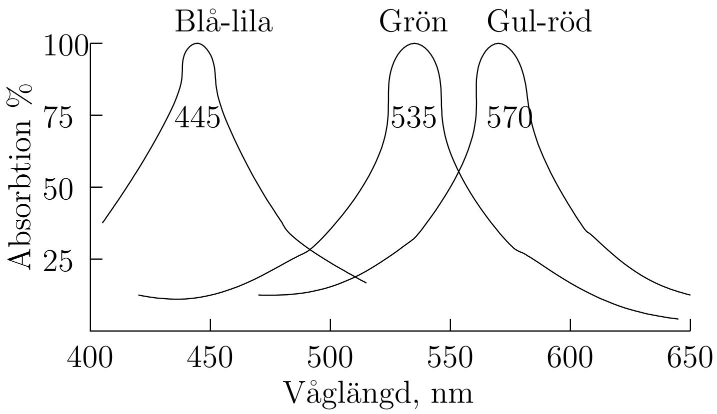
Peter Gärdenfors summerar den kognitiva synen med orden "betydelserna finns i huvudet" (från [Gärdenfors, 2000] s.160). Det betyder att symbolerna i språket, orden och konstruktionerna, hänvisar till någon kognitiv struktur i talarens huvud baserad på upplevelser av den saken, inte till sakerna själva i den yttre världen. Gärdenfors säger ytterligare "betydelsen kommer före sanningen" (från [Gärdenfors, 2000] s.160). Det betyder att en mening har en betydelse oberoende om meningen, som påstående, är sant. Meningen "Endast tomten är vaken" betyder något oberoende om påståendet är sant eller inte.

Ordens betydelse är alltså inte en hänvisning till något i den yttre verkligheten, utan en hänvisning till en inre upplevelse av detta något. Det betyder att alla människor har betydelsen i sig. Den kan antingen vara medfödd, men mer sannolikt är den inlärd. Som vi nämnde i kapitel 1.1.4, är språkförmåga något en kultur och ett samhälle äger kollektivt. Detta gäller särskilt betydelsen. Alla har (lite) olika upplevelser, men ordens betydelser är tillräckligt lika för att man ska förstå varann. Då måste samma ord kopplas ihop med samma betydelse för olika individer. Men hur går det till?

1.4.2 Symbolbindningsproblemet

Trots att vi precis förklarat att ett ords betydelse inte går att uttrycka som förhållandet mellan symboler och världen, är ordet själv till sin natur symboliskt. Det betyder att trots orden 'heta' och 'äta' liknar varandra i uttal, har deras betydelser inget med varandra att göra. Det finns alltså inte något systematiskt förhållande mellan hur ord låter och vad de betyder. Ordet är en symbol för sin betydelse. Ordet 'katt' syftar alltså på de upplevelser av riktiga katter man kan se och tänka sig.

Men hur lär man sig att när någon säger /katt/ betyder katt och inte, t.ex. diskborste? Ljudet i ordet måste på något sätt 'bindas' till en betydelse, men hur det går till är inte något enkelt problem. Därför har det fått ett eget namn: s.k. symbolbindningsproblemet (eng. symbol grounding problem).

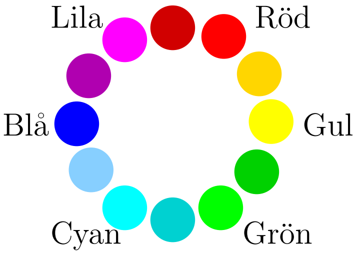


Figur 11: I ögat finns nervceller som är känsliga för olika våglängders ljus. Här är de receptiva fälten för dem. Ju högre värde grafen har för en specifik våglängd, desto mer reagerar cellerna på på den våglängden. Vi kan också notera att precis som vi vet att orange är färgen mellan gult och rött, så vet vi också att lila är mellan blått och rött. Men rent fysiskt är det så att våglängden för orange är mellan våglängden för rött och gult. Rött och lila däremot är i var sin ända av spektrumet, och ligger inte bredvid varandra. Bilden från [Buss, 1973].

Den enkla förklaringen är att man ser en katt och hör någon säga 'katt'. Jo, så är det säkert, men inte är det ändå riktigt så enkelt. Tänk dig att du ser en katt ligga på en matta och hör någon säga "katten ligger på mattan". Hur vet du då att det som den andra personen sade inte betyder "mattan är under katten"? Hur vet du att ordet 'katt' är djurets namn? Man kunde lika gärna säga 'djur', 'lurvig', 'söt' eller 'vit' när man ser katten. Dessutom finns det meningar som för en vilse. Tänk t.ex. att du ser en katt, men mamma säger "dags att lägga sig". Hur kan du veta att det inte var katten hon talade om [Landau och Gleitman, 1985]?

Svaret på dessa frågor är att det måste finnas något slags mekanism som är lika för alla människor som avgör vilka 'saker' är sannolika att bli begrepp som har medföljande ord. Tänker man sig att orden är symboler för något i världen, och inte namn för upplevelser via sinnena, blir det väldigt svårt att förstå hur orden kan hitta rätt bland de ofantligt många möjliga symbolerna. Om man istället tänker att människans kroppsliga världsuppfattning påverkar vilka betydelser som prioriteras i det hon ser, hör, smakar, luktar och känner blir det lättare att förstå. Människor är tillräckligt lika för att tänka tillräckligt lika om vad som är viktigt. Väldigt sällan säger man ju "mattan är under katten" istället för "katten är på mattan". Det första säger man bara ifall man verkligen talar om mattan, men det är en inbyggd mänsklig egenskap att vanligen anse katten viktigare.

Ett faktum som stöder att denna syn är riktig, är att den språkliga utvecklingen hos ett barn går hand i hand med den motoriska utvecklingen. Den enkla bortförklaringen att artikulationen av ord kräver ett visst motoriskt utvecklingsskede räcker inte till, eftersom många barn lär sig säga något ord innan de börjar jollra. Deras språkliga utveckling framskrider i övrigt alldeles normalt. De kan inte kombinera ord som 'daddy' och "bye - bye" till en mening, trots att de jollrar i långa 'meningar', med intonation och allt. Det betyder att deras motorik är tillräckligt utvecklad för att kunna artikulera ord, men det verkar som om deras kognitiva mekanismer inte är det [Lennberg, 1967].



Figur 12: Färgerna som man uppfattar dem. De färger som upplevs liknande är bredvid varandra. Så också lila och röd, som vi vet från figur 11 har de inte närliggande frekvenser. Ändå uppfattas de som besläktade färger. Denna kategorisering uppkommer alltså som en kombination av ljusets fysiska egenskaper och människans sinnen och hjärna.

1.4.3 Begrepp

Vi har alltså kommit till att orden är symboler som pekar på ett begrepp som är definierat på något sätt beroende av kroppen och sinnena. Men kan vi veta något mer om dessa begrepp? Hurudan struktur har de, är de lika för alla människor o.s.v.?

Peter Gärdenfors föreslår att begreppen kan uttryckas med hjälp av en rymd, en konceptuell rymd (eng. conceptual space). Den består av en topografi, d.v.s. en form och dimensioner som motsvarar olika parametrar som våra sinnen tar in [Gärdenfors, 2000].

Varför just en rymd? Varför inte symboler, vad är det för skillnad?

Den värld vi ser är inte diskret, så att det vi ser antingen skulle vara sant eller falskt. Det mesta har olika grader. T.ex. är människors längder inte lång eller kort, utan allt från en halv till två och en halv meter på en kontinuerlig skala. Samma gäller t.ex. mörk och ljus, kall och varm o.s.v. Då är det mer naturligt, eller rentav nödvändigt, att uttrycka saker med ett kontinuerligt konceptuellt system, som områden i en rymd, än som enbart symboler.

För att göra det hela mindre abstrakt kan man ta färgrymden som exempel. På basen av hur människor subjektivt uppfattar färger kan man organisera färgerna så att de som uppfattas som liknande är nära varann, och de som är mycket olika är långt från varann. Ett enkelt sätt att organisera de olika färgerna är då att lägga dem på en cirkelbåge (figur 12). Färger som ligger bredvid varandra är de som upplevs “nära varann”. Intressant nog upplevs rött och lila vara närliggande färger. Rent fysiskt motsvarar ju färger våglängder hos ljusstrålningen. Lila har den kortaste våglängden och röd den längsta (se figur 11. I den fysiska världen är dessa färger alltså inte närliggande, men nog i människans uppfattning av färger. Begreppen och fysiken är alltså inte i direkt förhållande med varann, utan människans sinnesegenskaper styr begreppsbildningen.

I termer av konceptuella rymder säger man att färgrymden har en topologi som är en cirkel och inte en rak linje.

1.4.4 Metaforer

Ett av de stora resultaten som den kognitiva vetenskapen kommit med, är insikten om hur metaforiskt språket är. En metafor innebär att man använder strukturer, eller egenskaper, från ett område för att beskriva något fenomen inom ett annat område. T.ex. använder man rymdbegrepp för att beskriva tid. “Det har jag lämnat bakom mig”, “Jag har ännu slutarbetet framför mig”. Framtiden beskrivs som kroppsligt framför en och den gångna tiden som kroppsligt bakom en [Lakoff och Johnson, 1980].

Tidigare uppfattade man metaforerna som något exotiskt, enbart använt i avvikande språk som poesi o.d. Senare har man upptäckt att metaforer förekommer i stor utsträckning i allt språk, vardagligt som poetiskt. Man måste anstränga sig för att inte använda metaforer, och mycket som vi säger varje dag går inte att uttrycka utan metaforer. Ex:

"Hur går det?"
"Tack, bra, det känns uppåt"

Den första metaforen är att jämföra sinnesstämning med att gå i frågan. Den andra är att uttrycka trevliga känslor som 'uppåt', ett ord för riktning i rymden.

Metaforerna antas fungera så att man tillämpar samma resonemang som man har för ett visst domän på ett annat domän. T.ex. "Han har nått toppen av sin karriär". Då använder man metaforen att karriären är en vandring, och att bättre är högre. Man beskriver karriären som något man går igenom fysiskt och när karriären går bättre beskrivs det som att det går uppåt på vandingen. Man vet från fysiska vandringar att inga kullar går uppåt för evigt, och när man nått toppen så börjar det gå nedåt. Denna kunskap om den fysiska verkligheten tillämpar man sedan på den abstrakta karriären.

Väldigt mycket abstrakta begrepp är metaforiska. Så många att man är frestad att påstå att alla abstrakta begrepp är metaforiska. Den teoretiska tanken bakom ett sådant påstående är att sinnet måste använda den kunskap som finns att få. Det betyder att de praktiska erfarenheter man har av konkreta, fysiska saker kroppen och sina sinnenena är den enda kunskapen som kan användas. Då är alla begrepp grundade i den kunskapen, och mekanismen för utvidgandet av tankar till det abstrakta går via metaforerna [Feldman och Narayanan, 2004]. Det betyder att tankarna i själva verket inte går från det generella till det specifika, utan tvärtom, från det specifika till det generella.

Praktiska tillämpningar av dessa idéer är bland annat ett experiment där David Bailey konstruerade ett program som skulle lära sig olika verb där en hand gör något. Dessa verb var t.ex. 'grab', 'grasp', 'slide', 'pinch' och 'lift'. Problemet är att för varje sådant verb finns inte en exakt motsvarighet i olika språk, utan handlingarna delas in i olika ord, på annorlunda sätt, i språken. T.ex. på spanska finns två ord som motsvarara svenskans 'trycka' (eller engelskans 'push'), 'pulsar' betyder att trycka på en knapp, och 'presionar' täcker de övriga betydelserna.

Baileys modell representerar de olika handlingar med olika parametrar som kontrollerar en hand, hur hastigt handen ska röra sig, med hurudan kraft o.s.v. Parametriseringen täcker inte exakta muskelsammandragningar, men är tillräckligt noggrann för att en konsgjord robothand kan kontrolleras med hjälp av den. Bailey tränade programmet med 15 engelska verb med 18 olika betydelser med neuralt trovärdiga metoder för inlärning. Sedan lät han systemet klassificera 37 förut osedda handlingar enligt vad det lärt sig. Systemet klassificerade helt rätt för 80% av fallen och alla felen det gjorde var för begrepp som gick på varann som 'move' istället för 'push' [Bailey, 1997].

Om abstrakta tankar fungerar genom att metaforiskt simulera samma rörelser för något annat domän borde det gå att konstruera en modell också av abstrakta idéer. Srinivas Narayanan utvecklade en sådan modell för att förstå texter om internationell ekonomi, via metaforiska associationer till fysiska rörelser, och experimentet bekräftar att en sådan utvidgning är möjlig, både att lära sig och för att förstå abstrakta texter ([Narayanan, 1997, Narayanan, 1999]).

1.4.5 Ett förslag för hur sinnet kunde vara uppbyggt

Men vad betyder detta för hur man gör system som behandlar naturligt språk? Man kan se att det leder till en ny sorts arkitektur som inte är vare sig som i artificiell intelligens eller neuralnät. I symbolisk AI måste allt uttryckas explicit. Någon måste formulera en regel för allting. Neuralnäten i sin tur lär sig nog från data men det går långsamt och kunskapen som lagras i nätet är svår att uttrycka explicit. Peter Gärdenfors [Gärdenfors, 2000] föreslår att man skulle indela kunskapen i tre skikt:

- Den subkonceptuella nivån
- Den konceptuella nivån
- Den symboliska nivån

Den subkonceptuella nivån är människor inte är direkt medvetna om, men den är aktiv hela tiden och tolkar omgivningen så att världen blir förståelig. Om vi tar språket som exempel så vet vi att när människor talar med varann överförs språket som ljud i luften mellan människorna. Ljudet består av vibrationer av olika frekvens och amplitud som lyssnarens öron tar emot. Den subkonceptuella nivån förändrar i detta fall vibrationerna till ord och meningar som lyssnaren uppfattar. Orden och meningarna tillhör den konceptuella nivån, de är *begrepp* som relaterar till människans erfarenheter och upplevelser.

Den symboliska nivån motsvarar den traditionella AI:n och Chomskys språkteorier, allt består av regler.

Den subkonceptuella nivån motsvarar neuralnäten. Man har en massa signaler (t.ex. ljuspunkter på näthinnan) och man ska lära sig något användbart från dessa.

Enligt Gärdenfors finns mellan den symboliska- och den subkonceptuella nivån en konceptuell nivå som kan beskrivas som en lågdimensionell rymd. Vi nämnde tidigare Baileys och Narayanans modeller där t.ex. handrörelser beskrevs med några parametrar. Den konceptuella nivån motsvarar denna parametrisering. Symboliska system använder sig inte av rymder och därför är det svårt att uttrycka avstånd och likhet. Vi kan göra observationen att avstånd och likhet uppenbart finns i vår begreppsvärld, t.ex. liknar en häst mer en åsna, än en sköldpadda. Sådana förhållanden är lätta att uttyrycka i en rymd, men mycket svåra att uttrycka symboliskt.

Neuralnät, å andra sidan, använder sig av vektorrymder där likheter och kontinuitet kan uttryckas enkelt. Men vektorrymderna i neuralnäten är väldigt högdimensionella och dimensionernas betydelse är inte lätta att tolka. Detta tyder på att dessa väldigt detaljerade komponenter inte har någon direkt motsvarighet i människans konceptuella system. Vi saknar vardagliga begrepp för att hantera högdimensionella vektorrymder.

Vi kan konkludera att Gärdenfors' modell verkar tilltalande framom enbart symboliska eller enbart neurala system. Den kan användas som en slags arbetshypotes som hjälper oss förstå och dela in språkhanteringens problematiker i mer hanterbara delproblem.

1.4.6 Kognitiv vetenskap och statistisk språkhantering

Om orden får sin betydelse ur kroppsliga erfarenheter kan man inte vänta sig att man skulle kunna konstruera system som kan behandla språk (ens) nästan lika bra som en människa, utan att systemet har kroppsliga erfarenheter. Däremot kunde det vara möjligt att på något sätt representera dessa upplevelser i någon standardform, så att man kunde överföra språkkapabilitet från en maskin till en annan, utan att de alla skulle behöva en egen kroppslig upplevelse. Då kunde man tillföra ett metaforiskt maskineri och få en maskin som skulle verka mycket intelligent, eftersom den 'förstår' vad orden betyder. Men t.o.m. en sådan maskin måste även veta hur olika konstruktioner förhåller sig till betydelsen. Förutom kunskap i ordens betydelser behöver man veta vad som händer med betydelsen när orden kombineras, och hur orden *får* kombineras enligt språkets regler. Dessa strukturella problem kunde statistisk analys kanske ge insikt i.

Den kognitiva vetenskapens resultat leder alltså till att man inte kan lära sig språk *enbart* statistiskt. Det går inte att bara ta en väldigt lång text och mata den åt ett rätt sorts dataprogram som sedan lär sig använda språk. Det kanske känns som en tråkig tanke man måste godkänna, men det är en ännu tråkigare tanke att hela sitt liv försöka göra något som inte är möjligt, och dessutom misslyckas.

Då blir den statistiska språkhanteringens lott att bättre förstå hur språket är uppbyggt. Mycket i språket är också statistiskt i sin natur, och det är möjligt att hitta mycket struktur bara ur text, som vi kommer att se i kapitel 4. Problemet är att hitta bra modeller för denna struktur. Modellen borde kunna göra mer än att bara visa strukturen på ett sätt som en människa kan förstå den, utan sparar den på ett sätt som är möjligt att använda maskinellt, t.ex. för att generera grammatikaliskt rätt språk.

1.5 Learning to Translate forskningsområdet

Det är stor skillnad mellan *data* och *information*. Detta har blivit allt tydligare p.g.a. informationsteknologins tillväxt, d.v.s tillämpningen av datamaskiner. Data är vad som helst som uppmätts eller sparats, t.ex. temperaturen i Helsingfors 1970-2000, eller alla avsnitt av Dallas sparade på videokasetter. Den viktiga skillnaden mellan data och information är att information är färdigt processerad för att säga någonting åt sin iakttagare.

Exempel på information som utvunnits ur temperaturdatan, kunde vara en analys om hur medeltemperaturen i Helsingfors utvecklats de senaste 30 åren, eller en analys av vilka avsnitt av Dallas man nödvändigtvis måste ha sett för att alls hänga med i intrigen.

Men, t.ex. Dallas är ju redan färdig information, för det är meningen att konsumeras som det är, kan någon påpeka. Visst, det är helt sant, och tar man den synvinkeln är motsvarande data den magnetiska signal på videobandet som sparar bild- och ljudinformationen.

Med andra ord kan sägas att vad som är data och vad som är information beror på hur man betraktar något.

Ett annat exempel är om man går till ett medicinskt bibliotek och sätter sig och läser:

“Av NASH-patienterna är 40-100% obesa, 20-75% har typ II diabetes, hyperglykemi eller glukosintolerans och 20-81% hyperlipidemi.”

Om man inte har medicinsk utbildning är denna text sannolikt rena rama grekiskan (eller kanske närmare bestämt latin) och man förstår inte särskilt mycket. Man kunde säga att det är mer data än information för läsaren. Om man däremot läste texten:

“NASH patienter, är personer med en sjukdom som leder till fettavlagring i levern, trots att deras alkoholkonsumption inte orsakar det. Av dem är 40-100% överviktiga. 20-75% lider av sockersjuka som bryter ut vid vuxen ålder, för högt blodsocker eller glukosintolerans. Glukosintolerans innebär att halten glukos, alltså druvsocker, i blodet avtar onormalt långsamt. Av NASH-patienterna har 20-81% hyperlipidemi, alltså rikligt med fett i blodet.”

Sannolikt var den senare texten lättare att förstå och ta del av. Båda texterna är på samma språk, men olika fackspråk. *Learning to Translate* forskningsområdet studerar dessa olika fackspråk och har som målsättning att automatiskt kunna översätta från det ena fackspråket till det andra. Innan man kommer så långt behöver man studera på vilka sätt dessa olika fackspråk skiljer sig från varandra. [Honkela och Kohonen, 2005]

1.5.1 Metodologi

Största delen av maskinell språkforskning har gjorts med symboliska metoder, t.ex. Chomskys teorier, som grund. Learning to Translate tar en mer kognitiv och konnektionistisk syn på det hela, genom att inte basera språkforskningen främst på symboler utan på människans upplevelser, erfarenheter och kognition. Det här innebär att människor inte förstår varandra enbart därför att de talar samma språk, utan främst för att deras gemensamma språk beskriver liknande upplevelser. Ordet "kasta" betyder inte exakt samma sak för olika människor i olika sammanhang, men eftersom de vet både hur det ser ut och hur det känns att kasta något kan de förstå varann. Därför är variationer mellan dialekter, fackspråk, kulturer och individer inte bara ett problem som man kan bli av med genom att välja "korrekt" språk för sin analys, utan en grundläggande egenskap hos språket. Språket förändras alltefter att människornas upplevelser förändras, och det är inte frågan om ett "förfall" av språket, utan en mekanism i språket som gör att det fungerar så bra som det gör som kommunikationsmedel. Kunskapen vi har om vår omvärld formar vårt språk. Språket är inte bara struktur och grammatik, utan ett systematiskt sätt vi beskriver våra upplevelser.

Den specifika kontributionen forskningsområdet vill ge vetenskapen är nyttjandet av ostyrd inlärning och statistiska metoder för dessa forskningsproblem. Centrala begrepp är SOM, ICA, Fuzzy Set Theory, Rekurrenta neuralnät och dynamisk systemteori.

1.5.2 Centrala delproblem

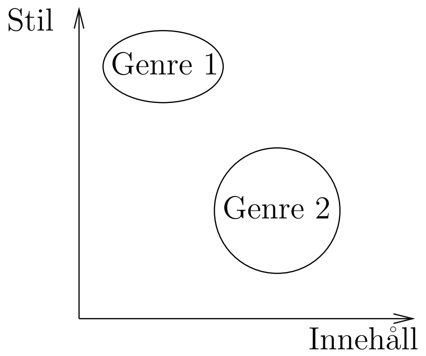
Learning to Translate är ett komplicerat forskningsområde. Därför kan det vara fördelaktigt att dela in det i mindre delar. Dessa alla är viktiga för att kunna lösa problemet av översättning mellan fackspråk.

- Stilanalys. Hur kan man detektera vilket fackspråk en viss text tillhör? Hur kan man klassificera texter enligt stil? Stilanalysen har ofta blivit lite forskad i jämfört med informationssökningen.
- Grammatikinlärning. För att processera och producera språk behövs information om språkets struktur. Alternativa modeller till den generativa grammatiken behöver utforskas.
- Statistisk översättning. Man har länge försökt göra program som automatiskt översätter mellan olika språk. Dessa metoder behöver anpassas för denna specifika översättningsuppgift. Som del av denna komponent finns centralt genereringen av språk.
- Emergenta representationer. Det finns mycket struktur i språket som är svår att uttrycka med symboliska modeller, men som kan uttryckas med kontinuerliga modeller. Sådana modeller borde därför utvecklas också för behandling av naturligt språk.

För ett så vitt forskningsområde finns givetvis hur mycket som helst relaterad forskning. Vi har koncentrerat oss på några huvudriktiningar av Learning to Translate-temat. Dessa är främst stilanalysen, något som ofta blivit i skymundan för analysen av innehåll i texter. Det vi kallar stilanalys kallas ofta i litteraturen för textkategorisering (eng. text categorization) [Sebastiani, 2002]. Specifikt gör vi textkategorisering på basen av stilegenskaper.

En annan huvudriktning är språkgenereringen. Automatisk översättning innehåller som nödvändig komponent generering av språk. Det känns mindre ansträngande för en människa att säga något hon vill kommunicera, än att översätta något en annan säger. Därför kan man spekulera om språkgenerering är mer än bara en del i ett översättningssystem. Vi behandlar generering, men inte översättning, eftersom det behandlats på andra ställen (se t.ex. [Hutchins, 1986]).

Ett tredje tema vi tar upp är emergenta strukturer. Ostyrd inlärning leder ofta till emergenta strukturer. Hurudan struktur får man när man tillämpar dessa metoder på naturligt språk? Hur kunde man eventuellt tillämpa det man hittar på generering?



Figur 13: Vår stildefinition: Stil och innehåll är ortogonala. Genre är kluster i detta plan.

2 Stilanalys

Vad är stilanalys? Vad är stil? Inom den statistiska stilanalysen är terminologin minst sagt en enda röra. Därför definierar vi en egen terminologi som åtminstone är entydig och har likheter med många andras definitioner.

Vi vet att en text har ett innehåll, det är vad texten handlar om. Samtidigt vet vi också att det finns andra aspekter av en text. Två vetenskapliga artiklar inom datateknik liknar varandra, trots att de kanske inte har samma innehåll. Vi kallar det som får dessa artiklar att likna varandra, som inte har med innehållet att göra, för *stil*. Vi tänker oss, att texten består av flera oberoende komponenter (tänk som i ICA 1.2.4). Då kallar vi två av dem för *innehåll* och *stil*. En *genre* är något som t.ex. science-fiction, dagstidningstext, vetenskaplig artikel o.s.v. Genren är en gruppering där de texter som tillhör genren liknar varandra till både innehåll och stil.

Men för att man ska kunna göra statistisk analys av texter så måste man konvertera dem till något slags nummerrepresentation. Vi ska ta en titt på några vanliga metorder.

2.1 Dokumentmodeller

En vektorrymdsmodell som är särskilt viktig för statistisk analys av texter är den s.k. *bag-of-words* modellen (även kallad "vector space model". Se t.ex. [Manning och Schütze, 1999] s. 539-544 för mer detaljerad behandling). Namnet kan översättas ungefär till "kasse som innehåller ord"-modellen. Vad betyder det att ha en kasse med ord i?

Tänk dig att du har en text. Texten består av ord. Ta loss alla orden ur texten och lägg dem i en kasse som i figur 14. Då vet du vilka ord texten använde och hur många gånger de orden användes, men du har ingen aning om i vilken ordning orden användes. Precis sådan är bag-of-words modellen: Den håller reda på vilka ord som finns i en text och hur många gånger, men glömmer gladeligen bort i vilken ordning de förekom.

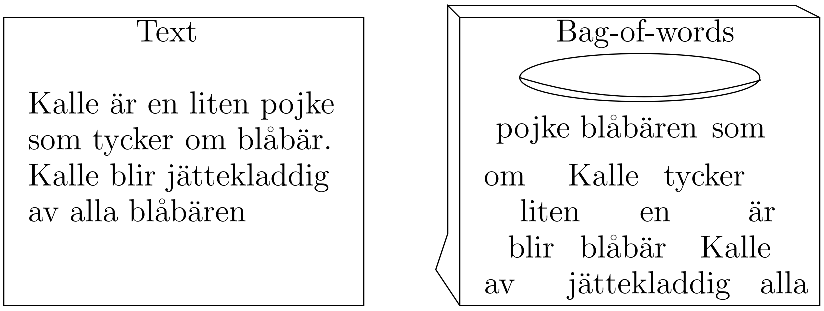
Matematiskt uttrycks texter i bag-of-words modellen som vektorer. Varje komponent i vektorn berättar hur många gånger ett visst ord förekommer. Som exempel kan vi ta texterna:

Text1: "Börje är snäll" Text2: "Börje är artig" Text3: "Urban är medgörlig"

Totalt har vi 6 olika ord. Då får vår vektor 6 dimensioner. Vektorns form blir:

$bagofwords_{dokument} = \begin{bmatrix} \#Börje & \#är & \#snäll & \#artig & \#Urban & \#medgörlig \end{bmatrix}$

Ovan står att första komponenten av vektorn är antalet 'Börje' som fanns i texten, den andra



Figur 14: Tag en text och lägg dess ord i en kasse. Observera hur man fortfarande på basen av kassens innehåll kan vara ganska säker på att texten handlar om Kalle och om blåbär, trots att vi inte vet i vilken ordning orden förekommit.

komponenten är antalet 'är' som fanns i texten o.s.v. Vektorerna för exempeltexterna blir då:

$$bagofwords_{Text1} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$
$$bagofwords_{Text2} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$
$$bagofwords_{Text3} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Varför är en sådan representation så mycket använd? Förlorar man inte en massa information när man glömmer bort ordningen orden haft? Orsakerna är många, men här är de viktigaste.

- Bag-of-words är bra för informationssökning. Om en människa skriver in ordet 'artig' i en sökmaskin så vill han antagligen ha texten 'Börje är artig' som svar hellre än 'Urban är medgörlig'. Detta för att den första innehåller ordet 'artig' och därför sannolikare handlar om artighet
- Det är effektivt att söka liknande texter med hjälp av representationen. Eftersom representationen är en vektor, kan man tänka sig att den motsvarar en punkt i ett koordinatsystem. Då är de punkter som är i närheten mer liknande än de som är långt borta. I praktiken fungerar detta bra, och det finns flera olika metoder för att räkna olika sorters likheter.
- Man förstår inte tillräckligt bra hur språket fungerar för att kunna göra effektiva metoder som nyttjar ordens ordning. Ordningen kan delvis utnyttjas, t.ex. genom att söka kollokationer vars betydelse är idiomatisk (t.ex. "röda torget" syftar till ett specifikt torg, och det är inte bara fråga om en vanlig förekomst av orden röda och torget). Förutom på detta begränsade sätt nyttjas ordningsinformation mer sällan. Därför är det ofta mer praktiskt att lämna bort den.

I praktiken brukar man lämna bort de allra vanligaste orden från bag-of-words vektorerna. Detta för att det inte spelar någon större roll för innehållet i en text om ordet "en" förekommer 23 eller 12 gånger.

2.2 Strategier för stilanalys

När vi betraktar en text, var i den finns stilen? En text består av ord och olika mellantecken som punkt, kommatecken och frågetecken. Men om vi minns vår tanke att innehåll och stil är två ortogonala komponenter i en text, vilka delar av texten är då innehåll och vilka är stil? Eftersom både stil och innehåll är teoretiska begrepp, latenta variabler som vi inte kan observera direkt, kan vi inte besvara frågan fullständigt. Vi är ändå övertygade att det i texten finns något som motsvarar vår intuitiva uppfattning av innehåll och stil. Om dessa aspekter av texten verkligen är ortogonala i strikt matematisk mening vet vi inte säkert. Vi gör detta förenklande antagande för att terminologin ska bli enklare. Men till hurudana representationer leder oss tanken om ortogonal stil och innehåll?

Vi har sett hur man kan representera dokument numeriskt med bag-of-words modellen. Representerar den dokumentets stil eller dokumentets innehåll?

Eftersom den primära användningen av bag-of-words modellen är i sökmaskiner, för informationssökning kan vi dra slutsatsen att bag-of-words är bra för att söka innehåll. Sökmaskiner har vanligen inget stilbegrepp. Det är t.ex. svårt att söka dokument som har samma stil som ett annat, men lätt att söka sådana vars innehåll (eller åtminstone innehållsord) är liknande.

Bag-of-words modellen använder sig av dokumentets ord för att beskriva dokumentet. Vad är det som gör att någon text har en viss stil? Ordvalen, ganska långt, och det syns ju i bag-of-words modellen till en del. Stilen märks också på användandet av olika konstruktioner: "Jag såg Elvis på scenen den kvällen" är av annan stil än: "Elvis sågs på scenen den kvällen". På samma sätt är "Företagets kundtjänst anses vara sämre än önskvärt" annorlunda stilmässigt än "Servicen är helt totalt urusel!!!"

Dessa skillnader tycks inte påverka innehållsorden så mycket, utan påverkar mest förekomsterna av de vanligaste orden, t.ex. de personliga pronomina, prepositionerna och t.o.m. punkternas och utropstecknens antal. Dessa hör till de saker som bag-of-words lämnar bort ur sin representation. De vanligaste orden lämnas ju ofta bort, och punkter och kommatecken tas sällan alls med i analysen.

Hur har andra närmat sig detta problem när de gjort automatisk statistisk stilanalys?

Jussi Karlgren [Karlgren, 2000] använder en representation som inte är riktigt bag-of-words, men väldigt liknande. Förutom orden använder Karlgren s.k. *textstatistiker*, t.ex. antalet personliga pronomen och meningars- eller ords medellängder. Att räkna antalet personliga pronomen är i praktiken det samma som att addera ihop antalet alla sådana i bag-of-words modellen. Med andra ord tillförs ingen ny information, men en känt stilbärande information görs mer framträdande. Karlgren använder också information om ordklasser, vilket kräver syntaktisk analys och därför är utöver bag-of-words modellen. Man kan summera Karlgrens tillvägagångssätt med att säga: Stilen finns i de ord som inte förutspår betydelsen särskilt starkt och i tekststatistikerna.

Aidan Finn och Nicholas Kushmerick [Finn och Kushmerick, 2003] jämför bag-of-words modellen med användadet av ordklasser och textstatistiker. De klassificerar en texts stil mellan objektivt faktum kontra subjektiv åsikt. Bag-of-words visar sig ge de bästa klassificeringsresultaten och textstatistikerna ger de sämsta. Bättre än någon av dessa data enskilda är kombinationen av alla tre. Finn och Kushmerick får som bäst ca 85% noggrannhet i denna klassificering.

Nigel Dewdney m.fl. [Dewdney m.fl., 2001] använder både bag-of-words och textstatistiker. Deras statistiker liknar Karlgrens, men tar dessutom i betraktande olika specialtecken som punkter och kommatecken m.fl., olika "smileys" och textens indentering. Klassificeringsresultaten mellan klasser som "Advertisement", "Frequently Asked Questions", "Reuters Newswire" m.fl. är som bäst 85% för bag-of-words, 87% för textstatistiker och båda kombinerat 92%.

Moshe Koppel m.fl. [Koppel m.fl., 2003a] utvecklar ett automatiskt mått för hur stilbärande en viss egenskap i en text är. De mäter något de kallar stabilitet, som säger hur ofta någon viss egenskap är exakt lika i olika versioner av samma text. De visar också att egenskaper med låg stabilitet är mest stilbärande. Problemet med detta sätt att gå tillväga är att man måste ha flera versioner av texterna till hands. Koppel utvecklar en annan metod för att söka stilbärande egenskaper i texten. Man börjar med alla tänkbara egenskaper och sedan väljer man bort dem som inte delar in träningsdokumenten i rätt klasser. Med denna metod får Koppel en otrolig 98% noggrannhet i klassificering mellan fiktiv och icke-fiktiv litteratur och en 80% noggrannhet i klassificering enligt författarens kön [Koppel m.fl., 2003b].

För ett bra sammandrag av textklassificering i allmänhet se [Sebastiani, 2002].

Från exemplen kan man säga att representationerna är ganska ad-hoc, och inte baserade på några särskilt välmotiverade principer. I praktiken verkar det ändå som att man med hjälp av bag-of-words modellen, förstärkt med textstatistiker kan detektera stilen hos ett dokument tillräckligt bra för att det ska vara nyttigt för praktiska tillämpningar.

Med andra ord kanske man kunde detektera fackspråk vilket är ett delproblem inom Learning to Translate forskningsområdet. Vi bestämde oss att undersöka det lite närmare. Vårt experiment beskrivs i kapitel 5.1.

3 Språkgenererande system

Automatisk språkgenerering innebär att man med hjälp av en dator skapar naturligt språk. En tillämpning är t.ex. att generera väderleksrapporter från numerisk väderinformation, men det kan också vara annat: Enkla återkommande nyhetsartiklar, personifierade nyhetsbrev eller agenter som man kan diskutera med. Som exempel på det sista kan man ta ELIZA. Att generera språk på dessa sätt kallas i den vetenskapliga litteraturen för *Natural Language Generation* eller *NLG* [Bateman och Zock, 2001].

3.1 Genereringssystemens arkitektur

Eftersom människor är mycket skickliga på att generera språk, skulle man gärna härma människans sätt att göra saker när man gör ett språkgenererande system. Då kan man använda psykolingvistiska modeller, d.v.s. teorier om hur människan producerar språk. Men språkgeneratorn måste också vara möjlig att förverkliga med dagens teknik och därför accepterar man att det blir en viss skillnad mellan den teoretiskt korrekta modellen och den man använder i praktiken i automatiska system.

Inom psykolingvistiken är en populär modell att språket produceras av en *modulär struktur* med *självmonitorering*. Det betyder alltså att man antar att det i människans huvud finns flera moduler som kommunicerar med varandra. En modul utför en viss uppgift. Man kan upptäcka moduler eftersom en modul har en struktur där det inom modulen finns många kopplingar, men mellan modulerna finns få. Självmonitorering betyder att människan hela tiden lyssnar på vad hon själv säger, så att hon märker när hon gör fel. Det betyder att alla moduler kollar vad de andra modulerna gör och ger *feedback* åt varann när de andra gör fel.

I stora drag är modulerna indelade i tre nivåer:

1. Den konceptuella nivån konstruerar det som ska sägas i form av begrepp.
2. Den formulerande nivån som omvandlar begreppen till fonetiska signaler, alltså vilka ljud som ska artikuleras
3. Den artikulerande nivån artikulerar de fonetiska signalerna så att det blir ljud, språk som kan höras.

Från den formulerande nivån ges feedback till den konceptuella nivån, genom att man lyssnar på det man säger och översätter det till begrepp som den konceptuella nivån förstår. På samma sätt finns det en koppling via lyssnandet från den artikulerande nivån till den formulerande. För en närmare beskrivning se [Levelt, 1989].

I automatiska system använder man vanligen en lite annorlunda modell: Den enkelriktade modulära pipeline-arkitekturen [Reiter, 1994, Reiter och Dale, 1999].

3.1.1 Den modulära pipeline-arkitekturen

Ordet *pipeline* kan översättas till svenska som "det löpande bandet". Systemens struktur är som ett löpande band, först gör man en sak i det första skedet och skickar resultatet vidare till nästa skede som gör nånting och skickar resultat vidare ända tills det färdiga språket kommer ut i processens slutända.

De olika skeden är i stora drag:

1. Bestämmandet av innehåll (eng. Content Determination) - Man besluter *vad* som ska sägas
2. Meningsplanering (eng. Microplanning eller Sentence planning) - *Hur* ska det kommuniceras, hurudana delar och meningar
3. Realisation - Man skapar den slutgiltiga texten

I det första skedet bestämmer man vad man vill kommunicera. Det kan t.ex. vara dagens väderdata som borde omvandlas till en väderleksrapport, eller resultatet av en sökning efter flygrutter från Helsingfors till New York. Det kan också vara något resultat från ett dataprogram som är svårt att läsa för en människa. T.ex. ett matematiskt bevis gjort av ett dataprogram är ofta komplicerat att läsa för en människa, det är för många mellansteg, det viktiga försvinner bland det triviala.

På basen av den data som ska kommuniceras skapas i första skedet en *dokumentplan* som innehåller den information som ska kommuniceras och informationens struktur (d.v.s. inte den språkliga strukturen, utan t.ex. vilken väderleksstation olika information kommer, eller vilken flygrutt som kräver byte i Paris o.s.v.).

I det andra skedet omvandlas dokumentplanen till menings- och frasplaner. Många innehållsord väljs redan i detta skede men den slutgiltiga texten är ännu inte konstruerad.

I realisationsskedet tar man meningsplanerna och skapar den slutgiltiga texten genom att fylla i funktionsord som 'en', 'ett', 'för', 'efter' o.s.v. Idén bakom att ha detta tredje skede är att man vill göra de två första delarna så oberoende av det specifika språket som möjligt, för att vid behov lätt kunna generera på fler språk. Därför är det först i det tredje skedet man tar ställning till saker som hur man böjer ord och strukturerar meningar grammatikaliskt rätt i språket.

Den modulariska pipeline-arkitekturen är 'felaktig', eller åtminstone överförenklad, ur psykolingvistiskt perspektiv. Den används i språkgenereringssystem för att den är praktisk och enkel att förverkliga. Realistiskt sett kräver inte självmonitoreringen att det finns så många feedback-kopplingar i pipelinen. Med andra ord kan man säga att pipelinen kanske bara är en modell som förenklar den 'rätta' modellen *lite grann*. Ur ingenjörssynvinkel anses pipeline-modellen vara tillräckligt bra.

3.1.2 Skede 1: Att planera innehållet

Före man börjar generera språk måste man ha något att säga. Innehållsbestäming (eng. Content determination) är den del av processen där man besluter vad man ska kommunicera och söker fram det. T.ex. om man har ett program som söker flygrutter så söker man i flygruttsdatabasen för innehållsbestämningen.

När det är klart vad som ska sägas går man över till dokumentstruktureringsskedet (eng. document structuring) där man organiserar textens stycken och meningar. I detta skede tar man ännu ingen ställning till meningarnas inre struktur. Av informationen som ska kommuniceras väljs lämpliga informationscentra som sedan blir stycken i texten. Man strävar efter att göra detta så att det språkliga uttrycket inte begränsas alltför mycket.

Ett exempel för ett sätt att representera dokumentplanen är Schemas [McKeown, 1985]. Schemas beskriver meningsstrukturer som förekommer ofta i texten och en ungefärlig ordning av dessa meningar. Fördelar med Schemas är att man kan förverkliga dem mycket effektivt med hjälp av formella grammatiker och att de är mycket uttryckskraftiga. Tyvärr är Schemas väldigt specifika för det tillämpningsområde man genererar text för. Schemas som utvecklats för väderleksrapporter kan inte användas för att generera flygruttsinformation, eftersom de innehåller så mycket detaljerad information.

Ett alternativ till schemas är att använda traditionell symbolisk AI och formell logik. Ofta skapar man regelsystem baserade på Rhetorical Structure Theory [Mann och Thompson, 1988] om hur en text ska struktureras. Man delar in texten i meddelanden. Dessa meddelanden ges olika egenskaper och man kombinerar meddelandena med hjälp av egenskaperna och kombinationsregler.

Resultatet av hela det första skedet är en *dokumentplan*, vanligen strukturerad som ett träd.

3.1.3 Skede 2: Att planera meningarna

Meningen med detta skede är att ta det föregående skedets dokumentplan och omvandla den till en sekvens menings- och frasplaner. Det innebär sådana saker som t.ex. att skapa rätt syftningar mellan meningarna så att texten inte blir klumpig eller alltför flertydig.

Exempel:
Kalle är en liten pojke. Han har en röd boll som han tycker mycket om.
∗ *Kalle är en liten pojke. Kalle har en röd boll. Kalle tycker mycket om den röda bollen.*
Den senare texten är klumpig för att syftningarna till Kalle återkommer i varje mening.

I meningsplaneringsskedet ska också relaterade saker kombineras:

Exempel:
Kalle har en röd boll och en blå kloss.
∗ *Kalle har en röd boll. Kalle har en blå kloss.*

I detta skede väljs också innehållsorden. I litteraturen kallas det för *lexicalization*. Det betyder att man väljer passande ord för de semantiska begreppen i dokumentplanen. T.ex. värme (5℃) och vind ($10\frac{m}{s}$) i väderleksinformationen ska omvandlas till:
(5℃ varm), (blåsa, 10 sekundmeter)

Som exemplet visas skapar man inte ännu en fullständig mening som: Det är 5℃ varmt och blåser 10 sekundmeter.

Man böjer heller inte innehållsorden ännu. Det är lämnat åt följande fas. Meningsplaneringens output är menings- och frasdefinitioner.

3.1.4 Skede 3: Realisation

De naturliga språken har rikligt med speciella egenskaper, som t.ex. kongruens, böjningsformer, oregelbunden syntax o.s.v. Realisationsskedets uppgift är att ta hand om sådana saker så att de andra modulerna i systemet inte behöver göra det. De menings- och frasplaner man fått, ofta i trädform, ska omvandlas till en sekventiell text. Funktionsorden ska läggas till, orden ska böjas rätt, orden ska ordnas i rätt ordning inom meningarna och ortografiska konventioner, som att börja meningar med stor bokstav, ska läggas till. Som resultat får man nu den slutgiltiga texten.

Ibland vill man också generera strukturerad text med rubriker och tabeller m.m. Sådan struktur läggs till i detta skede.

För realisationen används flera olika tekniker:

- Dubbelriktade grammatiker. Samma grammatik som används för att förstå text används för generering. Detta är teoretiskt elegant eftersom det antas vara så människan gör, och för att man då behöver skapa bara en grammatik. I praktiken är det mycket svårt att göra en dubbelriktad grammatik eftersom den semantiska form som används för förståelse av text är annorlunda den som används för generering.
- En grammatik specifikt för generering. Man skapar en grammatik specifikt för att generera text. Grammatiken baseras på någon av de många grammatikformalismer som finns, t.ex. LFG. Det finns gott om fungerande tillämpningar som använder sig av genereringsspecifika grammatiker (John Bateman: KPML [Bateman, 1997], Michael Elhadad: FUF/SURGE [Elhadad och Robin, 1999], CoGenTex RealPro [CoGenTex, 2000]).
- Template mekanismer. En *template* är en nästan färdig mening eller struktur i vilken man kan byta ett eller några ord. T.ex. "Börje tycker om Lisa" kunde komma från en template "(namn1) tycker om (namn2)" Där *namn*1 och *namn*2 är variabler som ersätts med två olika namn. Ibland är det enklare att göra bra templates än att göra en avancerad grammatik. Alla tillämpningar av språkgenerering behöver inte så varierande språk. Det finns också försök att få system att automatiskt lära sig templates. T.ex. [Ratnaparkhi, 2000] beskriver hur ett system lär sig templates från ett korpus i en tillämpning där man genererar flygtidtabeller. Templates kan också kombineras med grammatiker så att man i varje specifik situation kan använda den metod som passar bäst, grammatik när det behövs och templates annars (Stephan Busemann TG/2 [Busemann, 1996])

3.1.5 Ett exempelsystem: SURGE, en realisationskomponent

SURGE är en färdig realisationskomponent som producerar engelsk text. För att vara maximalt praktiskt användbar på olika tillämpningsområden förstår sig SURGE på många olika grammatikformalismer. SURGE tar som input ett FUF-träd. FUF är en implementation av *Functional Unified Formalism*. FUF-trädet innehåller frasinformation och som löv i trädet är innehållsorden (i grundform). SURGE fyller själv i många språkliga detaljer, t.ex. förstår den att ändra till pluralform om det man talar om har ett antal som är > 1 [Elhadad och Robin, 1999].

FUF-trädet är *kompositionellt*, vilket betyder att olika delar av trädet betyder det samma i olika sammanhang. Ett ord som 'sparka' är kompositionellt för det betyder samma sak både i sammanhanget "- en boll" och "- en sten". Som följd kan man ändra ett påstående, skrivet som ett FUF-träd, till en fråga med ganska få ändringar i trädet.

Genereringen från FUF-trädet går till på följande sätt:

1. FUF-trädet unifieras med grammatiken. Unifikation är en logisk operation där man kombinerar två påståenden så att båda uppfylls. När man unifierar grammatiken med FUF-trädet får man ett syntaktiskt träd innnehållande både FUF-trädets semantik och språkets syntax. I detta skede av processeringen innehåller det syntaktiska trädet alla ord, t.o.m. funktionsord som artiklar, prepositioner o.s.v.

2. Syntaxträdet omvandlas till en sekventiell text. Strukturella aspekter som punkter och kommatecken m.m. läggs till. Orden böjs i sina rätta former.

3.2 Statistiska metoder för automatisk generering

De system baserade på arkitekturen i det föregående kapitlet är till sin natur ofta symboliska. Det finns andra metoder för generering som inte använder sig av pipeline-arkitekturen. T.ex. kan man ta somliga metoder som används för maskinell översättning. Dessa metoder är rent statistiska till sin natur och använder sig inte av någon sorts symbolisk grammatik. Dessa metoder hittar struktur i stora textmassor och använder denna empiriska information som bas för sin språkbehandling.

Statistiska metoder kan också passa in i pipeline-arkitekturen. Irene Langkilde och Kevin Knight observerar att system som SURGE kräver som input lingvistiskt detaljerad information, som inte är så lätt att få tag på. Deras system tillåter att man underspecificerar inputen, bara uttrycker det man vet om meningen. Sedan använder de statistiska n-gram modeller för att välja den mest sannolika realisationen för den underspecificerade inputen [Langkilde och Knight, 1998]. På detta sätt kan statistiska metoder passas in i pipeline-arkitekturen.

Kunde man gå längre än så och göra en generator som lär sig från enbart ostrukturerad text? Språket är oerhört komplicerat och dessutom ofta oregelbundet. Dessutom har vi konstaterat att en kroppslig erfarenhet är en nödvändighet för att förstå ords betydelse. Då kommer lätt tanken att man inte kan lära något av värde från enbart text. Det är trots allt inte riktigt så heller. Språket innehåller mycket struktur trots att den kanske inte följer exakt någon rent symbolisk grammatik. Detta leder till att man med statistiska metoder kan göra ganska många saker som är nyttiga. Ett bra exempel är en internet-sökmaskin, som t.ex. Google. Man fyller i ett ord och maskinen returnerar en mängd sidor som på något sätt är relevanta för det sökordet. I praktiken kan man hitta väldigt mycket nyttig information på detta sätt.

Sökmaskinen är på många sätt ett vettigt sätt att hantera stora mängder ostrukturerad information. Ostrukturerad i detta sammanhang betyder att man inte organiserat artiklarna i någon viss ordning som i t.ex. ett uppslagsverk. Istället använder man sig av den inbyggda strukturen i språket: artiklar som handlar om *presidenten George Bush* innehåller sannolikt orden $george, bush, president$. Detta låter fullständigt trivialt men är i själva verket rätt intressant. Språket är för komplicerat för att man i dagens läge automatiskt ska kunna "förstå" fri text, men språket innehåller trots detta tillräckligt regelbunden statistisk struktur för att informationssökning ska vara effektivt. Denna struktur och dess ursprung är ett värdefullt forskningsområde för den kan lära oss nya saker om vår språkförmåga. Dessutom leder en bättre förståelse av strukturerna till bättre sökmaskiner och andra liknande tillämpningar.

3.2.1 Parallellkorpusmetoder

En parallellkorpus är ett korpus där samma texter finns på flera språk 'parallellt'. Om man t.ex. har ett svensk-engelskt parallellkorpus så finns samma text både på engelska och svenska i korpuset. Ofta strävar man efter att ha sådana parallellkorpusar där det märkts ut vilka av meningarna som motsvarar varandra, d.v.s. vilken mening på engelska som motsvarar en mening på svenska.

Genom att betrakta förhållandena mellan dessa texter på olika språk kan man göra system som kan översätta från det ena språket till det andra. Ingenting hindrar att det ena språket inte är ett naturligt språk. Ehud Reiter undersöker en språkgenerator baserad på en parallellkorpus där det ena språket är engelska väderleksrapporter och det andra språket är motsvarande väderleksdata. Samma teknik har också använts för generering av personliga brev för att hjälpa rökare att sluta röka. Breven var varierade för olika rökare baserat på hur de svarat i en enkät om sin rökning [Reiter m.fl., 2003].

Ett centralt problem för parallelkorpusmetoder är hur man hittar vilka ord och konstruktioner i texten som motsvarar varann. Detta problem kallas *avstämning* (eng. alignment). Ett intressant exempel på detta är Regina Barzilay och Lillian Lee som beskriver avstämning mellan matematiska bevis, sparade i formell logisk notation, med motsvarande bevis i textform. De använder en algoritm, ursprungligen utvecklad för att hitta motsvarande gener i DNA-molekylen, till att hitta vilka former i den logiska representationen som motsvarar ord i texten. Med hjälp av denna information kan man generera textformen för nya matematiska bevis från den logiska representationen [Barzilay och Lee, 2002].

3.2.2 Parafrasmetoder

Samma betydelse kan ofta uttryckas på många olika sätt. Dessa olika sätt kallas parafraser. T.ex. brukar man kolla om någon förstår något genom att be den berätta “med egna ord”, d.v.s. skapa en parafras av det man just sade. Människot skapar parafraser väldigt skickligt och naturligt och därför antas det vara en central del av människans språkliga förmåga.

I fråga om generering är parafraser en slags template-modeller. In i alternativa frasstrukturer placerar man alternativa ord. Man kan upptäcka parafraser automatiskt ur ett korpus som innehåller flera artiklar som motsvarar varandra, t.ex. för att de är olika tidningars reportage om samma händelse. Då samma ovanliga ord förekommer i olika meningskonstruktioner kan man anta att det finns en chans att meningskonstruktionerna är delvis synonyma [Barzilay och Lee, 2003].

I sin doktorsavhandling beskriver Regina Barzilay hur man med hjälp av ostyrd inlärning kan konstruera ett system som hittar parafraser i ett korpus med motsvarande texter. Dessa parafraser används som central del för att generera sammandrag av flera relaterade artiklar. Systemet är en imponerande demonstration av att statistiska metoder kan generera text med en kvalitet som är nyttig i praktiken [Barzilay, 2003].

3.3 Att bygga en automatisk språkgenerator

Om man idag går till en lokal databutik och ber om en språkgenerator får man sannolikt bara en konfunderad blick som svar. Sådana system måste ännu byggas specifikt för varje tillämpning. Ehud Reiter beskriver ett sätt att bygga en generator med följande steg [Reiter och Dale, 1999]:

1. Definiera målsättningar och krav för generatorn. Först samlar man en exempelkorpus av sådana dokument som systemet skulle producera men som hittills skrivits av människor. På basen av denna korpus kan man avgöra vad som krävs av generatorn. Om det inte finns artiklar till hands ber man folk som känner till området att skriva sådana.

2. Analys av exempeltexterna. För att kunna konstruera en generator måste man veta varifrån informationen i texterna kommer. Exempeltexternas delar klassificeras på följande sätt:

   - Text som inte förändras
   - Direkt tillgänglig information
   - Information som går att räkna ut
   - Information som saknas

   Den information som saknas måste åtgärdas på något sätt. Antingen måste informationen bli tillgänglig, eller sedan måste den delen av texten författas av någon människa.

3. En korpus av måltexter:

    - Ta bort delar som baserar sig på information som saknas.
    - Definiera vilka delar som ska författas av en människa.
    - Förenkla vid behov texternas struktur för att gör genereringen enklare. Förbättra den ursprungliga texten och se till att texterna är enhetliga, trots att de kanske är skrivna av olika individer.

4. Funktionell definition. Gör en noggrann beskrivning av vilka delar en människa ska skriva och av strukturen på det som produceras automatiskt. Bestäm även hur inputvärdena får variera, d.v.s. hur stora variationer systemet ska kunna klara av.

5. Fråga dig: Lönar det sig med NLG eller räcker templates? Ibland är templates tillräckligt bra och betydligt enklare. Det är lättare att hantera den språkliga variationen m.h.a. NLG, men det tar längre tid att få systemet färdigt.

3.3.1 Varifrån kommer information som ska kommuniceras genom språkgenerering?

Ett system som producerar språk behöver rätt information för att kunna fungera. Den information som behövs kan indelas i tre delar [Reiter m.fl., 2003]:

- Information om tillämpningsområdet. T.ex. väderdata som: Det regnar i Kuopio. Förutom ren data behövs också kunskap om hur datan ska tolkas. Vilka regler gäller inom tillämpningsområdet? T.ex. om vi vet att det regnar i Paris, påverkar det sannolikheten att det regnar i Helsingfors? Nej, inte åtminstone särskilt mycket. Däremot om det är flygstrejk i Paris kan det bra påverka trafiken i Helsingfors.

- Kommunikationskunskap inom tillämpningsområdet. Hurudan terminologi används? Ska man säga "uppehållsväder" eller "skiner solen"? Är det "milt" eller "ljummet"? Vilken information är viktig så att den bör kommuniceras? Om det regnar i Kuopio, vill Helsingforsborna veta det?

- Kommunikationskunskap. Sådant som grammatik, punkter, kommatecken, stora och små bokstäver. Kommunikationskunskapen motsvarar realisationskomponenten i pipeline-arkitekturen.

Av dessa tre är informationen om tillämpningsområdet och kommunikationskunskapen sådana att de potentiellt kan återanvändas i andra system. Den förra för att kunskap potentiellt kan representeras på samma sätt för olika tillämpningar, den senare för att språkets regler i stort är de samma för olika sorters text. Däremot är kommunikationskunskap inom tillämpningsområdet beroende av både tillämpningsområde och språk och är därför så gott som alltid specifik för tillämpningen.

Problem med informationsbehandling

- Komplexitet. Tillämpningsområdet innehåller oftast mycket olika sorters specialfall och konstigheter som inte kommer fram i en ytlig analys.

- Om uppgiften är ny. Man strävar inte alltid efter att automatisera en uppgift som människor utför. Om det inte finns folk som känner området, finns det inte heller någon korpus. Informationen måste fås fram genom vetenskapliga metoder.

- Det finns inga bra modeller för områdets information. Om det finns starka teoretiska modeller kan man använda dem för att styra genereringssystemet åt rätt håll. Många områden saknar sådana modeller. Om man tänker på de personliga breven för att hjälpa rökare sluta, så finns det inte detaljerade modeller för hur man ska få människor att förändra sitt beteende.

- Flertydighet. Experter är av olika åsikter i många frågor. Detta märks också när man gör en korpus för tillämpningsområdet. Särskilt illa är att man från exempeltexterna inte kan veta hur säker på specifika saker den som skrivit texten är.

3.3.2 Metoder att få bättre information för generering

Huvudsakligen finns det två familjer av metoder för informationsinskaffning: Att nyttja experters kunskap eller att utvinna kunskapen ur en korpus. Båda riktningarna har för- och nackdelar och det är ofta en bra idé att kombinera metoder från båda riktningarna.

Expertmetoder En expert är någon som känner till tillämpningsområdet. Expertens kunskap är något som finns inne i honom eller henne. Den kan med andra ord inte användas direkt. Därför används diverse metoder för att få fram kunskapen som experten besitter.

- Att fråga experten ger en bra överblick, men ofta glöms också viktiga saker bort.
- Strukturerade expertmetoder. Man använder sig av något systematiskt arbetssätt, t.ex. en strukturerad intervju eller ber experten tänka högt medan han eller hon utför en uppgift. Idén bakom dessa metoder är att man ska se hur experten utför uppgiften. Om man jämför en strukturerad undersökning med att bara fråga experten ger den strukturerade undersökningen en betydligt mer realistisk bild av uppgiften. Ett problem är att variationen mellan experter är stor och för att få bra täckning borde man göra väldigt många strukturerade undersökningar.
- Expertfeedback på de genererade texterna. När det språkgenererande systemet är färdigt är experternas feedback mycket nyttigt för lösa specifika problem. För mer allmäna problem är denna metod i regel inte särskilt effektiv.

Korpusbaserade metoder Korpusbaserade metoder antar att den viktiga informationen kan fås fram genom statistisk analys av stora textmassor. Ett fenomen som är speciellt problematiskt för språkgenererande system är individuell variation. Den statistiska variationen borde tas i betraktande i modellerna istället för att man bara räknar med genomsnitt, annars producerar systemet lätt nonsens. Ehud Reiter berättar om ett exempel där väderleksrapportgeneratorn använde konstiga siffror för vindhastighet. Generatorn använd “06” för 6 men “5” för 5. Det här berodde på att vissa av dem som skrivit texterna i korpusen ogillade ojämna tal och därför avrundade dem till jämna tal. Dessa personer skrev också siffrorna med en inledande nolla (“06”). Därför fanns det mer jämna tal i korpusen och den automatiska analysen hittade inte en enhetlig form för siffrorna. Det är svårt att bereda sig för alla dylika företeelser. Ett annat, kanske ännu värre, problem är felaktigheter i korpusen. T.ex. sparade väderleksrapportören ibland tid genom att lämna bort någon tidpunkt i rapporten. När denne senare granskade sin rapport erkände han att det var ett felaktigt handlingssätt [Reiter m.fl., 2003].

Det bästa med korpusbaserade metoder är att man kommer nära den praktiska tillämpningen och att man kan studera detaljer tillräckligt noggrant. Bäst fungerar det då man har ett stort, enhetligt korpus som innehåller korrekt information. Men också annars kan korpusen vara en nyttig källa för kunskap om hur man ska kommunicera olika saker, t.ex. använde Reiter på ett lyckat sätt ett korpus för att lära sig hur man ska uttrycka vindhastigheter i text [Reiter och Sripada, 2003].

Rekommenderad metodologi Det rekommenderade sättet att gå till väga när man bygger ett språkgenererande system är att börja utvecklingen av systemet med att fråga experterna. På basen av deras svar bygger man en prototyp av systemet. När det är klart fortsätter man utvecklingen av systemet m.h.a. strukturerade expertmetoder och korpusanalys. Det slutliga systemet konstrueras och värderas med experternas feedback.

I regel bör man alltid evaluera resultaten från en analys med en metod från den andra metodfamiljen.

3.3.3 Automatiska språkgenereringsmetoders nuvarande läge

Genereringsmetoderna är traditionellt symboliska, men de flesta nyare metoder är mer statistiska till sin natur. Huvudsakligen är det ändå så att i dagsläget är de statistiska metoderna mer begränsade till en viss uppgift än de symboliska. Parallelkorpusmetoderna genererar bara översättningar och parafrasmetoderna vanligen bara sammandrag. Det är trots detta möjligt att lyckat ersätta vissa traditionellt symboliska komponenter med statistiska motsvarigheter, som Regina Barzilay och Lillian Lee [Barzilay och Lee, 2002] visar.

Det är antagligen på sin plats att fråga sig om den traditionella pipeline-arkitekturen är den bästa möjliga för de statistiska metoderna? Skulle det t.ex. vara fördelaktigt att kombinera olika nivåers kunskap i en enda statistisk modell, så som i de psykolingvistiska modellerna?

Modellerna av världen och tillämpningsområdet görs också vanligen symboliskt. De symboliska metoderna för detta är visserligen välutvecklade men också deras begränsningar är välkända, som vi beskrivit i de inledande kapitlen. Därför skulle det vara intressant att se alternativa metoder tillämpade också i genereringssammanhang. Skulle det gå att härleda en genereringsmodell enligt Gärdenfors trenivåers arkitektur från kapitel 1.4.5? Och kunde man tillämpa liknande metoder som Narayanan använt för förståelse av metaforer på generering av metaforisk text [Narayanan, 1999], som vi nämnde i samma kapitel? Dessa är bra frågor värda att ställas.

Det görs också en skarp delning mellan information om tillämpningsområdet och språkkunskap. Man kan ifrågasätta om det är det bästa tänkbara att välja orden i den genererade texten mest på basen av språklig kunskap och inte tillämpningsområdets kunskap. En alternativ lösning skulle vara att försöka göra kunskapsmodellen sådan att den innehöll ord som begrepp som kunde vara med redan på den semantiska nivån. Nu väljer man ord som motsvarar begreppen på den semantiska nivån. Då skulle man inte behöva välja ord för begreppen på den semantiska nivån, utan orden skulle finnas med redan på den semantiska nivån och delta i den semantiska modellen. Detta skulle vara mer i linje med Gärdenfors konceptuella rymder [Gärdenfors, 2000] och kräver därför att man använder konceptuella rymder som en semantisk teori istället för en som baserar sig på formell logik. Det finns inte ännu särskilt mycket forskning baserat på den kognitiva vetenskapens resultat inom generering. Genereringen som område är starkt AI-influerat.

Som sammandrag kan man säga att det håller på att ske ett skifte bland de automatiska språkgeneratorerna från symbolism mot ett mer statistiskt håll. De statistiska metoderna har ännu inte hittat ett gemensamt ramverk som de symboliska har. Bristen av referensram beror antagligen på de statistiska metodernas omogenhet. Därför är det ännu för tidigt att säga åt vilket håll vågen kommer att luta.

3.4 Evaluering av genererad text

Ett stort problem med språkgenerering är hur man ska avgöra om genererade texter är bra eller inte. Allra mest praktiskt skulle det vara om man kunde definiera något slags automatisk procedur som skulle ge poäng åt en given text. Dessa poäng borde motsvara en allmän uppfattning om hur bra språket i texten är. Att skapa en sådan process med dagens bristfälliga förståelse av språket är tyvärr inte möjligt.

Det är dock möjligt att kringgå detta problem. Man kan låta människor använda den genererade texten för att utföra någon uppgift. Sedan mäter man hur bra de klarar uppgiften. Det man då mäter är beroende av hur bra den genererade texten kommunicerar med människan som utför uppgiften. Detta kallas *extrinsic evaluation*. Fördelan är att man får en objektiv bild av hur kommunikativ texten är. Nackdelen är att det är mycket arbetsdrygt att utföra ett sådant experiment. Man måste hitta på en bra uppgift för försökspersonerna att utföra, man måste samla försökspersonerna, övervaka experimentet och sedan ännu samla och tolka resultaten.

Förutom extrinsic evaluation finns något som kallas *intrinsic evaluation* vilken handlar om att undersöka egenskaper i texten själv [Bontcheva, 2003]. T.ex. finns det automatiska metoder för att avgöra hur lättläst en text är. Ett mycket vanligt mått är *Flesch*-måttet, som baserar sig på ordens och meningarnas längder i texten. Man kan också jämföra ordförråd och ordval genom jämförelser med andra texter. Sådana statistiker kan användas för att automatiskt ytligt betrakta hurudant språk en generator skapar. Olika statistiska mått finns samlade av Srinivas Bangalore m.fl. i [Bangalore m.fl., 2000].

Man kan också evaluera genererad text genom att använda program som kontrollerar grammatiken i ordbehandlingsprogram. Det är långt från fullkomligt, men kan vara en bra metod att hitta enkla fel.

Texternas kvalitet måste avgöras av en människa, men som hjälpmedel kan man använda automatiska metoder baserade på enkla statistiker och grammatikalisk kunskap.

4 Emergent struktur

Vi upptäckte att det finns regelbunden struktur i texter. Ett sätt att göra sådan struktur synlig är genom ostyrd inlärning. När man tillämpar ostyrd inlärning på korpus av naturligt språk får man representationer som återspeglar textens struktur och ibland har tydligt emergenta egenskaper. T.ex. kan syntaktiska eller semantiska kategorier framträda ur enbart statistiker om ordens kontext. Vi betraktar först forskning gjord med SOM, tar en liten titt på grammatikinlärning mot bakgrunden av ostyrd inlänning och riktar sedan in oss på relativt ny forskning av ord i en text med hjälp av ICA.

4.1 Språkforskning med SOM

Här är några axplock av språkforskning med SOM. Mycket mer har gjorts, men dessa exempel kan ge en känsla av ungefär vad analyserna oftast går ut på.

Vi nämnde Timo Honkelas experiment med bröderna Grimms sagor i kapitel 1.3.2. Där såg man hur kategorier, liknande dem som används inom språkvetenskapen, framträdde på en självorganiserande karta när den fick ordens kontexter som input. Analyser liknande som Honkelas har också gjorts för språk vars grammatik är känd. I [Ritter och Kohonen, 1989, Kohonen, 1990] rapporterar Helge Ritter och Teuvo Kohonen om hur den självorganiserande kartan hittade semantisk struktur då den klassificerade ord som genererats ur en enkel grammatik. Språkets ord klassificerades sedan enligt sin kontext och semantiska förhållanden mellan orden framträdde på kartan. De stora kategorierna substantiv, verb och adverb har var sitt större område. Dessa områden innehåller vart och ett mindre områden som motsvarar underkategorier.

På basen av de självorganiserande kartornas förmåga att organisera data skapades WEBSOM. WEBSOM är en sökmaskin för stora artikelsamlingar, t.ex. webben. Dokumenten i WEBSOM representeras på ett ganska intressant sätt. Först görs en ordkarta på basen av ordens kontexter, som presenterades ovan (se [Ritter och Kohonen, 1989] för noggrannare beskrivning). Varje ord hamnar i någon kategori, med liknande ord enligt den nod på kartan ordet aktiverar. Sedan kodas varje dokument i sökmaskinens index så att man använder noderna i ordkartan, som motsvarar kategorier av ord, som komponenter för en vektorrymdsmodell. Man summerar alltså antalet förekomster av alla ord som aktiverar samma nod, och låter det vara en komponent i en ny vektor. M.a.o. använder man alltså ordens semantiska kategorier, som den självorganiserande kartan hittat för att beskriva dokumenten. Detta sänker dokumentvektorernas dimension och man kan därför använda en större vokabulär i indexeringen än man annars kunde [Honkela m.fl., 1996, Kaski m.fl., 1998].

WEBSOM skiljer sig från de flesta sökmaskiner genom att presentera alla dokument visuellt på en karta. När man söker med ett sökord, inringas de noder på kartan där sökordet förekommer. Oftast finns sökresultaten motsvarande ordet på ett visst område på kartan, men ibland är de spridda på helt skilda platser på kartan, vilket tyder på att dokumenten då inte liknar varandra särskilt mycket. Det kan betyda att sökordet används i flera olika sammanhang, eller att sökordet är dåligt och förekommer i för många artiklar. Den visuella aspekten gör WEBSOM till en ganska speciell sökmaskin, en intressant tillämpning av den topologiska aspekten hos de sjävorganiserande kartorna.

Det är inte bara möjligt att upptäcka syntaktiska kategorier. Ord som förekommer i en viss kontext har en tendens att betyda samma sak som andra ord som förekommer i samma, eller liknande, kontext. Då kan man upptäcka också semantiska kategorier. Krista Lagus m.fl. analyserar [Lagus m.fl., 2002] finska verb genom att göra självorganiserande kartor baserade på verbens kontexter. De använder olika egenskaper i texten för att se vilka som mest motsvarar en handgjord kategorisering. De olika egenskaperna är omgivande verb, omgivande verbkategorier och omgivande morfosyntaktiska egenskaper. Av dessa ger de morfosyntaktiska egenskaperna en klassificering som är klart närmast den handgjorda. De självorganiserande kartorna visar en tydlig semantisk struktur, t.ex. ett hörn av den morfosyntaktiska kartan innehåller ord för positiv kommunikation av känslor, medan ett annat hörn innehåller ord för aggressiv och destruktiv användning av makt.

De självorganiserande kartorna är användbara som instrument i språkforskningen eftersom de både skapar kategorier och en lågdimensionell projektion av komplicerad data på en gång. Detta gör att man kan använda kartorna till hastig, utforskande, analys av stora textmängder. I fall av ord kan kartan visa både hur olika ord förhåller sig till varandra, och samtidigt vilka egenskaper i datan som leder till det beteendet. Dessa egenskaper kommer också vi att utnyttja i våra experiment.

4.2 Grammatikinlärning

Vi nämnde i kapitel 1.1.4 om E. Mark Golds resultat att man inte kunde lära sig en kontextberoende grammatik från bara exempel. Trots detta resultat finns det ändå sätt att maskinellt hitta grammatikalisk struktur i textkorpus. För att kunna hitta denna struktur krävs att man gör ytterligare antaganden om grammatiken. Om man söker en grammatik som inte bara är kontextberoende, utan begränsar den enligt något kriterium som är fördelaktigt, kan man få användbara grammatiker. T.ex. kan man begränsa grammatiken så att man försöker hitta kategorier lika dem som människor använder. Alternativt löser man något praktiskt problem med den resulterande grammatiken och avgör på basen av resultatet hur bra grammatiken var. Att söka grammatiken ur ett korpus kallas grammatikinlärning. Vi går inte in i området i detalj, men beskriver två exempelalgoritmer som används för att ge en liten inblick i hur man kunde konstruera en emergent grammatikmodell.

Menno van Zaanens ABL, Alignment Based Learning ([van Zaanen, 2002, van Zaanen, 2000]), är en grammatikinlärningsalgoritm som kan tillämpas på ett korpus av ostrukturerad text. Information om ordklasser eller dylikt behövs (och används heller) inte. Principen bakom ABL är hämtad från Zellig Harris (vars mest kända elev är Noam Chomsky f.ö.). Fritt översatt säger Harris: "två konstituenter av samma typ kan ersättas med varandra" [Harris, 1951]. Det betyder alltså att om t.ex. två ord har samma ordklass, så kan de ersättas med varandra och man får fortfarande en korrekt mening.

ABL hittar delar av meningar som är likadana. De varierande delarna av övrigt likadana meningar blir förslag på nonterminaler i en kontextfri grammatik. De likadana och skiljaktiga delarna hittas med hjälp av editeringsavstånd. Det betyder att man räknar hur många tillägg, raderingar och substitutioner som behövs för att ändra den ena textsträngen till den andra. Man kan variera editeringsavståndet som kostnadsfunktion genom att ge olika vikter åt de olika operationerna, t.ex. kan man välja att raderingar kostar dubbelt så mycket som tillägg, eller tvärtom. Med hjälp av dessa mått hittar ABL en grammatik som på många sätt liknar handgjorda klassificeringarna.

En annan grammatikinlärningsalgoritm är ADIOS ([Solan m.fl., 2004, Edelman m.fl., 2003]). ADIOS baserar sig på tanken att det finns redundant information i naturligt språk. Genom att komprimera texten kan man hitta generaliserande struktur. Komprimeringen sker genom en grafmodell där orden i texten fungerar som noder. Konstruktioner som förekommer i likadana kontexter bedöms höra till samma klass, även här följande Harris idéer. Modellen prestanda är rätt bra. Algoritmen testades med CHILDES-testet, som används för att avgöra hur bra niondeklassister är på engelska. Det fungerar så att eleverna väljer ett av tre ord som ska passa in i en given konstruktion (T.ex. "John came to ____ me", 1. seeing 2. see 3. saw). ADIOS fick betyget "intermediate" på CHILDES-testet, vilket är ett bra betyg då ett program tävlar mot människor.

En särskilt intressant aspekt av ABL och ADIOS är att modellerna lätt kan användas för generering. Generaliseringen som uppstår när man grupperar ord möjliggör nya meningar som inte förekom i inlärningstexten. Hur en sådan grammatikmodell kan kopplas ihop med ett verkligt genereringssystem är en öppen fråga. Visst kan modellerna ganska långt generera syntaktiskt korrekt språk, men hur ska man få dem att kommunicera det man vill? Man skulle kunde antingen koppla representationen till pipeline-arkitekturen, vilket inte är trivialt, eller sedan kunde man specificera innehållet på något sätt som lättare skulle kunna kopplas till modellen.

4.3 ICA och ordklasser

När man tillämpar självorganiserande kartor på ordkontexter får man en klassificering av orden som liknar traditionella, språkvetenskapliga ordklasskategorier. Men många ord har flera olika betydelser i en text trots att deras ortografiska form är den samma. T.ex. engelskans "can" betyder både "kunna", "konservera i burk", "konservburk" och "kanna". De två första betydelserna är verb och de två senare är substantiv. När vi alltså samlar genomsnittskontexten för ordet "can" får vi ett medeltal som innehåller en summa av alla dessa olika betydelser. Intuitivt fungerar ICA så att man får fram oberoende komponenter ur en summa. Så om vi tillämpar ICA på kontexterna hoppas vi få en signal som motsvarar t.ex. substantivkontext, och en som motsvarar verbkontext o.s.v. Timo Honkela m.fl. utför ett sådant experiment [Honkela m.fl., 2003, Honkela m.fl., 2004], och får mycket riktigt komponenter som liknar olika ordklasser och semantiska egenskaper. T.ex. framträdde komponenter som motsvarade adjektiv, possesiva pronomen. Men inte bara traditionella klasser, utan också andra grammatikaliska egenskaper som pågående handling (eng. suffix -ing) och substantivs pluralformer. Också komponenter som motsvarade semantisk samhörighet framträdde. Orden "adaptive", "artificial" och "cognitive" är exempelord, för vilka en viss komponent har ett stort värde, och de är alla ord som har sitt ursprung i ett gemensamt forskningsområde.

Jaakko Väyrynen m.fl. vidareutvecklade analysen genom att jämföra komponenterna med kategorier som lagts till manuellt till korpusen [Väyrynen m.fl., 2004]. De jämför sedan ICA med en annan populär metod för att hitta klasser för ord, LSA (Latent Semantic Analysis) [Deerwester m.fl., 1990]. LSA är en metod som baserar sig på PCAs reduktion av dimensionalitet. LSA har tillämpats på många olika sorters problem, främst inom informationssökning. Enligt Väyrynens resultat är ICA betydligt bättre på att hitta tilltalande grupperingar av orden. Väyrynens resultat tyder på att ICA kunde tillämpas på de samma problemen där LSA tillämpats, men med ännu bättre resultat.

5 Experiment

5.1 Stilanalys med SOM

Vi undersökte om de strategier beskrivna i 2.2 fungerar också för den stilanalys vi vill göra, d.v.s. skilja mellan och analysera olika sorters fackspråk.

Vi samlade ett korpus av tre olika sorters texter om samma ämne: Vetenskapliga artiklar om autoimmunsjukdomen Keliaki från PubMed-databasen, diskussion från en e-maillista bland människor som antingen är keliakiker, eller känner sådana och populärvetenskapliga artiklar om sjukdomen från www.celiac.com. Av de vetenskapliga artiklarna hade vi 354 st, 48 st från e-maillistan och 6 st av de populärvetenskapliga.

Vi undersökte hur bra en självorganiserande karta separerade de olika stilarnas artiklar. Först genom att göra kartan med bag-of-words och sedan med textstatistiker. Den självorganiserande kartan användes inte som klassifikator, trots att det antagligen också skulle vara möjligt, utan som en visualisering för att se om de olika representationerna skilde dokumenten från varann enligt klass eller på något annat sätt. Eftersom kartan organiserar dokumenten så att de som liknar varandra är bredvid varann på kartan önskar vi att de olika klassernas texter skulle bilda varsitt område på kartan. Det skulle betyda att man på basen av representationen av dokumentet kan veta vilken klass dokumentet tillhör. Om, å andra sidan, klasserna blandar sig med varandra på kartan betyder det att representationerna inte skiljer dokumenten enligt klasserna.

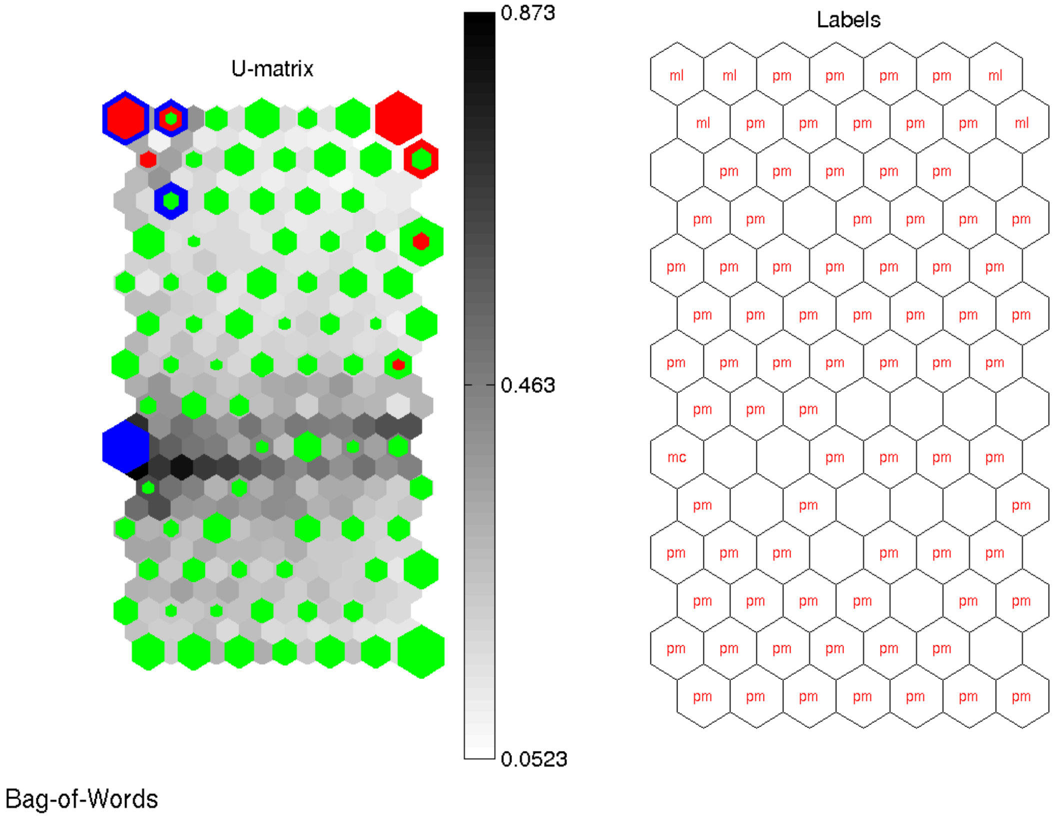
I figur 15 är en självorganiserade karta gjord med bag-of-words som representation för artiklarna. På högra sidan visas vilken klass av artiklar som upptar olika delar av kartan. “pm” är vetenskapliga artiklar från PubMed, “ml” från e-maillistan och “mc” de populärvetenskapliga artiklarna. Till vänster är en annan visualisering av det samma, grön motsvarar “pm”, röd “ml” och blå “mc”.

Färgerna på vänstra sidans bild motsvarar hur mycket en viss klass’ artiklar förekommer på det stället på kartan. T.ex. i högra nedre hörnet finns det mycket grönt, alltså vetenskapliga artiklar. Man kan se att färgerna blandas och upptar samma kartnoder. Dessutom har de olika färgerna inte några egna områden. Detta betyder att bag-of-words representationen inte skiljer dokumenten enligt dessa klassificeringar särskilt bra.

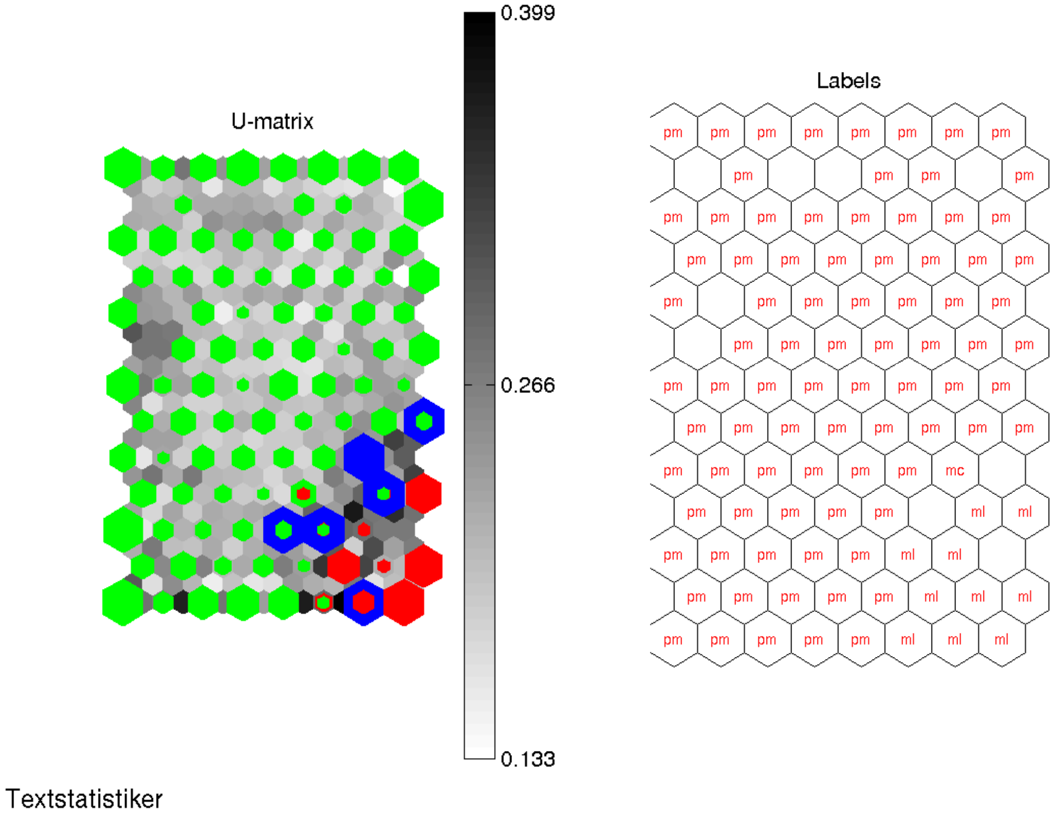
I figur 16 ser vi en motsvarande karta, men den är gjord med en annan dokumentrepresentation. Vi nämnde i kapitel 2.2 att stilen finns mer i funktionsord och mellantecken än i innehållsorden. Därför valde vi 20 textstatistiker (se tabell 1) plus de 50 vanligaste orden som dokumentrepresentation. Den självorganiserande kartan visar en betydligt vackrare klassindelning. Det gröna är för sig och det röda för sig, med det blåa mitt emellan. Dessutom ser man i den vänstra bilden att kartans gråa delar är mörkare vid gränsen mellan det röda och det gröna. Det betyder att dessa noder i kartan är längre ifrån varandra i den ursprungliga representationen än de noder som är på ett ljusare område. Kartan visar på detta sätt egenskaper i den 70-dimensionella representationen som inte lätt kan visas i två dimensioner. Man kan dra slutsatsen att denna representation betydligt bättre skiljer på olika stilar än bag-of-words också i uppgiften att skilja fackspråk och vardagsspråk.

Andelen nummertecken	Medellängd på meningar	Medellängd på ord	.
,	!	?	:
;	’	“	-
(	)	%	/
=	+	[	]

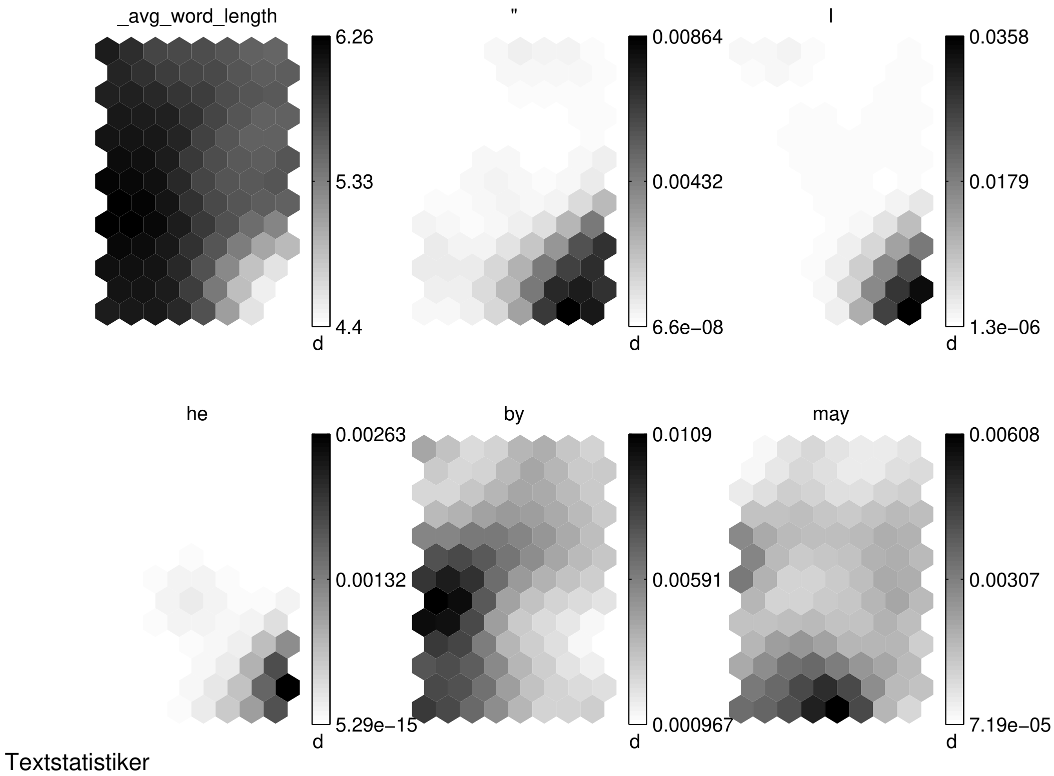
Tabell 1: Förutom de 50 vanligaste ordens relativa antal användes de textstatistiker och antalen av de specialtecken som finns i tabellen för att göra kartan i figur 16.



Figur 15: Stilanalys med bag-of-words. På högra sidan är vilken klass av artiklar som upptar olika delar av kartan. “pm” är vetenskapliga artiklar från PubMed, “ml” från e-maillistan och “mc” de populärvetenskapliga artiklarna. Till vänster är en annan visualisering av det samma, grön motsvarar “pm”, röd “ml” och blå “mc”. Texten i en viss nod motsvarar den klass som mest förekommer i nod. Klasserna blandas på kartan, en klass upptar inte något enskilt område, utan olika färger finns på många delar av kartan.



Figur 16: Stilanalys med textstatistiker. På högra sidan är vilken klass av artiklar som upptar olika delar av kartan. “pm” är vetenskapliga artiklar från PubMed, “ml” från e-maillistan och “mc” de populärvetenskapliga artiklarna. Texten i en viss nod motsvarar den klass som mest förekommer i nod. Till vänster är en annan visualisering av det samma, grön motsvarar “pm”, röd “ml” och blå “mc”. Nu organiseras klasserna till egna områden. “ml” i nedre högra hörnet, “pm” på de övriga områden av kartan och “mc” mittemellan “ml” och “pm”.



Figur 17: Textstatistiker som separerar kategorierna olika bra. För att se vilken klass ett område motsvarar, se figur 16. Den första bilden visar ordens medellängd som är stor för "pm" och liten för "ml", de tre följande egenskaperna har motsvarande struktur men tvärtom. De två sista skiljer inte åt klasserna, utan varierar på annat sätt och är därför dåliga stilrepresentationer.

I figur 17 ser vi hur de olika dimensionerna i den 70-dimensionella representationen varierar över kartan. Det är alltså samma karta som i figur 16. Först har vi ordens medellängd uppe till vänster (_avg_word_length). Mörk färg motsvarar stort värde hos variabeln. I det nedre högra hörnet är ordens medellängd kort och på vänster sida är orden längre. Från figur 16 vet vi att i högra nedre hörnet finns e-maillistans artiklar och till vänster PubMed-artiklarna. Orden i PubMed-artiklarna är alltså längre i medeltal.

De tre följande är '“', 'I', 'he'. De har motsatt struktur i jämförelse med ordens medellängd. Dessa är vanliga på e-maillistan men ovanliga i PubMed. Dessa fyra första variabler skiljer alltså bra mellan stilarna. Som vi ser, gränsen mellan deras ljusa och mörka område är precis där var klassernas gränser går i figur 16.

De två sista är med som exempel på ord som inte delar artiklarna enligt klasser. Orden 'by' och 'may' förekommer i alla tre sorters artiklar och deras ljusa och mörka områdens gränser motsvarar inte alls klasserna.

Vi kan alltså se att textstatistikerna och vissa ord är bättre på att skilja mellan fackspråk och vardagsspråk än bag-of-words. Detta är inte så överraskande med tanke på att bag-of-words är tänkt att representera innehåll, inte stil. Resultaten tyder på att det är helt möjligt att med automatiska metoder baserade på väl valda textstatistiker klassificera artiklar enligt hurudan publik de är skrivna för, åtminstone i fall som detta.

5.2 Analys av Shakespeares sonetter med SOM

En sonett är en dikt med viss, välstrukturerad form. Det finns olika sorters sonetter. Ursprungligen härstammar sonetterna från Sicilien där de uppkom under 1200-talet, och senare spreds de till alla västländer. Shakespeares sonetter har en form som består av 14 strofer med en regelbunden rimstruktur. Först tre stycken bestående av fyra strofer var och en avslutande två strofers del. Rimmen går enligt följande mönster: *abab cdcd efef gg.*

En exempelsonett: *III*

Look in thy glass and tell the face thou viewest
Now is the time that face should form another;
Whose fresh repair if now thou not renewest,
Thou dost beguile the world, unbless some mother.
For where is she so fair whose unear'd womb
Disdains the tillage of thy husbandry?
Or who is he so fond will be the tomb
Of his self-love, to stop posterity?
Thou art thy mother's glass and she in thee
Calls back the lovely April of her prime;
So thou through windows of thine age shalt see,
Despite of wrinkles this thy golden time.
But if thou live, remember'd not to be,
Die single and thine image dies with thee.

I sonetterna inträffar en semantisk vändpunkt mellan de tre fyra raders stroferna och de avslutande två raderna. Vändpunkten kallas *volta*. Den innebär att innehållet i dikten är annorlunda efter volta är före. T.ex. kan de två sista raderna summera av den övriga dikten, eller också kan stämningen ändra kraftigt. I exempeldikten ovan ser vi hur stämningen ändrar till väldigt allvarlig efter voltat. Vi undersöker om man kan hitta semantiska vändpunkter, som voltat, med hjälp av självorganiserande kartor. Resultaten beskrivs i [Kohonen m.fl., 2005].

Hur relaterar detta experiment till emergenta representationer eller språkgenerering? Dikter och poesi har i regel betraktats som mycket underligt språk ur språkforskningens ögon, bl.a. eftersom dikter ofta kringgår grammatikaliska regler för att t.ex. rimma rätt. Men om vi vill studera språket med alla dess olika stilistiska former som ett enhetligt fenomen, så kan det vara intressant om man hittar struktur med samma metoder i poesi som i prosa.

5.2.1 Data och Metoder

Vårt korpus bestod av 154 sonetter av Shakespeare. Av dessa följde två stycken inte den 14 strofers formen och därför inkluderades de inte i analysen (nummer 99 har 12 och nummer 126 har 15 rader). Vi delade in dikterna i ord genom att ändra texten till små bokstäver, ta bort alla specialtecken förutom apostrofen och sedan dela vid varje mellanslag. Det som då återstår är en sekvens ord med sina morfologiska former bevarade (t.ex. *beauty's*). I sin helhet bestod sekvensen av 17 000 ord i löpande text (eng. tokens), och 3 300 olika ord (eng. types). För vår analys valde vi att undersöka ord som förekom tio gånger eller mer, totalt 228 stycken.

Vi hade som målsättning att analys av ordens fördelningar i sonetten kan avslöja semantiska vändpunkter. För att kunna göra analysen med självorganiserande kartor måste man först omvandla ordrepresentationen till en vektor. Vi experimenterade först med en 14-dimensionell vektor, så att ordets frekvens på varje rad i sonetten analyserades. T.ex. om ordet "for" förekom två gånger på första raden, tre gånger på femte, fyra gånger på tolfte och en gång på fjortonde och inte annars får vi följande representation:

$v = [2, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 4, 0, 1]$.

Tyvärr ledde denna representation inte till särskilt bra visualisationer, eftersom datan då var helt för "gles", d.v.s. mycket nollor och lite information. Man borde i såfall haft åtminstone tusen sonetter istället för bara 154. Som lösning skapade vi en representation som följde sonettens form, styckena på fyra strofer och det avslutande strofparet. Den nya representationen har fyra dimensioner: De första tre dimensionerna motsvarar summan av ordets frekvens på fyra rader var, och den fjärde dimensionen är summan av de två sista stroferna, multiplicerat med två så att ordmängderna ska bli lika stora för både två och fyra rader. Om vi kodar om order "for" ovan får vi:

$$v = [2 + 0 + 0 + 0, 3 + 0 + 0 + 0, 0 + 0 + 0 + 4, 2 * (0 + 1)]$$
$$= [2, 3, 4, 2]$$

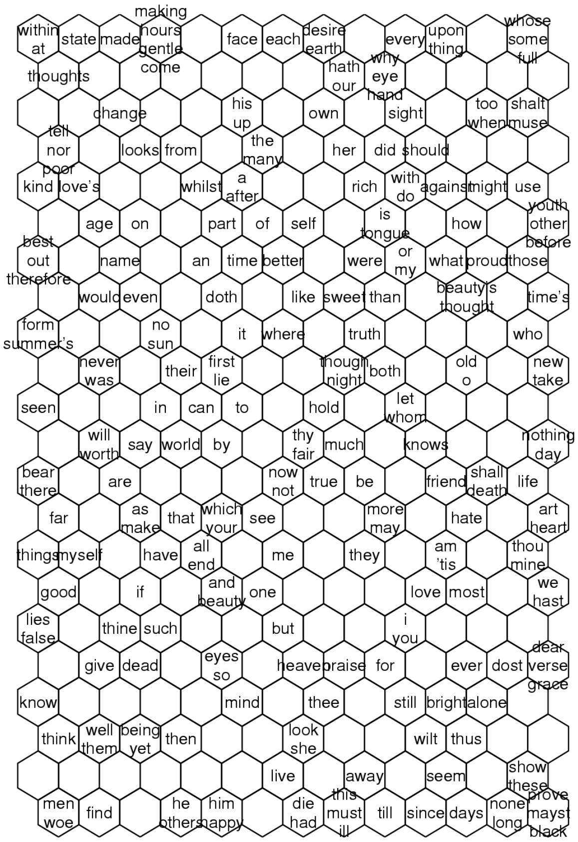
5.2.2 Resultat

Vi skapade självorganiserande kartor av orden m.h.a. den fyrdimensionella vektorrepresentationen. Orden som har liknande distribution i sonetten hamnar alltså nära varandra på kartan.

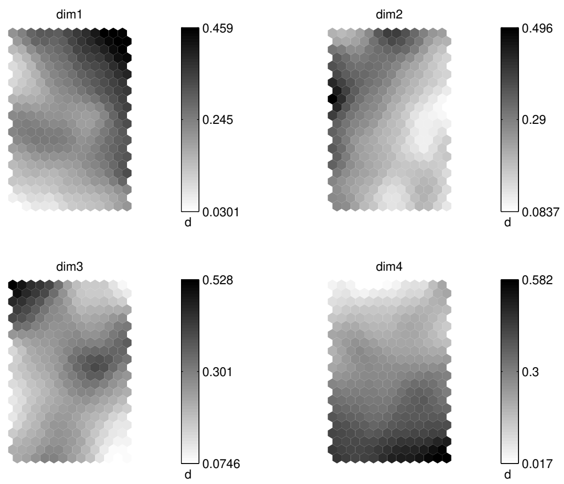
Vi visar fyra olika bilder av samma karta. De olika bilderna tar fram olika aspekter av samma karta, och bör därför iakttas parallellt. Positionen på kartan är det som kopplar ihop bilderna. På bilden 18 ser vi hur specifika ord placerats på kartan.

Eftersom inputdatan inte är kontextinformation som i de andra experiment vi diskuterar, är ordningen som framträder på kartan inte enligt syntaktiska eller semantiska kategorier, utan återspeglar hurudan funktion ordet har i sonetten.

Figur 19 visar hur värdena på de fyra dimensionerna varierar på kartan. Mörkare färg betyder att värdet för dimensionen är stor. Man kan se att de olika dimensionerna får stora värden på olika områden på kartan, vilket betyder att orden på kartan förekommer ofta på olika positioner i sonetten. T.ex. har den första dimensionen, motsvarande de fyra första raderna i sonetten, ett stort värde i hörnet uppe till höger, och ett litet värde i hörnet nere till vänster. Dimension tre är nästan



Figur 18: De 228 vanligaste orden i korpusen på en självorganiserande karta. Ordningen baserar sig på den fyrdimensionella fördelningen av orden i sonetten olika delar. Ord som förekommer i liknande fördelningar är nära varandra på kartan



Figur 19: Värdena för dimensionerna på kartan. Mörkare färg motsvarar större värde. Notera hur den fjärde dimensionen får stora värden mot kartans nedre kant, medan de andra dimensionerna får mindre värden på det området.

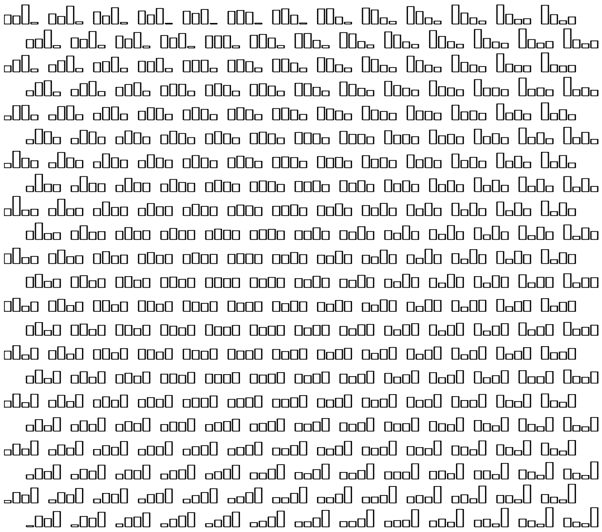
en spegelbild av detta. Figur 18 visar att orden "prove" och "black" förekommer i det nedre högra hörnet på kartan. Vi ser att dimension fyra har ett stort värde på det stället. Det betyder att dessa ord är vanliga på sonettens två sista rader, efter voltat. De specifika fördelningarna för dessa ord är $v_{prove} = [3, 2, 2, 10]$ och $v_{black} = [4, 1, 3, 10]$ (notera att den sista dimensionen är antalet gånger ordet förekommer gånger två, medan de andra dimensionerna direkt motsvarar antalet gånger ordet förekommer).

På motsvarande sätt finns det ord som "whose" och "full" som oftast förekommer på de första fyra raderna i sonetten. Deras distributioner är: $v_{whose} = [8, 6, 2, 6]$ och $v_{full} = [6, 4, 1, 4]$. Eftersom de har stora värden på den första dimensionen finns de i det övre högra hörnet på kartan (figur 18). Man kan se att trots att det finns en tendens mot den första dimensionen så förekommer orden också mycket på andra ställen i sonetten. För att få en visualisering av hur fördelningarna varierar på kartan, se figur 20, som visar ett pelardiagram av fördelningen i varje nod på kartan.

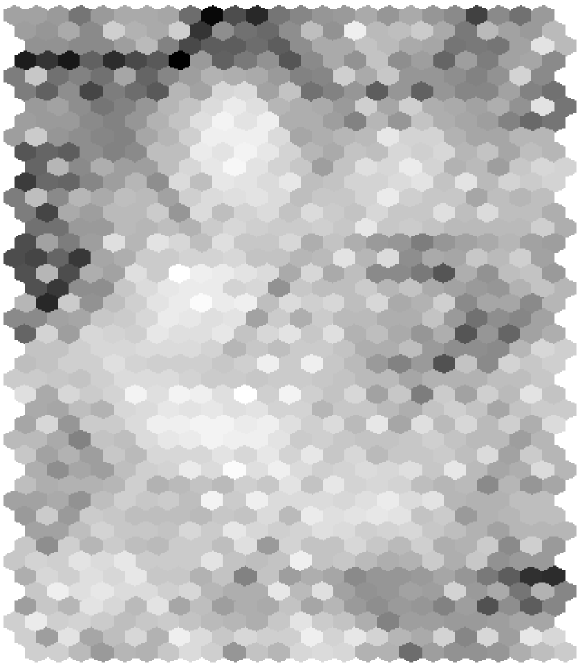
Att de olika dimensionerna får höga värden på olika delar av kartan betyder att olika ord är vanliga för olika delar av sonetten. Det i sin tur betyder att det kan vara möjligt att hitta vändpunkter med hjälp av en analys av vilka ord som förekommer på raden. Särskilt intressant är att den fjärde dimensionen, d.v.s. den efter voltat är betydligt annorlunda de andra. De andra dimensioner tenderar mot övre ändan av kartan, medan den fjärde dimensionen dominerar den nedre ändan, skild från de andra.

Om två celler på kartan är bredvid varann, betyder det att den data de representerar liknar varann. Men eftersom kartan transformerar fyrdimensionell data till ett tvådimensionellt plan på ett icke-linjärt sätt varierar avståndet mellan bredvidliggande celler i den fyrdimensionella rymden. För att visualisera det här används olika mörka färger i figur 21. Denna bild visar hur det det övre vänstra hörnet, samt en del av den vänstra sidan och det högra nedre hörnet är längre från de övriga delarna av kartan. Detta kan ses från de mörkare områden som omger dessa delar. I figur 21 ser vi också att trots att den fjärde dimensionen besitter nedre delen av kartan, finns ingen skarp gräns mellan den nedre delen av kartan och övriga kartan. Det finns ingen "bergskedja" som skulle göra en sådan indelning. Detta är en liten besvikelse för vårt sökande efter vändpunkten. Texten före voltat är olik texten efter, men gränsen är inte skarp utan glidande.

Analysen med den självorganiserande kartan visar att det skulle vara möjligt att hitta vändpunkten



Figur 20: Fördelningarna för de olika elementen på kartan, visade som stolpdiagram. Fördelningarna i en viss cell motsvarar fördelningarna för orden i samma nod i figur 18. I en cell motsvarar den första pelaren, förekomsten på de första fyra raderna, den andra pelaren förekomsten på de följande fyra raderna o.s.v. Man kan se att mitten av kartan tenderar ha jämna fördelningar, medan kartans hörn visar fördelningar som är förskjutna mot någon viss dimension.



Figur 21: U-matrisen visar avstånder mellan två celler på kartan. Mörkt motsvarar större avstånd. Ju ljusare cellerna är, desto närmare sina grannar är de i den ursprungliga rymden. Metaforiskt kan man tänka sig att det ljusa är dalar och det mörka är bergskedjor som skiljer dalarna från varann på kartan.

i Shakespeares sonetter, åtminstone med någon noggrannhet, enbart med ordfrekvensser från omgivande rader i sonetten. Orden som förekommer i det nedre högra hörnet förutspår att vi passerat voltat. Orden i det övre vänstra hörnet förutspår att vi är på de fyra raderna före voltat. Orden i mitten av kartan har jämna fördelningar och förutspår därför ingen speciell position i sonetten. Analysen visar också att sökandet efter voltat i Shakespeares sonetter inte skulle vara särskilt lätt, p.g.a. avsaknaden av skarpa gränser i ordens fördelningar, och för detta skulle det löna sig att söka fram mer prediktiva egenskaper i sonetten.

5.2.3 Slutsatser

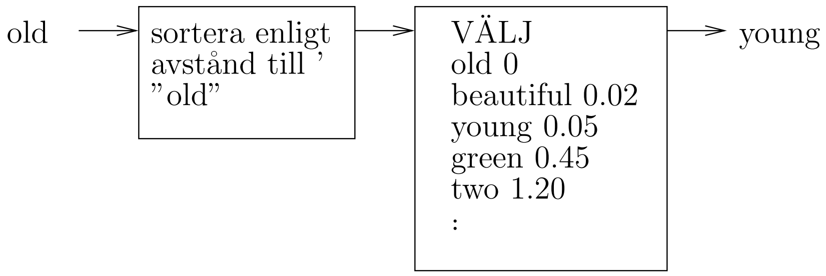
Det skulle vara möjligt att utvidga analysen av vändpunkter i litterära verk även till prosatexter. Fördelarna med att använda självorganiserande kartor är att man kan utföra analysen på ett explorativt sätt, eftersom kartan ger en kompakt visualisering av de olika fördelningarna. För prosatexter kunde man använda sidor, eller paragrafer som dimensioner och på det sättet uppnå den krävda vektorstrukturen. Alternativt kunde man använda korta kontexter (t.ex. en sida) och orden som förekommer i den kontexten som dimensioner.

5.3 Att generera språk från ICA representationer

I kapitel 4.3 berättade vi om hur man kunde använda ICA för att hitta kategorier för ord i ett korpus. Tanken bakom analysen är alltså att ord som passar i samma kontexter också i genomsnitt har liknande kontexter. Därför är de oberoende komponenterna klasser av ord som passar på samma ställe i meningen. Ord som liknar varandra i ICA-representationen borde gå att byta ut mot ord med liknande ICA-representationer och resultatet borde fortsättningsvis vara en syntaktiskt korrekt mening.

Med hjälp av denna idé gjorde vi en nonsens-generator, en generator som skapar nya möjliga meningar på basen av en inputmening. Som inlärningsdata använde vi ett korpus av texter samlade från Project Gutenberg (www.gutenberg.org). Korpusen innehöll olika prosatexter som getts ut före år 1923. Vi estimerade 50 ICA-komponenter för de 1000 vanligaste orden i korpusen. Som representation för orden användes vänsterkontexten för ordet, if förhållande till de 1000 vanligaste orden. Sedan använde vi de rader i A-matrisen som motsvarar ordet som representetion för ordet i vår generator. För varje ord i den ursprungliga meningen byter vi ut ordet mot ett annat ord, slumpmässigt ur en exponentialfördelning, så att sannolikheten att man väljer ett ord avtar snabbt enligt det euklidiska avståndet i ICA-representationen. Med andra ord är det mest sannolikt att man väljer ordet själv (som är på avstånd 0), och sedan följer de andra orden med avtagande sannolikhet. Figur 22 visar hur processeringen framskrider för ordet "old".

Detta var enbart ett hastigt experiment för att se om ICA-representationerna alls kunde vara användbara för generering.



Figur 22: Generatorn söker ersättare åt ordet "old". Först sorteras alla ord enligt det euklidiska avståndet till "old" i ICA-representationen. Då bildas listan i den stora lådan till höger, där avståndet är skrivet efter ordet. Sedan väljs ett av orden ur listan enligt en exponentiell fördelning, så att sannolikheten att något av de närmaste orden väljs är mycket stor. I detta fall väljs ordet "young".

Exempel på genererade meningar.

Originalmening: *old men perhaps never die*
Genererade menigar:
old people perhaps always die
beautiful men perhaps never die
old men perhaps never die (original igen)
old people perhaps never die
old men perhaps never panel
old people perhaps never die
young men perhaps hardly die

De genererade meningarna är grammatikaliska, och de ersättande orden passar in i meningarna.

Dessa exempel verkar alla vara rätt lyckade, men hur går det om vi tar en längre mening?

Originalmening: *alice was beginning to get very tired of sitting by her sister on the bank and of having nothing to do . once or twice she had peeped into the book her sister was reading . but it had no pictures or conversations in it . and what is the use of a book . thought alice . without pictures or conversations*

Genererad mening:*virgil was unwelcome to get too candid of sitting on her aunt on the bank and of having funny to do but finally or twice she had peeped into the book her chap was reading . but it has no vessels . conversations in it . and what has the circumference of a trick . thought daniel and without vines or relations .*

Man kan säga att början av meningen ännu är rätt lyckad, men att det finns en hel del ogrammatikaliskt i den genererade texten. ICA-representationerna verkar ändå rätt lovande som utgångspunkt för denna sorts uppgift. De stora problemen är hur man integrerar en generator som producerar grammatikaliskt nonsens med ett system som vill säga något. Man kanske kunde använda samma approach som Langkilde och Knight i kapitel 3.2, att använda den statistiska informationen för att räkna ut den mest sannolika av olika tänkbara verbalisationer.

Hittills har vi utgått från att ICA hittar "ordklasser" i textkorpus, mest enligt intuitionen att de olika signalerna motsvarar klasser. Nu kommer vi att vända oss till att försöka besvara frågan, om vad det är som ICA egentligen hittar i en kontextsignal. Om vi själva genererar olika symboler från en känd grammatik, vad hittar ICA för signaler då? Motsvarar de samma klasser som i grammatiken 1:1 eller är det mera bara något åt det hållet?

5.4 WordICA för en känd grammatik

Honkela, Hyvärinen och Väyrynen fick intressanta och intuitivt tilltalande komponenter. Men hur kommer det sig att en komponent motsvarar en ordklass? Det är inte alls så självklart. Tänk om modellen får fram andra intressanta egenskaper än bara grupperingar. Vad motsvarar ICA-signalerna egentligen?

Dessa frågor kunde vara lättare att besvara om vi undersökte hurudana resultat en motsvarande analys skulle ge för ett korpus genererat ur mycket enkla grammatiker. Därför genererar vi texter med hjälp av generativa grammatiker och sedan analyserar vi texterna dem med WordICA analysen.

Men var det inte så att vi tog avstånd från den generativa grammatiken som modell för språket? Jo, det gör vi, som fullkomlig modell för språket. Däremot är den generativa grammatiken ett utmärkt verktyg för att generera symbolsträngar med struktur. Därför använder vi generativ grammatik för att generera ett korpus som vi sedan analyserar med andra metoder. När vi känner till processen som genererar korpusen kan vi se hur olika egenskaper i grammatiken som leder till olika resultat i ICA-analysen. Vi kan också se om det finns en tillräckligt tydlig koppling mellan grammatik och ICA-komponenter för att eventuellt kunna härleda approximationer av en grammatik också från ICA-komponenter som estimerats ur ett korpus av naturligt språk.

5.4.1 Data och Metoder

Vi genererade tre olika korpus med olika komplicerade grammatiker. Från en trivial till en relativt komplicerad. Grammatiken är en probabilistisk kontextfri grammatik (eng. PCFG). Den fungerar precis som de generativa grammatikerna i kapitel 1.1.2, förutom att varje alternativ produktionsregel har en medföljande sannolikhet. Den sannolikheten anger hur ofta regeln väljs. När vi genererar meningar startar vi alltså från startsymbolen S och väljer slumpmässigt av de möjliga produktionsreglerna, enligt sannolikheterna som specificerats i grammatiken. Sannolikheterna är fördelade så att om grammatiken innehåller en regel som genererar nya nonterminaler, så väljs regeln som antas ur en jämn fördelning, d.v.s. om det finns tre alternativ så väljs ett av dem med en tredjedels sannolikhet. De regler som genererar var sin ordklass, alltså enbart terminalsymboler behandlar vi lite annorlunda. Eftersom de vanligaste orden i ett korpus är mycket vanliga och de mindre vanliga orden är ungefär lika (o)vanliga approximerar vi en sådan fördelning genom att generera orden i varje ordklass ur en exponentiell distribution (se [Manning och Schütze, 1999] s. 23-29 om ords fördelningar).

Alla ord i grammatikerna tillhör entydigt exakt en ordklass. Det betyder att "noun001" alltid genereras av nonterminalsymbolen "Noun" och alltså inte tillhör flera klasser. I verkliga naturliga språk finns det ord som enligt kontext tillhör olika ordklasser, t.ex. engelskans "can" som kan vara verbet kunna, verbet konservera eller substantivet konservburk. ICA-modellen kan teoretiskt beskriva ett sådant ord, eftersom de olika klasserna antas vara oberoende komponenter. Ordet skulle då vara en blandning av två komponenter, den som motsvarar verb och den som motsvarar substantiv. Det skulle vara möjligt och intressant att i framtiden utvidga vår analys till sådana ord.

Grammatikerna är inspirerade av två källor [Dougherty, 1994, Fry, 2004] som beskriver grammatiker av förenklad engelska. De grammatiker som vi använder finns inte exakt i dessa källor, men många konstruktioner i grammatikerna är hämtade ur dem. Sannolikheterna i grammatikerna finns inte utmärkta, eftersom de är desamma i alla grammatiker. Om vänster sida är en nonterminal som har alternativa produktioner väljs någon möjlig produktion ur en jämn fördelning. Om nonterminalen däremot producerar bara terminalsymboler (vilka i grammatikerna är ersatta med symboler som börjar med versal, men i övrigt har små bokstäver), så väljs en specifik terminalsymbol ur en exponentiell fördelning.

Eftersom vi minskar datans dimension med PCA innan vi estimerar ICA kan det vara en bra jämförelse att jämföra ICA resultaten med PCA resultaten (se slutet av kapitel 1.2.4). Eftersom denna analys till sin natur är visuell och inte kvantitativ, är det inte trivialt att jämföra dessa två metoder. Men eftersom vi vet att ICA görs efter PCA som en efterprocessering, så borde ICA-resultaten bli bättre än PCA-resultaten för att vara motiverade.

Grammatik 1 Grammatik 1 är mycket enkel, bara en produktionsregel:

S → Noun Verb Noun

Noun och Verb är nonterminaler som motsvarar substantiv och verb, båda genererar 100 olika substantiv, respektive verb. Ordens form är "noun001", "noun002"... "noun100".

Grammatik 2 Grammatik2 är också rätt enkel, men innehåller fler effekter, såsom ordklasser vars ord är betydligt vanligare än de andra klassernas.

S → NP VP .
NP → Det Noun
VP → Verb NP

Det motsvarar en bestämd, eller obestämd artikel. Artiklar förekommer två gånger i varje mening, och eftersom det vara finns två av dem så är de mycket vanliga. Dessutom har vi med punkten, som förekommer i varje mening och därför är mycket vanlig. Detta för att kolla hur dessa variationer i frekvens mellan klasser påverkar ICA resultaten.

Grammatik 3 Grammatik 3 är den mest komplicerade, och är en kraftigt förenklad version av engelskans grammatik. Den innehåller några flera ordklasser än de förra: *Adj*, *Prep* och *Pron*. Den första klassen innehåller 100 ord och de två senare 10 var. Detta för att simulera att prepositioner och pronomen är slutna klasser med färre ord. I vårt experiment är alla klasser i praktiken slutna, så egentligen rör det sig bara om en skillnad i antal, vilket också kan påverka ICA resultaten. Grammatik 3 är också rekursiv, d.v.s. en mening kan innehålla en bisats som i sig själv är en mening. Detta leder till att meningarna teoretiskt sätt kan vara hur långa som helst, men i praktiken är långa meningar mycket osannolika, eftersom man måste välja en och samma produktion om och om igen, vilket har låg sannolikhet.

S→ S' .
S'→ NP VP

NP → Det Noun
NP → Det Adj Noun
NP → Pron

VP → Verb NP
VP → Verb NP PP

VP → Verb PP PP
VP → Verb NP S'

PP → Prep NP

Korpusstorlek och preprocessering För att undvika effekter orsakade av för lite data valde vi att generera korpus ur de tre grammatikerna så att varje korpus består av en miljon meningar ur grammatiken. Detta leder till att de ovanligaste orden förekommer över 300 gånger i korpusen, vilket borde vara tillräckligt.

För att kunna tillämpa ICA behöver orden kunna sparas i vektorform. Vi använder den ofta använda metoden att låta varje ord representeras av sin kontext. Varje ord sparas som en vektor där komponenterna är antalet gånger ett visst ord förekommer till vänster om ordet. Vi gör ingen skillnad på meningsgränser, enligt vanlig praxis. Om ett ord alltså är först i en mening så är dess vänstra kontext det ord som är vänster om det, d.v.s. ordet som är sist i föregående mening. Det som på detta sätt kommer över meningsgränsen anses vara slumpmässigt brus, som i medeltal inte påverkar. I detta fall är det inte riktigt sant att säga att ordet i slutet av meningen är slumpmässigt, eftersom meningarnas form är så likadan. I praktiken leder denna regelbundenhet till att det som går över meningarnas gränser är nyttig information för ordklassificering.

Som exempel av preprocesseringen, tar vi två meningar ur korpusen genererat av grammatik 1:

noun006 verb018 noun017
noun015 verb035 noun005

Sedan räknar vi ut vilka ord som ligger bredvid varann:

(noun006, verb018) (verb018, noun017), (noun017, noun015), (noun015, verb035), (verb035, noun005)

Dessa omvandlas till en matris där raderna motsvarar ett ord t.ex. “noun006” och komponenterna i radvektorn innehåller antalet gånger som de andra orden förekommer vänster om “noun006”.

Matrisens preprocesseras ytterligare genom att ta logaritmen av alla värden enskilt. Kontextmatrisen är cX så är X:

$$X = log(cX + 1) \tag{17}$$

Vi adderar 1 för att logaritmen inte är definierad då dess argument är noll och $log(1) = 0$.

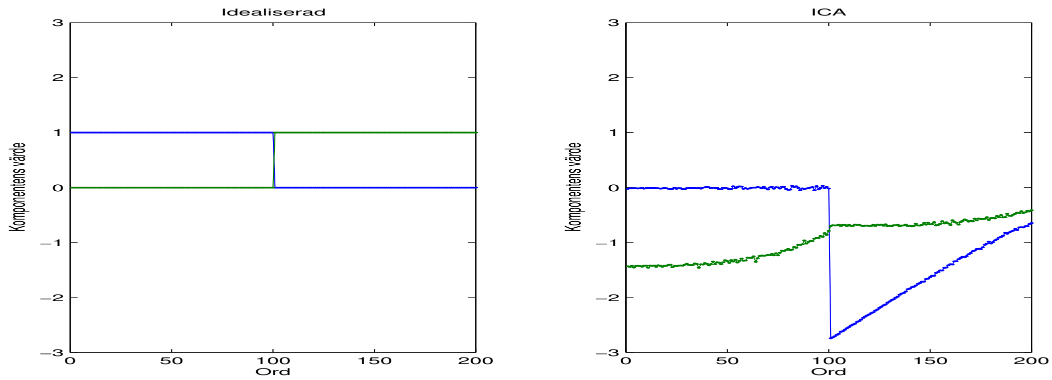
5.4.2 Estimering av ICA

ICA-Komponenterna estimerades med Matlabs FastICA-paket, med kommandot:
$[icasig, A, W] = fastica(X,'lasteig', nr\_of\_components,'approach','symm','stabilization','on');$

Kommandot innebär att fastica först minskar datans dimension till $nr\_of\_components$ med hjälp av PCA, och sedan tillämpas ICA i den sänkta dimensionen.

5.5 Resultat

Inspirerade av resultaten som Honkela, Hyvärinen och Väyrynen fått hoppas vi att man skulle kunna hitta klasserna i ICA-komponenterna, helst så att en ICA-signal motsvarar kontexten hos



Figur 23: Till vänster: Den idealiserade A-matrisen. Till höger: A-matrisen som estimerar ICA. På x-axeln är de 200 olika orden. De 100 första tillhör klassen $Noun$ och de 100 senare klassen $Verb$. Enligt den idealiserade uppfattningen borde komponenterna ha sådana värden att A-matrisen skulle vara som i figuren till vänster: noll när ordet inte tillhör kategorin, och större än noll när ordet tillhör den. I den högra bilden ser vi att ICA inte ger exakt detta, men ändå något ditåt. Den blåa grafen i högra komponenten liknar den gröna komponenten i den vänstra, men har omvänt förtecken. Eftersom förtecknen i ICA är godtyckliga, är detta inget problem för modellen. Däremot tycks den gröna signalen i den högra bilden inte direkt motsvara den blå signalen i den vänstra bilden. Det betyder att den ena klassen har en egen ICA-signal, men den andra klassens kontextdata kan uttryckas som en summa av de båda komponenterna.

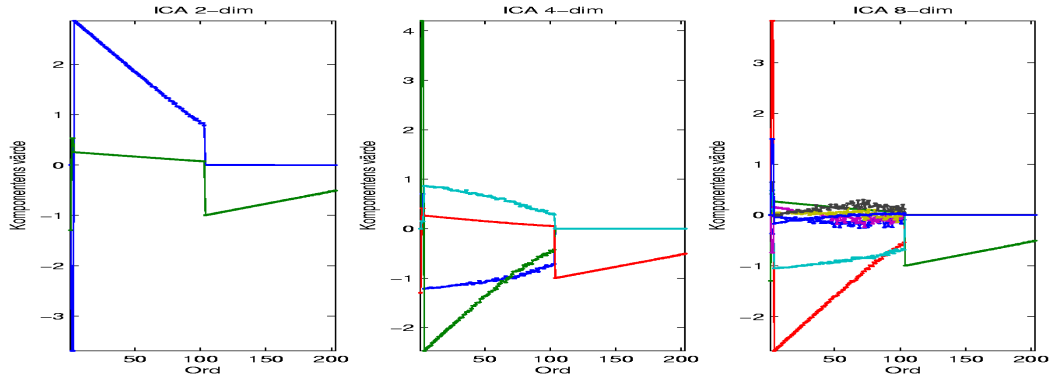
en klass. Varför det? Tänk att du har ett ord "noun021" som tillhör klassen "Noun". Om det skulle finnas en ICA-signal s_t som motsvarar kontexten för klassen "Noun" så skulle "noun021" ha en rad i A-matrisen där koefficienterna skulle vara nära noll, förutom för signalen s_t, där koefficienten skulle ha ett större absolut värde. Detta kommer vi att kalla för att signalen s_t *beskriver* klassen $Noun$. Vi skulle gärna anta att en ICA-signal beskriver en klass då vi tillämpar ICA på ett korpus av naturligt språk.

Andra önskvärda scenarion skulle vara att ICA-signalerna skulle specialisera sig på olika delar av kontexterna. Det skulle kunna innebära att det t.ex. fanns en komponent som beskriver substantiv, och en annan som beskriver pluralformer. Då skulle ett ord som är både substantiv och pluralis ha ett stort värde för båda komponenterna. Vi har inga sådana fenomen som pluralformer i vårt experiment, men om t.ex. klassen $Noun$ skulle beskrivas av två signaler som samtidigt var starka i A-matrisen, skulle vi godkänna det som ett bra resultat.

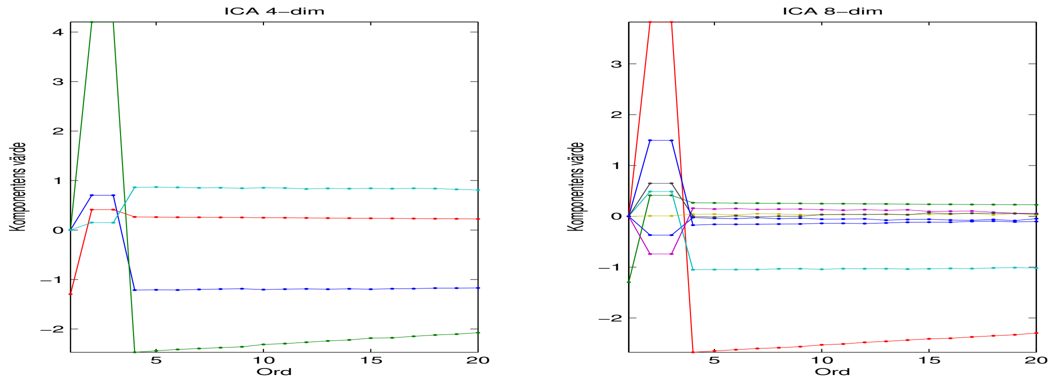
Hurudana resultat får vi då med vårt enklaste korpus ur grammatik 1? Eftersom grammatik 1 innehåller bara två klasser $Noun$ och $Verb$ är det naturligaste komponentantalet att estimera två stycken. Vi skulle hoppas få ett resultat där den ena komponenten motsvarar $Noun$, och den andra motsvarar $Verb$. I figur 23 jämförs den idealiserade A-matrisen med den som ICA ger i verkligheten. Man kan se att den ena klassen har en associerad ICA-komponent, men den andra klassen bildas som en blandning av två komponenter.

Detta resultat visar att ICA hittar klasserna ganska bra, om än inte riktigt i den form vi tänkte oss. Men hur ser det ut för de mer komplicerade grammatikerna? I figur 24 är A-matrisen för grammatik 2. Grammatik 2 innehåller intressanta fenomen, såsom punkten och artiklarna. Dessa förekommer väldigt ofta i datan, både punkten och nåndera artikeln finns i varje mening i korpusen. Detta kunde leda till att dessa vanliga datapunkter beskrivs noga av modellen, medan substantiven och verben som det finns 100 olika av skulle bli mindre väl beskrivna.

Vi kan se i figur 24 att figurerna ser betydligt rörigare ut än för grammatik 1. En orsak är att det



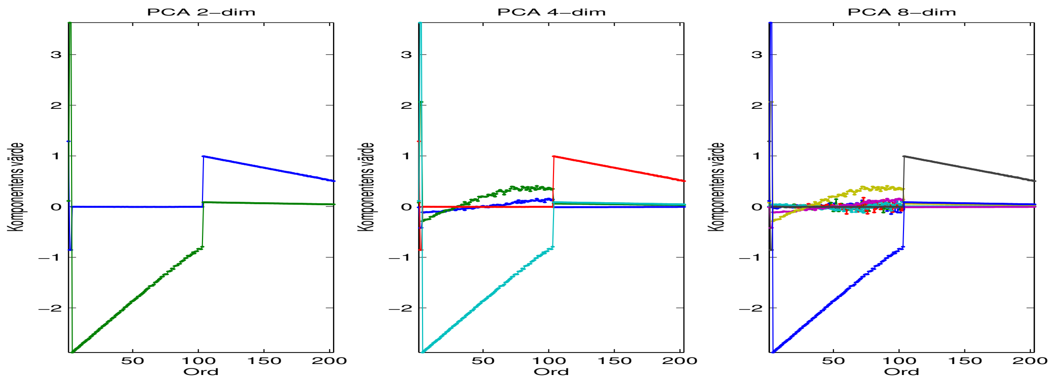
Figur 24: ICA-komponenter för grammatik 2. Till vänster: 2 komponenter estimerade, mitten: 4 komponenter och till höger 8 komponenter. På x-axeln hör index 1 till klassen *Dot* (“.”), index 2-3 tillhör klassen *Det*, index 4-103 till $Noun$ och index 104-203 till $Verb$



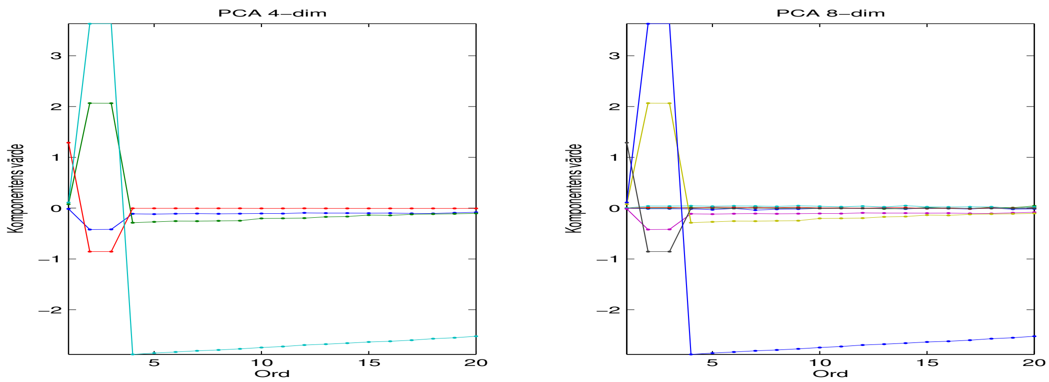
Figur 25: Förstoring av figur 24 för 4 och 8 ICA-komponenter. Observera t.ex. den blåa komponenten i vänstra bilden (och mittersta bilden i figur 24). Den är > 0 för index 2-3, d.v.s. *Det* och < 0 för index 4-103, alltså $Noun$. Den cyanfärgade komponenten får värden > 0 för $Noun$ och är nära 0 i övrigt. Den röda komponenten får negativa värden för index 0 och index 104-203, d.v.s *Dot* och $Verb$. Liknande fenomen kan observeras i de högra bilderna.

finns fler klasser i grammatiken. En annan orsak verkar vara att när man estimerar fler komponenter än det finns klasser så börjar de innehålla mer brus, vilket syns i det 8 dimensionella fallet i början på x-axeln. En ytlig blick visar att många av komponenterna har värden som är skilda från noll på samma ställen. Den idealiska klassindelningen tycks alltså inte riktigt uppfyllas. Eftersom de mest intressanta sakerna händer i början av x-axeln finns de 20 första ordens komponenter visualiserade i figur 25

Ur figurerna framgår att komponenterna inte motsvarar den idealiska modell vi gjorde upp tidigare. Trots detta följer de kraftigt den klasstruktur som finns i datan. Komponenterna tycks vara mer kontrastiva än beskrivande för en klass. T.ex. den gröna komponenten i figur 25, till vänster, beskriver något som är en *Det* men inte en $Noun$ (eller vice versa). Istället för att beskriva en klass ensam, beskriver den alltså en kombinationseffekt av två klasser. Denna egenskap kan innebära att ICA-komponenterna är mycket användbara vid klassifikation, men att deras tolkning kanske inte borde göras simplistiskt, en komponent i taget, utan man borde ta alla komponenters värden i betraktande.



Figur 26: Hur de olika PCA-komponenternas värden varierar för olika klasser. Motsvarar figur 24, grammatik 2. På x-axeln hör index 1 till klassen *Dot* (“.”), index 2-3 tillhör klassen *Det*, index 4-103 till *Noun* och index 104-203 till *Verb*. Den vänstra bilden är praktiskt taget identisk med den i figur 24. I den mittersta bilden ser man aningen mer brus än i det motsvarande fallet för ICA, och samma kan sägas om den högra bilden.

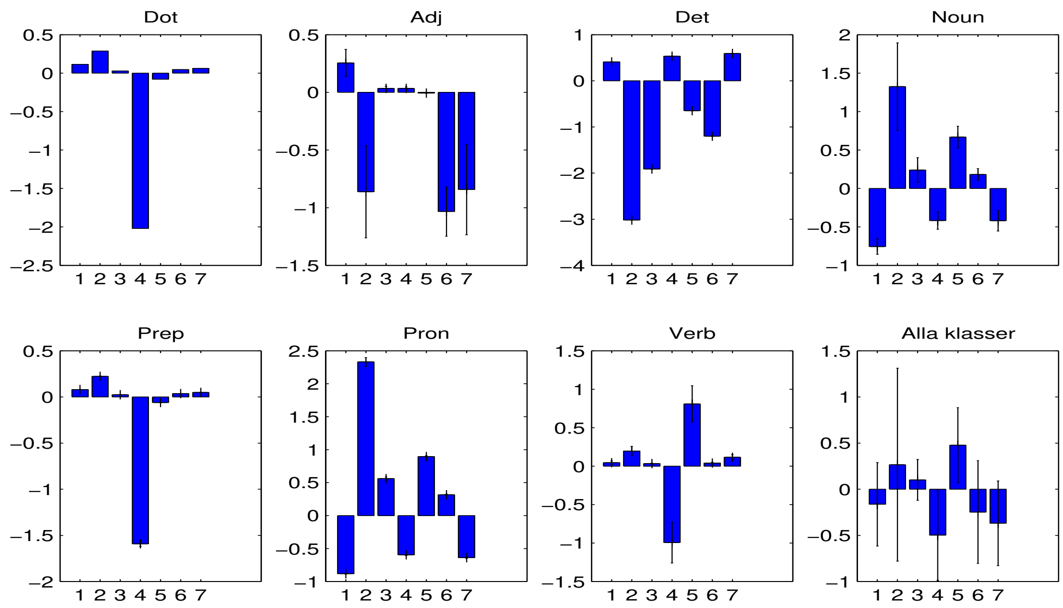


Figur 27: Förstoring av figur 26 för 4 och 8 ICA-komponenter. Som i figur 25 kan man se komponenter som följer en viss klass. T.ex. den röda komponenten i den vänstra bilden är > 0 för index 1, alltså *Dot*, < 0 för index 2-3, *Det* och sedan 0 tills 104-203 alltså *Verb* (se mittersta bilden i figur 26).

Jämförelse med PCA. Tyvärr verkar det inte vara så tydligt att ICA skulle vara bättre för denna specifika analys. Vi kommer att spekulera i orsakerna till detta senare, men först några illustrationer som visar att så verkar vara fallet. Vi såg bl.a. i figurerna 24 och 25 hur komponenterna inte direkt motsvarade klasser i datan, utan var mer av naturen: “Hör till klass A, hör inte till klass C”. I figurerna 26 och 27 finns motsvarande grafer för PCA-komponenterna.

Man kan se stora likheter med de tidigare ICA-graferna. Särskilt de grafer i låga dimensioner ser så gott som identiska ut. Antagligen beror det på att dimensionen reducerats så kraftigt att viktig information kommit bort, och då saknar ICA information att jobba med. I de fallen där vi estimerat fler komponenter kan man tyda skillnader, om än inte så väldigt tydliga. I vissa fall verkar ICA-komponenterna innehålla mindre brus och följa klasstrukturen mera, men det är inga stora skillnader det är fråga om.

I grammatik 3 har vi betydligt mer av det naturliga språkets intressanta egenskaper, mer variation, mer olika klasser. Det är inte förnuftigt att använda samma visualisering, eftersom de många



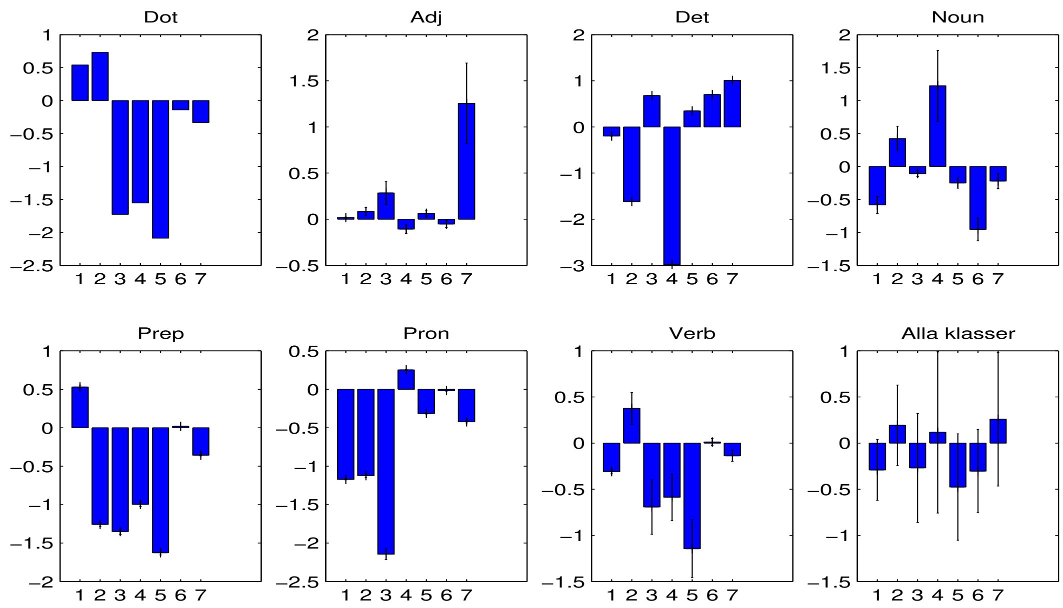
Figur 28: De olika klassernas ICA komponentdistributioner. Den svarta linjen ovanpå staplarna visar standardavvikelsen. Nere, längst till höger är motsvarande graf över alla klasser. Som kan ses på standardavvikelsen är distributionerna ganska lika mellan de olika orden inom klasserna, men olika mellan klasserna.

klasserna och komponenterna bara gör grafen otydlig. Därför tar vi istället medeltalet av alla komponenter i en klass och standardavvikkelsen. Då får vi grafen i figur 28. Vi ser att de komponenter som är nära noll är mycket stabila inom klassen, medan de som har större värden varierar mer. Detta är ett tecken på att de komponenter som varierar mer innehåller information som är olika inom klassen, p.g.a. varierande frekvenser och slumpmässig variation.

Det mest intressanta är att se att distributionerna hos ICA-komponenterna är väldigt lika inom klassen. Detta betyder att de kunde användas till att klassificera ord. En annan mycket intressant slutsats är att trots att komponenterna säger mycket om klasstillhörighet, så är inte tillhörigheten av det slaget som vi antog, att varje klass har en eller några komponenter med stora absoluta värden som bäst indikerar klasstillhörighet. De små värden hos komponenter tycks vara mer stabila indikatorer. Visst kan man säga att de komponenter med stora absoluta värden beskriver information för den klassen, men det kan man också säga om komponenter med små absoluta värden för den klassen. Hela distributionens form är alltså viktig att betrakta.

Vi kan också se att de klasser som innehåller fler ord, *Adj*, $Noun$, $Verb$ har större variation inom klassen än de som innehåller få ord. Detta är egentligen att vänta, eftersom många ord leder till en mer komplicerad kontextsignal, med större variation mellan olika ord.

En detalj som kan observeras i figur 28 är att vissa klasser har nästan identiska distributioner, *Dot* och $Prep$ samt $Noun$ och $Pron$. Detta beror på att dessa klassers kontexter är mycket lika varandra, då man använder enbart vänsterkontext. Om vi utvidgar analysen till att använda både vänster och högerkontext ur grammatik 3 får vi grafen i figur 29. Vi ser att trots att *Dot* och $Prep$ har väldigt liknande komponentfördelningar så är de inte identiska. Det finns vissa komponenter som är klart olika. Å andra sidan har $Noun$ och $Pron$ inte längre stora likheter. Detta beror på att trots att vänsterkontexterna är så gott som lika mellan dessa två klasspar, så påverkar högerkontexten hur lika slutresultatet blir. I fall av *Dot* och $Prep$ är också högerkontexten liknande, däremot för $Noun$ och $Pron$ är högerkontexterna alldeles olika (se figur 30). Det är i själva verket överraskande



Figur 29: Samma tanke som i figur 28, men ICA tillämpas på en signal som innehåller både vänster- och högerkontext för varje ord.

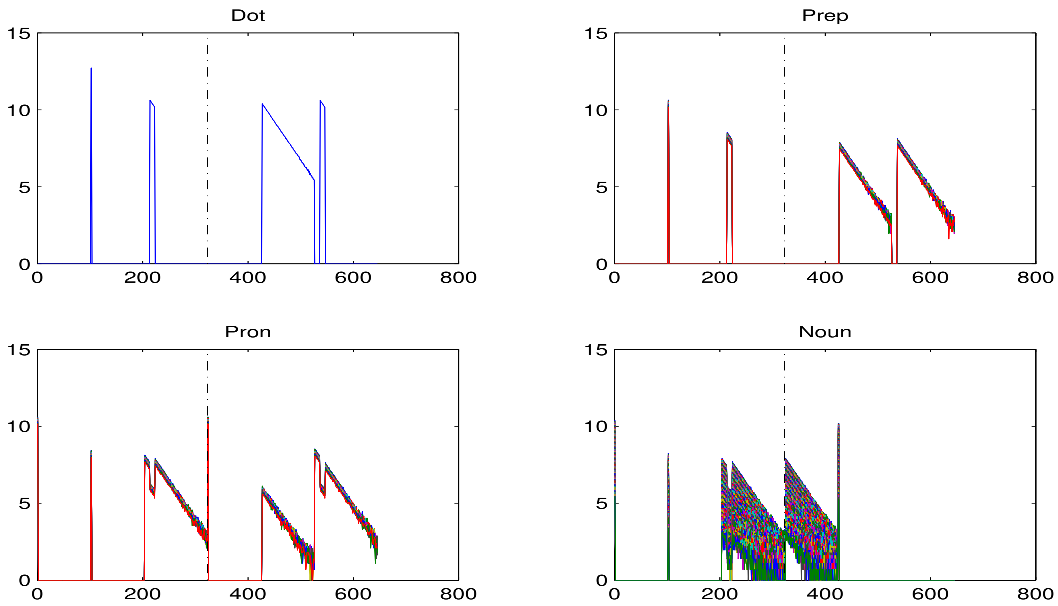
hur olika distributionerna för två klasser vars halva kontextvektor är nästan identiska blir.

Vi gjorde också ett försök att tillämpa det separationsmått som Väyrynen och Honkela utvecklat för att jämföra ICA och SVD [Väyrynen och Honkela, 2005]. På grund av den kontrastiva naturen hos komponenterna vi fick, visade sig detta mått vara missvisande. Följaktligen måste vi lämna den approachen. Antagligen fungerar måttet bättre för naturligt språk.

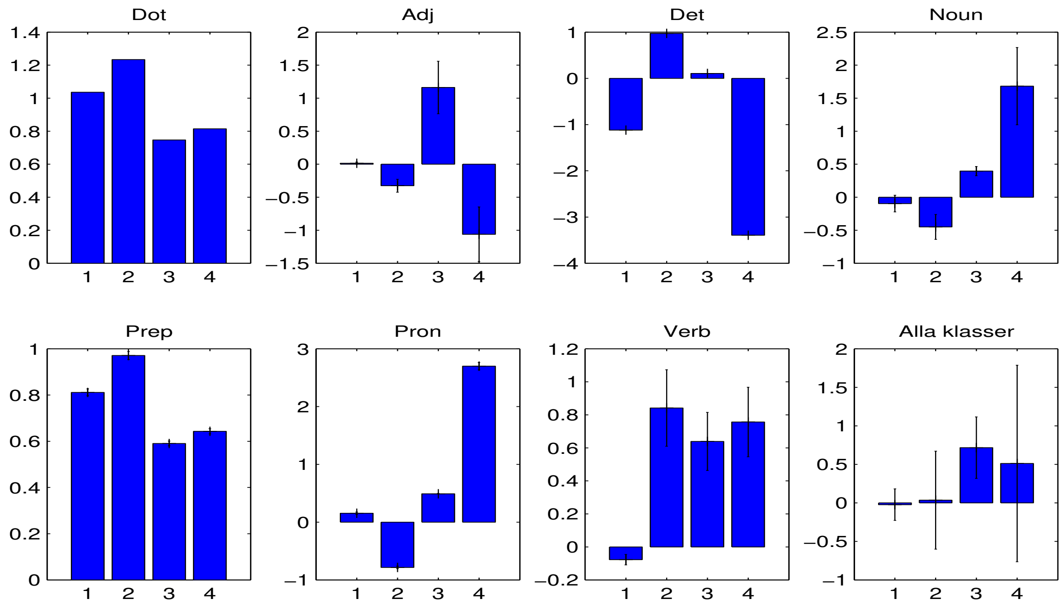
Jämförelse med PCA. Hur förhåller sig dessa resultat till PCA?. Hur varierar PCA-komponenterna inom och mellan klasserna? Vi gjorde motsvarande grafer som i figur 28 och 29. Det är alltså komponenterna som estimerats för grammatik 3. I figur 31 ser PCA-komponenterna för klasserna, estimerade enbart ur vänsterkontext (motsvarande ICA-komponenterna i figur 28). I 32 är PCA-komponenterna som estimerats med både höger och vänsterkontext (motsvarande ICA-komponenterna i 29). Enligt dessa grafer verkar båda metoderna ge sådana komponenter som skulle vara nyttiga för klassificering. Distributionerna är stabila inom gruppen, och varierande mellan grupperna.

5.6 Diskussion

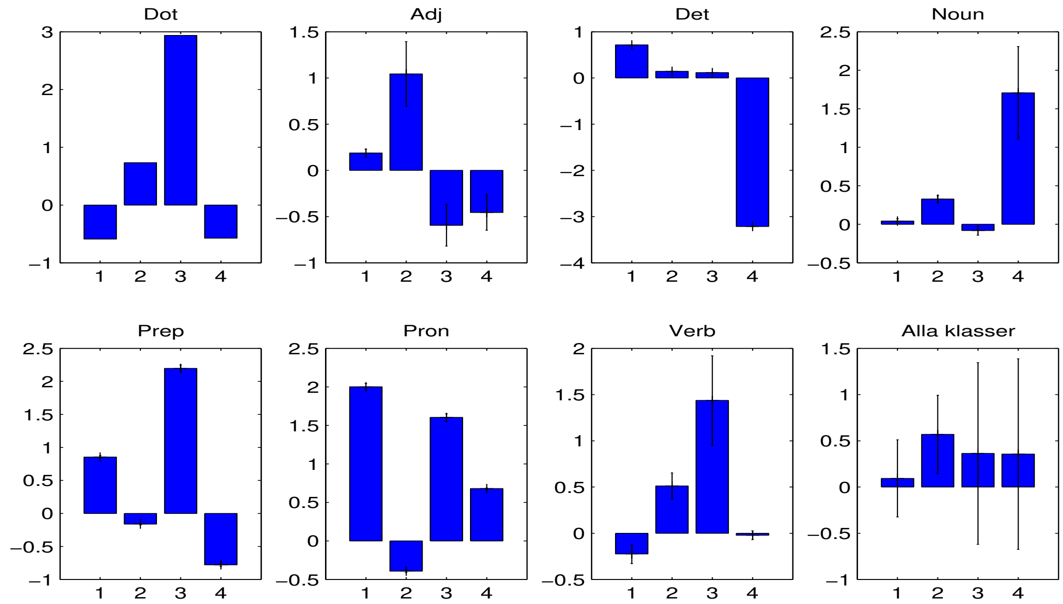
Vi observerade att ICA ger intressanta signaler som innehåller gott om klassinformation. Signalerna verkar inte motsvara enskilda klasser utan verkar ha en tendens att vara sådana att flera komponenter måste kombineras för att beskriva en klass. Vi märkte också att PCA gav väldigt liknande resultat som ICA, trots att Väyrynen och Honkela fick betydligt bättre resultat med ICA än SVD (som är ungefär PCA) [Väyrynen och Honkela, 2005]. Vi drar därför slutsatsen att våra grammatiker inte innehåller de egenskaper hos det naturliga språket som får ICA att fungera bättre. Kanske är orsaken att vi inte har ord som tillhör flera klasser samtidigt. Detta kunde vara något värt att undersöka. Den visualiseringsmetod vi använt, att lägga ord av samma klass efter varann på x-axeln och följa komponenternas värden på y-axeln verkar användbart. Man kunde or-



Figur 30: Kontextvektorerna för ord i de olika klasserna. Det som är till vänster om den svarta streckade linjen är vänsterkontexten och till höger om den streckade linjen är högerkontexten. De färgade kurvorna är kontextvektorerna för varje ord i klassen var för sig, så att en färgad kurva motsvarar ett ords kontextvektor. I de två graferna uppe ser vi *Dot* och *Prep* vars vänsterkontext är nästan identisk. Högerkontexten är också liknande, följaktligen ser klassernas ICA-komponenter liknande ut i figur 28 och 29. Nere ser vi samma information för klasserna *Pron* och $Noun$. Vi ser att eftersom det finns fler ord av klass $Noun$ så är också variationen inom klassen större. Vi märker att trots att vänsterkontexten är nästan identisk så är högerkontexterna alldeles olika. Detta leder till att distributionerna i figur 28 är nästan identiska, men i figur 29 liknar de inte varann särskilt mycket.



Figur 31: De olika klassernas PCA komponentdistributioner motsvarande fallet i figur 28, grammatik 3 endast vänsterkontext.



Figur 32: De olika klassernas PCA komponentdistributioner motsvarande fallet i figur 29, grammatik 3 höger och vänsterkontext.

ganisera orden ur ett korpus naturligt språk enligt ordklasstillhörighet och se hur komponenterna varierar för de olika orden. Graferna skulle inte vara lika jämna och enkla som för vår genererade grammatik, men kanske ändå skulle ge en insikt i varför och hur ICA fungerar för naturligt språk.

Ett sätt att utveckla resultaten kunde vara att begränsa ICA så att den bara hittade signaler som är av en viss natur. T.ex. eftersom vår data är positiv så kunde det vara en fördel att kunna begränsa signalerna till att vara antingen positiva eller negativa, men inte både och. Då skulle man undvika effekter som gör att signalerna motsvarar "hör till klass A men inte klass B". Detta skulle leda till mer intuitiva komponenter. Det finns metoder för att begränsa ICA så att A-matrisen strävar efter färre koefficienter med stort absolut värde [Hyvärinen och Karthikesh, 2002]. Detta skulle kunna ge bättre klassinformation för språkdata.

Det är beklagligt att vi inte lyckades evaluera komponenternas separation av kategorierna från varandra med det separationsmått utvecklat av Väyrynen och Honkela. Tydligen var skillnaden mellan korpusen för stora, så måttet mätte fel saker för våra resultat. Måttet idealiserar fallet att klasserna har varsin komponent, och straffar fall då det inte är så. Som vi observerat i figurerna har inte ICA signalerna en sådan form i våra experiment. I vissa fall är t.ex. komponenterna sådana att det som skiljer klasserna åt är ett negativt / positivt värde för en viss komponent. Det fallet klarar inte separationsmåttet av, och man borde utveckla specialfall för detta. Fortsättningsvis skulle det vara bra att analysera vilken egenskap i datan som gör att en komponent motsvarar en klass, och varför detta gäller för naturligt språk.

6 Diskussion

Våra experiment kunde utvidgas på många sätt i framtiden. Vår litteraturöverblick av stilanalys i kapitel 2.2 visar att valet av egenskaper för analysen i regel görs på ett rätt så ad-hoc sätt. Koppel m.fl. löste detta genom att välja egenskaper som klassificerar datan som önskat [Koppel m.fl., 2003b]. Vårt experiment visade att det var kritiskt vilken representation som väljs när analysen utförs. Därför kunde det vara en bra idé att tillämpa mer allmäna metoder för att lära sig vilka egenskaper i texten som är betydelsefulla för stilklassificering (eng. learning metrics, se [Kaski och Sinkkonen, 2004]).

Analysen av Shakespeares sonetter visar att förändringar i en text kan hittas automatiskt. En naturlig utveckling av detta skulle vara att tillämpa liknande metodologi på prosatexter.

Att generera språk i olika stilar är ett intressant problem, men också ett mycket svårt sådant. Trots att vi kan skilja olika stilar, vet vi inte riktigt vad som gör stilarna olika. Statistisk analys är värdefull eftersom man kanske kan göra analysen av existerande texter så noggrann att man börjar se vad de underliggande mekanismerna bakom olika stilar kan vara. Hypotetiskt kunde också statistisk kunskap ge möjligheten att generera text som stilmässigt liknar någon annan text. Problemen som måste lösas innan man kan göra något sådant är många. Det kanske mest avgörande är problemet hur man ska representera innehållet i en text. För att kunna generera samma innehåll med olika stil behöver man ett stiloberoende sätt att representera textens innehåll. Det här är ett problem som gäller all språkgenerering.

ICA experimenten kunde tänkas utvidgas så att man försöker rekonstruera den ursprungliga grammatiken med hjälp av ICA-representationerna. Att kunna göra det kunde ge intressanta insikter i hurudana grammatiker som kunde användas för att beskriva stora korpus. Ännu mer intressant skulle det vara om man kunde hitta grammatiker, som inte är generativa grammatiker, utan beskriver korpusens struktur i stora drag. Statistiska metoder har traditionellt lyckats bäst i uppgifter som inte är så väldefinierade, och där "i stora drag" ger ett bra resultat, t.ex. informationssökning. För detta skulle det krävas en djupare insikt i ICA-modellen tillämpad på naturligt språk.

Sammandrag

Det finns olika teoretiska sätt att närma sig språket: T.ex. de som är baserade på symboliska grammatiker samt formell logik, och de som är baserade på kognitiv vetenskap. Att det finns flera teoretiska ramverk gör området svårt att greppa. Där de symboliska metoderna ofta har nått en mognad med välutvecklad matematisk teori, är de kognitiva teorierna ofta mindre mogna. De symboliska metoderna har tillämpats i stor utsträckning på naturligt språk, och dominerar också genereringen av naturligt språk. Många språkliga aspekter, som t.ex. metaforer, ords pragmatiska betydelser är dock enklare att beskriva inom ramen för de kognitiva teorierna, medan de symboliska teorierna har svårt att betrakta dem. Att producera ett system som genererar språk med symboliska metoder är väldigt tidskrävande och ofta faller man tillbaka på att använda enklare metoder, som t.ex. templates. Detta ser vi som ett tecken på att de symboliska metoderna är otillräckliga för många praktiska språktillämpningar. Mekanismerna bakom människans språkproduktion är ännu dåligt kända och de kan därför inte återskapas maskinellt. Som följd måste forskningen på detta område sakta maka sig framåt genom att producera språk som liknar det människor producerar, utan kunskap om vad som försiggår när människan utför denna uppgift.

Med tanke på denna svåra uppgift ser det ut att finnas gott om utrymme att utveckla adaptiva, datadrivna metoder inom automatisk generering av naturliga språk. Området är inte särskilt mycket utforskat, och de dominerande metoderna är opassande för många tillämpningar. Problemet är att de adaptiva metoderna behöver ett stödande språkvetenskapligt ramverk som till stor del saknas.

Vi har visat att det är möjligt att detektera olika stilars språk med hjälp av statistisk analys. Nästa steg på vägen mot Learning to Translate-temat skulle vara att identifiera vad det är som gör stilar olika. Detta är en mycket svår fråga, eftersom den kräver att man har insikt i hur människor producerar språk. Denna fråga kan inte förväntas bli löst inom snar framtid. Medan vi väntar på en modell för människans språkförmåga kan statistiska metoder tjäna som en approximation av något som kan konvertera texter från en stil till en annan. För att ett sådant system skall kunna realiseras skulle det kräva ett sätt att beskriva en texts innehåll, oberoende stilen, och sedan ett sätt att generera texten med en önskad stil.

Att beskriva betydelsen av en text är inte något trivialt problem. Det skulle antagligen vara enklast att närma sig problemet för något mycket avgränsat specialområde inom vilket man kan göra en detaljerad och välmotiverad modell av betydelserna, t.ex. baserat på Gärdenfors konceptuella rymder. Att sedan kombinera en sådan modell med något slags stilrepresentationer kunde föra oss en bit på vägen.

Stilrepresentationerna skulle behöva vara modeller av texten som kan användas för att konvertera en betydelserrepresentation till en riktig text. Detta är i praktiken generering av naturligt språk. Ironiskt nog visar denna studie att det naturliga språket är så komplicerat att det är lättast att närma sig det naturliga språket utifrån en konstgjord värld, och följaktligen också ett konstgjort språk.

För att kunna utveckla sådana stilrepresentationer, som är generatorer av naturligt språk från en semantisk specifikation, skulle det också kräva att vi kan beskriva språkets strukturer i grammatikalisk mening på ett effektivt sätt. Vi har kritiserat symboliska grammatiker för att vara arbetsdryga, men har själva mest ignorerat frågan om struktur. Vill man realisera Learning to Translate-tanken så kan man inte undvika frågan om struktur.

De metoder som vi använt som exempel på inlärningsmetoder som producerar emergenta representationer, SOM och ICA, har båda en väldigt begränsad syn på struktur. De behandlar vektordata. Om det finns ett välmotiverat sätt att beskriva språklig struktur i vektorform skulle det vara ett steg på vägen. Överhuvudtaget har statistiska metoder oftast en tendens att ignorera strukturella frågeställningar, t.ex. genom att anta en viss struktur.

Vi visade att man kan tillämpa SOM för att analysera semantiska egenskaper i Shakespeares sonetter. Detta var en mycket ytlig analys, men den påvisar att semantiska egenskaper hos en text inte är helt utom räckhåll för statistisk processering. I rättvisans namn bör ändå påpekas att den semantiska insikt som nåddes genom analysen var rätt ytlig, och att den inte alls minskar behovet av nya semantiska modeller.

Vi visade också att ICA-representationerna gick att använda för generering, åtminstone i grammatikaliskt hänseende. Detta kanske närmast är en skojig demonstration, men visar också att de ordklass-liknande klasser som ICA hittar inte bara är något påhittat, utan verkligen något som kan användas för att skapa (nästan) grammatikaliskt språk.

Enligt vår undersökning hittade ICA sådana komponenter som till en del svarade mot klasser i vårt genererade korpus. PCA hittade också en liknande struktur, vilket tyder på att grammatiken var för enkel. Man kan fråga sig hur man kunde förbättra sökandet efter klasser. Vi undersökte hur användandet av vänsterkontext och högerkontext förbättrade resultatet. Längre kontexter kunde därför vara intressanta att undersöka. Det finns också ett behov av att förbättra den matematiska modell vi använt, så att det man hittar skulle likna ännu mer de intuitiva klasser vi söker. Det skulle också behövas experiment som jämför resultaten vi fått med resultat från naturligt språk. Våra enkla grammatiker simulerar språket till en viss grad, men är en väldigt drastisk förenkling av det verkliga språket. Finns fenomenen vi upptäckt också i naturligt språk? Fungerar ICA lika då också? Man kunde exempelvis ordna orden enligt en förut känd ordklass och se hur ICA-komponenterna varierar inom klassen. Är det som för vårt korpus att komponentfördelningen är mer avgörande än några få komponenter med stort absolut värde? Svaren på dessa frågor skulle ge oss möjlighet att förbättra våra metoder, och att bättre förstå den statistiska strukturen hos naturligt språk.

De statistiska metoderna strävar från text till betydelse. Vi är övertygade om att de behöver möta något som strävar från betydelse till text. Processerna bakom dessa två är båda oerhört komplicerade, men den ena kan antagligen ge insikt om den andra. Det finns ännu mycket att göra innan de naturliga språken är lösta till den grad att man maskinellt kan efterlikna språklig förmåga.

Referenser

[Bailey, 1997] David Bailey (1997). *When Push Comes to Shove: A Computational Model of the Role of Motor Control in the Acquisition of Action Verbs.* PhD thesis, Computer Science Division, University of California, Berkeley.

[Bangalore m.fl., 2000] Srinivas Bangalore, Owen Rambow, och Steve Whittaker (2000). Evaluation metrics for generation. In *International Conference on Natural Language Generation (INLG 2000) , Mitzpe Ramon, Israel*.

[Barzilay, 2003] Regina Barzilay (2003). *Information Fusion for Multidocument Summarization: Paraphrasing and Generation.* PhD thesis, Columbia University.

[Barzilay och Lee, 2002] Regina Barzilay och Lillian Lee (2002). Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of EMNLP*, sid. 164–171.

[Barzilay och Lee, 2003] Regina Barzilay och Lillian Lee (2003). Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT*.

[Bateman och Zock, 2001] John Bateman och Michael Zock (2001). *Natural Language Generation*, kapitel 17. Oxford University Press.

[Bateman, 1997] John A. Bateman (1997). Enabling technology for multilingual natural language generation: the kpml development environment. *Journal of Natural Language Engineering*, 3:15–55.

[Bates, 1999] Elizabeth Bates (1999). *Plasticity, localization and language development.*, sid. 214–253. New York: Oxford University Press.

[Bontcheva, 2003] Kalina Bontcheva (2003). Reuse and problems in the evaluation of nlg systems. In *Proceedings of EACL'03 Workshop on Evaluation Initiatives*.

[Busemann, 1996] Stephan Busemann (1996). Best-first surface realization. In *Proceedings of the Eighth International Natural Language Generation Workshop (INLG '96), Herstmonceux, Sussex*, sid. 101–110.

[Buss, 1973] Arnold H. Buss (1973). *Psychology: Man in Perspective*. New York: Wiley.

[Chalmers, 2002] David J. Chalmers (2002). Varieties of emergence. Templeton Foundation workshop on emergence in Granada, August.

[Chomsky, 1956] Noam Chomsky (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124.

[Chomsky, 1957] Noam Chomsky (1957). *Syntactic Structures*. Mouton, The Hague.

[Chomsky, 1986] Noam Chomsky (1986). *Knowledge of Language: Its Nature, Origin and Use*. New York: Prager.

[CoGenTex, 2000] CoGenTex (2000). Realpro general English grammar user manual. http://www.cogentex.com/papers/realpro-manual.pdf.

[Deerwester m.fl., 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, och Richard A. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(16):391–407.

[Dewdney m.fl., 2001] Nigel Dewdney, Carol VanEss-Dykema, och Richard MacMillan (2001). The form is the substance: Classification of genres in text. Workshop on HLT and KM, ACL.

[Dougherty, 1994] Ray C. Dougherty (1994). *Natural Language Computing An English Generative Grammar in Prolog*. Lawrence Erlbaum Assoc.

[Edelman m.fl., 2003] Shimon Edelman, Zach Solan, David Horn, och Eytan Ruppin (2003). Rich syntax from a raw corpus: Unsupervised does it. a position paper to be presented at Syntax, Semantics and Statistics; a NIPS-2003 workshop.

[Elhadad och Robin, 1999] Michael Elhadad och Jacques Robin (1999). Surge: a comprehensive plug-in syntactic realization component for text generation. *Computational Linguistics*, 99(4).

[Elman, 1990] Jeff L. Elman (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

[Feldman och Narayanan, 2004] Jerome Feldman och Srinivas Narayanan (2004). Embodied meaning in a neural theory of language. *Brain and Language*, 89:385–392.

[Finn och Kushmerick, 2003] Aidan Finn och Nicholas Kushmerick (2003). Learning to classify documents according to genre. IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis (Acapulco).

[Frege, 1879] Gottlob Frege (1879). *Begriffschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle a. S.: Louis Nebert.

[Fry, 2004] John Fry (2004). Context-free grammars for english. Lecture slides for Linguistics 165: Computers and Written language, San Jose State University.

[Gold, 1967] E. Mark Gold (1967). Language identification in the limit. *Information and Control*, 10(5):447–474.

[Gärdenfors, 2000] Peter Gärdenfors (2000). *Conceptual Spaces - The Geometry of Thought*. MIT Press.

[Hardin, 1988] C.L. Hardin (1988). *Color for Philosophers - Unweaving the Rainbow*. Hackett Publishing Company.

[Harris, 1951] Zellig S. Harris (1951). *Structural Linguistics*. University of Chicago Press, Chicago: IL USA and London, uk, 7th (1966) edition edition.

[Haykin, 1999] Simon Haykin (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, second edition.

[Hebb, 1949] Donald O. Hebb (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.

[Honkela m.fl., 2003] Timo Honkela, Aapo Hyvärinen, och Jaakko Väyrynen (2003). Emergence of linguistic representations by independent component analysis. Technical report, A72, Helsinki University of Technology, Laboratory of Computer and Information Science.

[Honkela m.fl., 2004] Timo Honkela, Aapo Hyvärinen, och Jaakko Väyrynen (2004). Emergence of linguistic features: Independent component analysis of contexts. In *A. Cangelosi et al. (eds.), Proceedings of NCPW9, Neural Computation and Psychology Workshop, Plymouth, England*.

[Honkela m.fl., 1996] Timo Honkela, Samuel Kaski, Krista Lagus, och Teuvo Kohonen (1996). Newsgroup exploration with websom method and browsing interface. Technical report, A32, Helsinki University of Technology, Laboratory of Computer and Information Science.

[Honkela och Kohonen, 2005] Timo Honkela och Oskar Kohonen (förbereds, utkommer senare under 2005). Learning to translate. Technical report, Helsinki University of Technology, Laboratory of Computer and Information Science.

[Honkela m.fl., 1995] Timo Honkela, Ville Pulkki, och Teuvo Kohonen (1995). Contextual relations of words in grimm tales, analyzed by self-organizing map. In *Proceedings of ICANN-95*, sid. 3–7.

[Hutchins, 1986] John Hutchins (1986). *Machine Translation: past, present, future*. New York: Halsted Press.

[Hyvärinen m.fl., 2001] Aapo Hyvärinen, Juha Karhunen, och Erkki Oja (2001). *Independent Component Analysis*. John Wiley & Sons.

[Hyvärinen och Karthikesh, 2002] A. Hyvärinen och R. Karthikesh (2002). Imposing sparsity on the mixing matrix in independent component analysis. *Neurocomputing*, 49:151–162.

[Hyvärinen och Oja, 1997] Aapo Hyvärinen och Erkki Oja (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492.

[Hyvärinen och Oja, 2000] Aapo Hyvärinen och Erkki Oja (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13:411–430.

[Johnson, 2001] Steven Johnson (2001). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Touchstone, Rockefeller Center, NY.

[Karlgren, 2000] Jussi Karlgren (2000). *Stylistic Experiments for Information Retrieval.* PhD thesis, Stockholm University, department of linguistics.

[Kaski m.fl., 1998] Samuel Kaski, Timo Honkela, Krista Lagus, och Teuvo Kohonen (1998). Websom - self-organizing maps of document collections. *Neurocomputing*, 21:101–117.

[Kaski och Sinkkonen, 2004] Samuel Kaski och Janne Sinkkonen (2004). Principle of learning metrics for data analysis. *Journal of VLSI Signal Processing*, 31:177–188.

[Kohonen m.fl., 2005] Oskar Kohonen, Sakari Katajamäki, och Timo Honkela (2005). In search for volta: Statistical analysis of word patterns in shakespeare's sonnets. In *Proceedings of the AKRR'05.*

[Kohonen, 1982] Teuvo Kohonen (1982). Self-organizing formation of topologically correct feature maps. *Biol. Cyb.*, 43(1):59–69.

[Kohonen, 1990] Teuvo Kohonen (1990). The self-organizing map. In *Proceedings of the IEEE*, volume 78, sid. 1464–1480.

[Kohonen, 2001] Teuvo Kohonen (2001). *Self-Organizing Maps.* Springer, Berlin, Heidelberg, New York, third extended edition edition.

[Koppel m.fl., 2003a] Moshe Koppel, Navot Akiva, och Ido Dagan (2003a). A corpus-independent feature set for style based text categorization. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis, Acapulco, Mexico.*

[Koppel m.fl., 2003b] Moshe Koppel, Shlomo Argamon, och Anat R. Shimoni (2003b). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17:401–412.

[Lagus m.fl., 2002] Krista Lagus, Anu Airola, och Mathias Creutz (2002). Data analysis of conceptual similarities of finnish verbs. In *Proceedings of the CogSci 2002, the 24th annual meeting of the Cognitive Science Society*, sid. 566–571.

[Lakoff och Johnson, 1980] George Lakoff och Mark Johnson (1980). *Metaphors We Live By.* Univ. of Chicago Press (Chicago).

[Lakoff och Johnson, 1999] George Lakoff och Mark Johnson (1999). *Philosophy in the Flesh - The Embodied Mind and its Challenge to Western Thought.* New York: John Wiley.

[Landau och Gleitman, 1985] Barbara Landau och Lila R. Gleitman (1985). *Language and Experience: Evidence from the Blind Child.* Cambridge, MA: Harvard University Press.

[Langkilde och Knight, 1998] Irene Langkilde och Kevin Knight (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of COLING-ACL.*

[Lennberg, 1967] Eric H. Lennberg (1967). *Biological Foundations of Language.* New York: John Wiley.

[Levelt, 1989] Willem J. M. Levelt (1989). *Speaking: From Intention to Articulation.* MIT Press.

[Luger och Stubblefield, 1994] George F. Luger och William A. Stubblefield (1994). *Artificial Intelligence - Structures and Strategies for Complex Problem Solving.* Addison Wesley, second edition edition.

[Maass och (eds.), 1998] Wolfgang Maass och Christopher M. Bishop (eds.) (1998). *Pulsed Neural Networks.* MIT Press, Cambridge, MA.

[Mann och Thompson, 1988] William C. Mann och Sandra A. Thompson (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8:243–281.

[Manning och Schütze, 1999] Christoph D. Manning och Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

[MathWorks, 2005] MathWorks (2005). The Mathworks - MATLAB and Simulink for technical computing. http://www.mathworks.com/.

[McKeown, 1985] Kathleen McKeown (1985). *Text Generation*. Cambridge University Press.

[Minsky, 1961] Marvin L. Minsky (1961). Steps toward artificial intelligence. *Proceedings of the Institute of Radio Engineers*, 49:8–30.

[Minsky och Papert, 1969] Marvin L. Minsky och Seymour A. Papert (1969). *Perceptrons*. Cambridge, MA:MIT Press.

[Narayanan, 1997] Srinivas Narayanan (1997). *KARMA: Knowledge-based active representations for metaphor and aspect*. PhD thesis, Computer Science Division, University of California, Berkeley.

[Narayanan, 1999] Srinivas Narayanan (1999). Moving right along: A computational model of metaphoric reasoning about events. In *Proceedings of the National Conference on Artificial Intelligence AAAI-99*.

[Pinker, 1984] Stephen Pinker (1984). *Language Learnability and Language Development*. Cambridge, MA:MIT Press.

[Powers, 1996] David M. Powers (1996). What unsupervised learning tells us about language models. Symposium on Tacit Assumptions in the Study of Language, Helsinki, September 1996.

[Ratnaparkhi, 2000] Adwait Ratnaparkhi (2000). Trainable methods for surface natural language generation. In *Proceedings of the NAACL*, sid. 194–201.

[Reiter, 1994] Ehud Reiter (1994). Has a consensus NL generation architecture appeared, and is it psychologically plausible? In David McDonald och Marie Meteer, editors, *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*, sid. 163–170, Kennebunkport, Maine.

[Reiter och Dale, 1999] Ehud Reiter och Robert Dale (1999). Eacl-99 tutorial on building natural language generation systems.

[Reiter och Sripada, 2003] Ehud Reiter och Somayajulu G. Sripada (2003). Learning the meaning and usage of time phrases from a parallel text-data corpus. In *Proceedings of the HLT-NAACL03 Workshop on Learning Word Meaning from Non-Linguistic Data*, sid. 78–85.

[Reiter m.fl., 2003] Ehud Reiter, Somayajulu G. Sripada, och Roma Robertson (2003). Acquiring correct knowledge for natural language generation. *Journal of Artificial Intelligence Research*, 18:491–516.

[Ritter och Kohonen, 1989] Helge Ritter och Teuvo Kohonen (1989). Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254.

[Rosenblatt, 1958] Frank Rosenblatt (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

[Rumelhart och McClelland, 1986] David E. Rumelhart och James L. McClelland (1986). *Parallell Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA:MIT Press.

[Sebastiani, 2002] Fabrizio Sebastiani (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

[Shannon och Weaver, 1949] Claude E. Shannon och Warren Weaver (1949). *The Mathematical Theory of Communication.* University of Illinois Press.

[Solan m.fl., 2004] Zach Solan, David Horn, Eytan Ruppin, och Shimon Edelman (2004). Unsupervised context sensitive language acquisition from a large corpus. In Sebastian Thrun, Lawrence Saul, och Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA.

[Tarski, 1944] Alfred Tarski (1944). The semantic conception of truth and the foundation of semantics. *Philos. and Phenom. Res.*, 4:341–376.

[Tomasello, 2000] Michael Tomasello (2000). The item-based nature of children's early syntactic development. *Trends in Cognitive Sciences*, 4.

[Turing, 1950] Alan Turing (1950). Computing machinery and intelligence. *Mind*, 59.

[Ultsch, 1994] Alfred Ultsch (1994). *Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series*, sid. 33–46. Elsevier.

[van Zaanen, 2000] Menno van Zaanen (2000). Bootstrapping syntax and recursion using alignment-based learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, sid. 1063–1070. Stanford University, Morgan Kaufmann Publishers.

[van Zaanen, 2002] Menno van Zaanen (2002). *Bootstrapping Structure into Language: Alignment-Based Learning.* PhD thesis, University of Leeds, Leeds, UK.

[Vapnik, 1998] Vladimir N. Vapnik (1998). *Statistical Learning Theory.* New York: Wiley.

[von Neumann, 1958] John von Neumann (1958). *The Computer and the Brain.* CT: Yale University Press.

[Väyrynen m.fl., 2004] Jaakko Väyrynen, Timo Honkela, och Aapo Hyvärinen (2004). Independent component analysis of word contexts and comparison with tradional categories. In *Proc. of the 6th Nordic Signal Processing Symposium (NORSIG 2004), Espoo, Finland*, sid. 300–303.

[Väyrynen och Honkela, 2005] Jaakko J. Väyrynen och Timo Honkela (2005). Comparison ofindependent component analysis and singular value decomposition. In *Proceedings of the AKRR'05.*

[Weizenbaum, 1966] Joseph Weizenbaum (1966). Eliza–a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9.

[Werbos, 1974] Paul J Werbos (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* PhD thesis, Harvard University, Cambridge, MA.

[Wiener, 1948] Norbert Wiener (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine.* New York: Wiley.

[Zuidema, 2003] Willem Zuidema (2003). How the poverty of the stimulus solves the poverty of the stimulus. In *Proceedings of NIPS'02*, sid. 51–58.