Benny Brodda
Institutionen för Lingvistik
Stockholms Universitet
S-106 91 Stockholm

RÄTTSINFORMATIK OCH LINGVISTIK

0. Inledning

I denna rapport skall ges en kort översikt över projektet Rättsinformatik och Lingvistik, vilket är ett samarbetsprojekt mellan Institutet för Rättsinformatik (IRI) och Institutionen för Lingvistik, båda vid Stockholms Universitet, och bekostat av medel från DFI (400.000:- för budgetåren 83/84-84/85). Huvudansvarig för projektet i sin helhet är prof Peter Seipel, IRI, och huvudansvarig från Lingvistiska Institutionen är författaren. Av naturliga skäl skall jag här i huvudsak beröra de (dator)lingvistiska delarna av projektet.

Men först litet bakgrundsinformation. Allsedan början av 70-talet har man under överinseende av Samarbetsorganet för Rättsväsendets Informationssystem (SARI) systematiskt byggt upp publikt tillgängliga databaser innehållande bl a allt nytt SFS-material (dvs väsentligen de nya lagarna), material från domstolsverket (sedan 1980 t ex de fulla texterna till rättsfallsreferaten HD och hovrätter, Regeringsrätt och kammarrätter samt från arbetsdomstolen och bostadsdomstolen). Rent fysiskt administreras dessa informationssystem av den statliga datamaskinscentralen DAFA. DAFA handhar också ett stort antal andra, icke publika, statliga informationssystem, men dem kan vi lämna därhän i detta sammanhang. De publikt tillgängliga informationssystemen omfattar i dag (vid årsskiftet 83/84) databaser om totalt c:a 30 miljoner ord löpande text, och de har för närvarande en tillväxttakt av flera miljoner ord per år. (Uppgifterna här ovan är hämtade ur RÄTTSDATA, utveckling, nuläge och framtid, Justitiedepartementet, Jan 1984.)

Denna tillväxttakt kan förväntas öka ganska kraftigt de närmaste åren, för man diskuterar nu möjligheterna att dels lägga in tidigare material än det som nu finns där, dvs SFS-material från tiden före 1970, domstolsmaterial från tiden före 1980, mer full-

Rättsinformatik och lingvistik
Benny Brodda
Proceedings of NODALIDA 1983, pages 48-61

ständigt riksdagsmaterial (fn lagras enbart register över riksdagstrycket), samt börja föra in material som i dag inte alls finns där: föreskrifter, anvisningar, material från andra domstolar än de ovan uppräknade, på sikt också lagförarbeten, relevant juridisk litteratur mm, mm. Det är långt ifrån orealistiskt att gissa att dessa informationssystem kan komma att omfatta någonting i storleksordningen en miljard löpord vid slutet av 90-talet. Till yttermera visso diskuterar man redan idag möjligheterna att på ett eller annat sätt också länka dessa system till andra informationssystem inom och utom landet, särskilt då motsvarande system i våra nordiska grannländer. Etc, etc. Det är alltså ganska gigantiska informationssystem som är under uppbyggnad.

1. Rättsinformatik och Lingvistik

Har nu detta något med lingvistik att skaffa? I högsta grad, ty den grundläggande frågeställningen här är naturligtvis exakt densamma som i alla IR-system (IR=Information Retrieval), nämligen problemet att finna alla dokument (i en given dokumentsamling) som handlar om en viss sak, och helst bara dem. Denna frågeställning är i sin tur mycket nära besläktad med den lingvistiska frågeställningen om hur man med formella och automatiska metoder skall beskriva och känna igen det semantiska innehållet i en given text, en frågeställning som också råkar vara en av de mest aktuella just nu inom den komputationella lingvistiken. Skillnaden här från grundforskning som den normalt bedrives inom lingvistiken är först och främst skalan, men också att behovet av fungerande lösningar är överhängande. (Insikten om att denna koppling finns mellan IR och lingvistik är naturligtvis inte ny; jfr - t ex - Brodda & Karlgren 1965.)

Det finns också skäl som gör just Rättsinformatiken intressant. Rättsinformationssystemen är förmodligen de informationssystem i Sverige som i dag är snabbast växande, vilket medför att behovet av bra lösningar är mer överhängande för detta område än något annat. I den internationella IR-forskningen diskuterar man vanligtvis informationsåtervinningssystem för vetenskaplig litteratur, ett självklart både intressant och viktigt område. Sådana system är dock vanligtvis inriktade på återvinning av engelskspråkiga texter (eller möjligtvis texter från något annat internationellt språk-

område), medan rättsinformationssystem med nödvändighet måste vara inriktade på språket i det aktuella landet, i Sverige alltså svenska. Här finns alltså ett direkt behov av utvecklandet av datorlingvistiska metoder för analys av svenska, ett språk som det av naturliga skäl inte forskas så väldigt intensivt om utanför Sveriges gränser.

Det finns också andra aspekter av detta. Det kan väl inte ha undgått någon att det pågått en mycket intensiv debatt om datoriseringen och ett förment därmed förknippat hot mot demokratin. Och det är klart, den som i framtiden behärskar dessa mycket stora informationssystem, han kommer att ha ett oerhört försteg före alla andra bara genom att veta mer.

Enligt mitt förmenande kan man i dag skönja två ganska klara och varandra motstridiga utvecklingstendenser i samhället i detta avseende. Den ena är just den som väl vanligen diskuteras, nämligen en hotande utvecklingen mot en teknokratiskt styrd oligarki, ett fåtalsvälde där just de mimarober som behärskar de enorma informationssystemen innehar nyckeln till makten. Den andra utvecklingstendensen är minst lika stark, tycker jag, men kanske inte lika mycket uppmärksammad i den allmänna debatten, nämligen tendensen mot en verkligt öppen demokrati, där i princip alla medborgare har tillgång till lika mycket information som vilken expert som helst. När kabel-TV-systemen är utbyggda kommer vi dessutom att ha dessa system tillgängliga hemifrån via höghastighetsterminaler.

Hur det går är väldigt avhängigt av hur vi agerar i dag, det är nu vi formar utvecklingslinjerna för det framtida samhället, och här tycker jag man måste diskutera forskningens - och forskarens - ansvar. Den humanistiska språkvetenskapen kan här hjälpa till att utveckla systemen till sant mänskliga och lättåtkomliga informationssystem för envar. Det här är alldeles för allvarliga problem för att lämnas till militären och datamaskinsfabrikanterna att göra upp om på stängda kontor.

2. Nya, okonventionalla angreppssätt efterlyses

Jag skall i det här avsnittet genom ett par enkla tillämpningar av vanlig, hederlig reguladetri visa att helt nya angreppssätt på informationsssökningsproblematiken är av nöden påkallade. När det

gäller existerande, i praktiskt bruk varande informationssöknings-system så arbetar dessa nästan undantagslöst enligt den booleska principen: sökfrågan utformas som ett booleskt uttryck på ett antal sökord eller deskriptorer, och systemets "svar" utgöres av de artiklar där orden eller deskriptorerna förekommer i den sökbara delen av artiklarna i enlighet med det angivna booleska villkoret.

Det finns ganska goda skäl till varför booleska söksystem är så spridda: sökmetodiken är konceptuellt enkel, och det behövs följaktligen väldigt litet av inlärning för att förstå den. De arbetar med en enkel, välutprovad och välfungerande "teknologi", som bl a erbjuder snabba svarstider samt, icke minst viktigt, erbjuder en enkel lösning på uppdateringsproblematiken. Allt detta gör nog att boolesk sökning för överskådlig tid kommer att utgöra en viktig komponent i framtida söksystem .

Varför duger då inte sådana system allt framgent? Ja, förenklat uttryckt kan man säga att de inte klarar uppskalning utan vidare. Ett enkelt räkneexempel kan få illustrera detta. Ett typiskt söksystem har, säg, 10.000 dokument i databasen. Antag sedan att systemet på en viss given sökfråga producerar 20 referenser som svar. Detta är en både typisk och rimlig situation. Om man söker med samma sökfråga i ett analogt system men nu med 10 miljoner dokument i databasen skulle man få 20.000 referenser som svar vid bibehållen precision i sökmekanismen. Detta är inte längre rimligt. Visserligen kan man foga till ytterligare booleska villkor i sin sökfråga, men mycket tyder på att booleska system erbjuder alltför trubbiga instrument för att tillåta en att mejsla ut tillräckligt finstrukturerade svarsmängder i dokumentrymden. (Jfr Walker & Karlgren & Kay 1973.)

Vad är då vägen ut ur detta dilemma? Ja, jag har ju ovan påpekat att IR-problematiken i sig är ekvivalent med att definiera (och känna igen) det semantiska innehållet i text, och på något sätt är det ju just det som den komputationella lingvistiken i dag explicit ser som ett av sina kanske mest centrala forskningsområden, alltså syntaktisk/semantisk parsing, och det ligger nära till hands att börja undersöka om metoder hämtade från detta forskningsområde kunde vara tillämpliga i det här sammanhanget. På sikt är detta säkert möjligt, men inom en mer överskådlig framtid är jag mer pessimistisk, åtminstone om man rör sig inom den mer storskaliga miljö jag diskuterat. Denna pessimism grundar jag på rena

performansöverväganden: algoritmer rapporterade i litteraturen anger ofta 1 sekund per ord och väl det som typiska värden för processning av löpande text, och det även med mycket begränsade lexika och mycket begränsade textkorpora; exempel på sådana algoritmer återfinns i denna volym.

Med en algoritm som tar en sekund per ord att processa löpande text, och med en miljard ord i textbasen tar det faktiskt drygt 30 år att ladda sökdatabasen (1 miljard sekunder = 31.7 år, för att vara exakt), och det säger sig självt att detta inte är intressant i något som helst praktiskt sammanhang. Till yttermera visso är det idag högst oklart vad man skulle göra med sina parsningsträd när man väl är klar; jfr Walker & Karlgren & Kay. (Det man efterlyser är en "metrik" över mängden av sådana här grafer, med vars hjälp man kan definiera semantiskt avstånd mellan i första hand meningar och i andra hand grupper av (textkonstituerande) meningar. Någon effektiv sådan metrik är mig veterligt inte beskriven någonstans. (Se vidare avsn 6, nedan.)

Å andra sidan är det inte så lätt att säga vad man skall göra i stället. I själva verket torde det inte finnas något enskilt "guldägg" som i ett enda slag löser alla problem, utan man får snarare rikta in sig på att tålmodigt pröva sig fram efter många olika vägar och ta vara på varje tänkbart uppslag. Det kommer att bli den samlade kören av åtgärder som (i bästa fall) ger en totalperformans som är den man skulle vilja ha eller har anledning att förvänta sig. En sak kan dock kanske vara värd att påpeka i det här sammanhanget, nämligen att den problemställning vi diskuterar är fullständigt resultatinriktad - ett bra söksystem är bra oavsett hur det fungerar inne i maskinen. Detta medför att det är fritt fram att komma med hur okonventionella analysmetoder som helst, fungerar de så fungerar de. Och den saken är klar, skall man åstadkomma något radikalt genombrott här, så måste man vara okonventionell; traditionella metoder inom den komputationella lingvistiken tycks alltid ha de inherenta performansbegränsningar jag ovan påpekade. (Därmed inte sagt att jag tror på ad hoc metoder - den som har friska och djärva och rimliga hypoteser om hur människan bär sig åt när hon avgör att två texter handlar om samma sak och har förmåga att fånga dessa hypotser i en algoritm har nog bättre chanser att lyckas än den som "bara" är duktig på att sätta ihop fiffiga hashtabeller eller vad det nu kan vara som han/hon är duktig på.)

I det följande skall jag ge en kort översikt över några av de delprojekt av datalingvistisk natur som är planerade eller påbörjade inom ramen för projektet Rättsinformatik och Lingvistik.

3. Heuristisk Parsning

De ursprungliga principerna för den variant av heuristisk analys som vi börjat tillämpa vid Institutionen för Lingvistik, SU, finns först beskrivna i Brodda, 1979, och där finns också redogjort för ett system enligt dessa principer tillämpat på morfologisk analys av svenska. Principerna är senare ytterligare utvecklade i Brodda, 1983, och där ges också en skiss till en tillämpning på syntaktisk nivå. Källgren har sedan ytterligare följt dessa riktlinjer, och bl a utvecklat ett mer fullständigt system för ytsyntaktisk parsning; se denna volym och Källgren 1984.

Heuristisk parsning enligt vår uppfattning av denna innebär att man inte följer en strikt algoritm, utan låter flera av varandra oberoende och parallella processer gripa in i analysen enligt ett ganska stokastiskt mönster. Hitintills har vi också envist hållit fast vid att så långt som möjligt enbart utnyttja information på språkets ytnivåer ("analys utan lexikon"), dock inte så mycket av princip utan fastmer av nyfikenhet på att se hur långt man kan driva analysen med enbart ytkriterier. Enligt vårt sätt att se det hela innebär problemet med att foga till olika typer av lexika bara att länka in ytterligare processer i schemat, parallellt med andra strukturigenkännande processer. Systemmässigt innebär det alltså ingen artskillnad i vårt heuristiska schema att använda lexikon (alternativt låta bli), utan skillnaden kommer bara att synas som en skillnad i performans. Härigenom kan vi alltså exakt se vad lexikonen bidrar med för slags information (i informationsteoretisk mening) - och vad de kostar (i körtid).

I nästa avsnitt presenteras en direkt tillämpning av den heuristiska metoden.

4. Automatisk indexering

Ovan antydde jag att det är/var mer eller mindre av vetenskaplig nyfikenhet som vi försökte hålla oss till strikt ytstrukturell analys. Detta är naturligvis en sanning med modifikation. I själva verket hade vi flera skäl att pröva den linjen. Ett var väl just

vetenskaplig nyfikenhet, men där hade vi också en bestämd hypotes, nämligen att ytstrukturen borde kunna tappas på bra mycket mer information än vad som hitintills antagits. Veterligt har man i praktiskt taget alla parsningssystem som diskuterats i litteraturen nästan axiomatiskt utgått från att man måste ha ett (stam)lexikon med som bas för analysen, och det var det axiomet vi ville utmana, åtminstone så långt analysen rör det rent syntaktiska. (Kommer man in på den semantiska analysen måste man ju rimligen ha ett riktigt lexikon med - lexikonet utgör ju basen för den semantiska komponenten.) Hitintills tycker jag nog att våra datorexperiment i vart fall inte jävat den hypotesen; jfr Källgren, denna volym. Jfr också Hornstrand 1983 där ett experiment redovisas för att utröna i vad mån försökspersoner känner igen satsers syntax med utgångspunkt från samma typ av information som våra parsningssystem utnyttjar, och inte heller denna undersökning jävar vår grundläggande hypotes. (Samma experiment håller just nu - VT 84 - på att upprepas med engelskt material, och det är ytterst intressant att notera, att de preliminära resultaten helt motsvarar de för svenska.)

Ett annat skäl till att pröva den ytstrukturella ansatsen var att i den mån den lyckades så bör man rimligen kunna få relativt snabba system. (Att slå i stora, skivminneslagrade lexika är en inte alldeles kostnadfri sysselsättning i datasammanhang.) De experiment i den riktning vi hitintills genomfört tyder på att den typ av parsing som redovisas i Källgren, denna volym, bör kunna drivas därhän att man får uppåt en 85-90 % korrekt genererade ytträd ur en godtycklig, opreparerad text med en performans av kanske 100 ord/s (på en DEC-10:a). Nu är det klart att i dagens läge har vi samma problem med dessa ytträd som man i IR-sammanhang har med varje form av syntaktiska/semantiska strukturer erhållna ur löpande text, nämligen: Vad skall man göra med dem?

En användning av dem har vi dock funnit i ett av delprojekten under projektet Rättsinformatik och Lingvistik, i ett experiment med automatisk indexering kallat Substantivjakten, närmare redovisat i Källgren, 1984. Ideen vi ville testa var att se i hur hög grad substantiv excerperade ur en löpande text kunde fungera som indikatorer på innehållet i texten i fråga. Detta experiment är fortfarande under utvärdering, men preliminärt törs vi nog påstå att substantiven tycks vara utomordentligt goda indikatorer på textens innehåll, åtminstone som i det här fallet juridiska texter

(närmare bestämt lagtexter), medan t ex verben excerperade på samma sätt ger mycket svaga ledtrådar om innehållet. (Det kan kanske också vara värt att påpeka att användare i mycket stor utsträckning utnyttjar just substantiv i sina sökuttryck.)

Låt mig kort antyda på vad sätt den heuristiska parsningen kan utnyttjas för att identifiera substantiv. Vissa ord signalerar själva att de är substantiv: MYNDIGHETEN, TIDNINGAR, LAGARNA. Ordsluten ("kadenserna", jfr Brodda, 1979, 1982) -(IG)HETEN, -NINGAR, -ARNA fungerar alla som ganska starka indikatorer på substantiv. Andra typer av kadenser är i och för sig också indikatorer på substantiv, men inte särskilt starka: jfr -ER i MAGER, NEGER, NIGER, SÖNDER och BÖNDER, vilket betyder att det är mycket vanskligt att utgå från att ord på -ER är substantiv. Tillsammans med andra ytelement i omgivningen kan de dock förvandlas till starka indikatorer: ALLA (SVERIGE)S (BÖND)ER. Ibland är det bara sådana kringliggande ytelement som indirekt pekar ut ett ord som ett substantiv: ETT (LITE)T (HUS) PÅ (LAND)ET. HUS och LANDET blir här utplockade som substantiv, LANDET enligt konfigurationen PÅ -ET, HUS helt på grundval av indirekta indikationer.

5. "Huru känna igen ord ute i texten fastän de är böjda?"

Som jag tidigare påpekade så är alla i dag i praktiskt bruk varande söksystem väsentligen sk booleska söksystem. En sökfråga till systemet utgöres av ett antal sökord hopkopplade med booleska villkor avseende ordens förekomster i de sökta texterna. Ett problem i det här sammanhanget är att orden ute i texterna inte ser ut på det sätt som det angavs i sökfrågan. Det naturliga är nämligen att man ger sökorden i sin grundform i sökfrågan (eller i vart fall i en enda form), men att det sedan är stor sannolikhet för att orden återfinns i texten i andra former än de man angav. Problemet är då att ändå känna igen orden som lika.

Det traditionella sättet att göra detta är att utnyttja någon form av trunkering. Antag att ett av sökorden är REGEL och att man vill finna alla instanser av det ordet, men också REGELN, REGLER, REGLERNAS... Man kan ju då i och för sig ange alla tänkbara böjningsvarianter själv, men detta kan vara ganska arbetsamt - i en faktisk söksession kan man behöva ange sökfrågor i flera omgångar, var och en innehållande flera sökord - och dessutom är det ju lätt

att glömma någon i hastighten. Trunkering innebär att man anger sökordet t ex som REG$, därmed menande alla ord som inleds med strängen REG (detta kallas högertrunkering); i det aktuella fallet får man då (bl a) de böjda formerna av REGEL som träff.

Trunkering är tyvärr ett ganska trubbigt instrument; i det ovan angivna fallet skulle vi också få REGERING, REGN, REGEMENTE m fl som träff förutom de sökta. Problemet är alltså hur man skall finna bara sökordet och dess böjningsvarianter men inga andra ord (förutom då eventuella homonymer; dessa kan man ju aldrig komma ifrån).

Nu finns det ett par, tre olika sätt att åstadkomma detta. Ett sätt är att gå igenom hela databasen och för alla ord i den löpande texten utföra en (manuell eller) automatisk morfologisk analys, resulterande i en (rot)lemmatisering, alltså en identifiering av de grundord som ordet i fråga är uppbyggt av. Dessa lemmatiserade ord är sedan de som ingår i dokumentindexet, alltså de strukturer som sökningen utförs på. Ett sådant förfarande beskrives av Fjeldvig & Golden, denna volym. Jfr också Brodda, 1982, där problemet att känna igen sammansättningar med utgångspunkt från heuristiska principer diskuteras.

Ett annat sätt är att man utifrån det angivna sökordet genererar de möjliga böjningsstammarna, alltså de former av ordet som ändelser hängs på. Lemmat REGEL har böjningsstammarna REGEL och REGLE. På den förra kan ändelserna 0, -N, -S, -NS hängas, på den senare -R, -RNA, -RS och -RNAS. På samma sätt har BONDE dels BONDE dels BÖNDE som böjningsstammar. Denna ansats har visat sig vara mycket effektiv för finskan (Karlsson & Koskenniemi, personlig ref) och torde också framgångsrikt kunna tillämpas på svenskan.

Problemet att generera böjningsstammarna är i stort sett ekvivalent med att identifiera böjningsklass. I och för sig kan man genom direkt lexikonslagning få reda på den (och för de oregelbundna orden är detta nästan den enda möjligheten, men dessa representerar ett marginellt problem i svenskan), men i viss utsträckning kan man också direkt av ordets form predicera dess böjningsklass; ord som slutar på -NING är 2:a deklinationen, ord på (IG)HET är 3:e etc. Värre är det när man på detta sätt inte har ett morfologiskt stöd, men där gäller ofta vissa "default"-förhållanden. Så t ex kan man defaultmässigt anta att tvåstaviga ord slutande på -E tillhör 2:a deklinationen, och undantagen (som måste testas mot ett lexikon) är ett femtiotal neutrer på -E (VÄRDE, VITTNE, SÄTE, ...).

Etc.

Det vi närmast tänkte testa är ett förfarande som man lite löst kunde kalla "ordsubtraktion". Om man bokstav för bokstav "subtraherar" sökordet REGEL från textordet REGLER får man en "slatt" kvar i vardera ordet, en residualsträng, nämligen EL i det första och LER i det andra. Nu råkar det vara så att bägge dessa två residualsträngar är typiska ordslut till ord i samma böjningskategori, och vi kan ta det som kriterium för att träff föreligger. Gentemot t ex ordet REGERING får vi residualsträngarna EL resp ERING, och dessa är inte typiska kadenser till ord i samma böjningsklass. REGEL och REGERING är alltså inte böjningsvarianter av varandra. (Tekniskt innebär ordsubtraktionsprincipen att ord lokaliseras till sin position i texten via hashtabeller utgående från ordens invarianta stam. I ordet REGEL är orddelen REG hela tiden oförändrad, REG är ordets invarianta stam. På samma sätt har ordet POJKE POJK som invariant stam. Etc.)

Jag skulle kanske kort beröra andra typer av trunkering också. Vid sidan av den typ jag ovan diskuterade (som alltså inte finns tillgänglig i söksystem i dag) utnyttjar man vanligtvis traditionell trunkering. Denna innebär att systemet uppsöker de ord som inleds med den angivna strängen. Detta betyder att också sammansättningar med den angivna strängen som första led hittas, och en sådan funktion är naturligtvis mycket användbar och måste rimligen också kunna erbjudas. Därför tänker jag mig att man skall kunna ange t ex REGEL$ för äkta högertrunkering (som ger ord av typ REGELBOK) och REGEL= för att ange att det enbart är böjningsvarianter jag önskar, alltså av den typ jag tidigare diskuterade.

Sedan har vi problemet med vänstertrunkering, alltså att man anger att det ord man söker på också får ingå som slutled i en sammansättning; ex: MYNDIGHETSREGEL som träff på REGEL. Denna typ av trunkering kan tekniskt sett vara mycket besvärlig att effektuera, beroende på att sökningen vanligtvis organiseras efter ordinitiala sekvenser av tecken, antingen via sk hashtabeller eller via successiva binära träd, bokstav för bokstav eller något liknande. (Det är sådana tekniker som möjliggjort den höga effektiviteten hos de booleska söksystemen.) Ett sätt att åstadkomma fri vänstertrunkering är att låta sökningen gå på rotlemmatiserade former (jfr ovan) av orden i den löpande texten. Ordet MYNDIGHETSREGEL skulle då få ett internt utseende ungefär som MYNDIG>HET´S-REGEL, där

bindestrecket skulle markera sammansättningsgräns. Vänstertrunkeringsuttryck av typ $REGEL skulle då uppfattas som att man sökte normalt men nu i den analyserade strukturen (obs REGEL blir med lämplig tolkning av bindestrecket ett självständigt ord i exemplet ovan). Om man ville ha ordet ovan också i sina böjda former skulle man alltså ange något i stil med $REGEL=.

Det finns också andra sätt att komma tillrätta med problematiken kring vänstertrunkeringen, t ex att man som alternativ till att organisera sökningen efter ordinitiala teckensekvenser gör det hela med utgångspunkt från ordfinala sekvenser. Dock måste man även här ha möjlighet att bortse från böjningsändelserna men exakt hur det skulle kunna gå till är inte alldeles klart. Det är dock klart att detta skall kunna gå att åstadkomma med en kolossalt mycket enklare morfologisk komponent än den som förutsätts i ett system med fullständig morfologisk analys, bl a bör man klara sig nästan helt utan stamlexikon.

6. "Gimme more o´ that!"

Det sista delprojektet jag här skall ta upp är kanske inte i den nu pågående fasen så där direkt tillämpad lingvistik, och därför tänker jag inte behandla det så utförligt. På sikt kommer det dock att bli så i högsta grad, ty det förutsätter att man faktiskt löst problemet med att definiera (semantiskt) "avstånd" mellan texter (jfr avsn 3, ovan). I Brodda 1970 införde jag en mycket enkel metrik för att mäta avståndet mellan deskriptorvektorer (ungefär "den procentuella överlappningen"), som där närmast användes för att mäta avståndet mellan sökfråga och dokumentindex. Ingenting hindrar emellertid att samma mått används för att mäta avståndet mellan (innehållet) i olika textavsnitt. I det aktuella projektet avser jag att tillämpa det hela på paragraf- eller styckenivå (paragrafer och/eller stycken är de naturliga grundenheterna i t ex lagar).

Det problem som diskuterades i Brodda 1970 tog sin utgångspunkt från följande enkla observation: antag att vi har ett sökuttryck med två sökord, A och B. Om vi kopplar ihop dem med OCH-operatorn innebär det att vi enbart betraktar sådana dokument som träff där både A och B ingår i dokumentet (eller rättare sagt dess index, dvs den sökbara delen av dokumentet). Om vi i stället väljer ELLER-

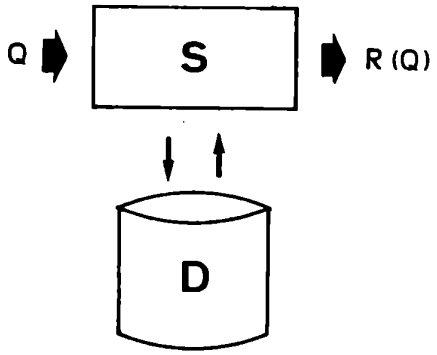
operatorn innebär det att det "räcker" med att endera A eller B ingår (och för all del, även bägge). Antag att vi nu i stället tillämpar följande metod att beräkna om träff föreligger, nämligen att vi åsätter vardera av orden i sökuttrycket ett poängtal, t ex poängtalet 1, och med utgångspunkt från dessa poängtal tilldelar vi sedan varje dokument i sin tur ett poängtal, nämligen totalsumman av de poäng för ord som vi återfinner i dokumentet, och sedan anger vi separat en tröskel, ovanför vilken vi kräver att ett dokument skall hamna för att träff skall föreligga. (Sådana sökmekanismer finns implementerade i många söksystem.) Låt oss nu se hur träffmängden varierar med olika val av tröskel. Sätter vi tröskeln = 1 räcker det uppenbarligen med att endast ett av orden skall ingå för att träff skall föreligga, medan om vi kräver totalpoängen 2 måste uppenbarligen båda orden vara närvarande. Vi ser alltså att tröskeln satt till 1 är ekvivalent med ett ELLER-villkor, medan tröskeln satt till 2 ger något som är ekvivalent med ett OCH-villkor.

Problemet jag diskuterar i ovan nämnda artikel är det generella hur booleska sökförfaranden förhåller sig till "poäng"-förfaranden, och det jag visar är kort följande: Varje poängförfarande är ekvivalent med ett booleskt sökuttryck. Omvänt kan däremot ett givet booleskt uttryck bara approximeras med ett pänguttryck, men lustigt nog så, att approximationen normalt faktiskt blir "bättre" (naturligare) än det ursprungliga booleska uttrycket. För att åstadkomma denna approximation utnyttjas en speciell teknik att väga samman deskriptorvektorer, lämpligen kallad (boolesk) faltning (hur den utföres är för komplicerat att här gå in på), och det är denna sammanvägningsteknik som jag avser utnyttja i det delprojekt som rubriken till detta avsnitt syftar på.

Bakgrunden är följande: Antag att man under en söksession "råkat" finna några dokument (paragrafer, stycken) som visar sig vara relevanta. (Man kan t ex ha startat sessionen med en sk "screening" - "browsing", översiktssökning. Man kan ju också ha fått tips av en kollega, mm.) En mycket naturlig frågeställning är nu om man inte borde kunna få söksystemet att med utgångspunkt från de funna dokumenten självt plocka fram alla liknande. Kort sagt, det man efterlyser är existensen av ett funktionskommando (kanske till och med en funktionstangent) just med innebörden "Gimme more o´ that!". (Jfr också Seipel 1976, där just frågeställningen hur

man skall kunna finna "liknande" lagparagrafer diskuteras.)

Matematiskt kan det nya i denna frågeställning formuleras i följande termer. Antag att vi har ett söksystem av befintlig typ givet, omfattande ett sökförfarande S och en databas D. För en given sökfråga Q ("Query") räknar så söksystemet ut "svaret", responsmängden R(Q) som (hänvisningar till) de dokument som enligt sökförfarandet S uppfyllde sökfrågan Q. Vi har alltså en situation som



Figuren ovan representerar den normala söksituationen. (Denna är dock inte så enkelriktad som figuren ovan låter antyda. Man har ju mycket starka feed back mekanismer, genom att man i en given söksituation vanligtvis börjar med en relativt vag formulering av sitt sökuttryck, och beroende på det svar man erhåller modifierar man successivt sökuttrycket tills responsmängden svarar mot ungefär det man tycker sig vilja ha.)

I "Gimme-more-o'-that"-situationen är frågeställningen den omvända. Där har man en hygglig mängd RO given, och det man frågar efter är om det finns ett sökuttryck Q sådant att R(Q) dels omfattar RO och dels alla liknande dokument. R(Q) skall alltså vara en i någon mening optimalt utvald supermängd till den givna mängden RO.

Nu visar det sig att detta problem rent matematiskt är väldigt underbestämt. Rent teoretiskt kan det finnas ett mycket stort antal supermängder till RO som kunde vara pretendenter på att vara den eftersökta supermängden och samtidigt svarande mot någon sökfråga. För att få någon ordning på det hela måste man då lägga till ytterligare kriterier på hur denna eftersökta sökfråga skall konstrueras. Det ena och mycket nödvändiga kriteriet är att man skall kunna bestämma Q på ett effektivt sätt (dvs Q skall gå att bestämma med en enkel algoritm) ur dokumenten ingående i RO. Ett andra kriteriet är att den erhållna supermängden R(Q) skall vara naturlig

och "bra". Nu visar det sig att den ovan omnämnda faltningsoperationen utförd på deskriptorvektorerna till de dokument som ingår i RO tycks uppfylla dessa villkor. Det första villkoret är mycket enkelt uppfyllt, och en del simuleringsexperiment jag utfört under senare tid tyder också på att de genererade sökfrågorna ger mycket naturliga responsmängder som resultat. Storskaliga "riktiga" test för att utröna det senare planeras så snart vi funnit något lämpligt söksystem som erbjuder en tillräckligt experimentbetonad miljö.

Referenser

Brodda, B. "Document Retrieval - a Topological Problem", SMIL 1970.

Brodda, B. "Något om de svenska orden fonotax och morfotax", i Papers from the Institute of Linguistics, Stockholm University, PILUS no. 38, Stockholm 1979.

Brodda, B. "Yttre kriterier för igenkänning av sammansättningar" i Förhandlingar rörande svenskans beskrivning, nr 13, Helsingfors 1982.

Brodda, B. "An Experiment with Heuristic Parsing", i Papers from the 7th Scand Conf of linguists, Helsinki University 1983.

Brodda, B. & Karlgren, H. "Informationssökningsmetodik", Rapport nr 1 till Kgl Statskontoret ang informationssökningsmetodik. Skriptor, Stockholm 1965.

Hornstrand, Ch."Något om de syntaktiska ytmönstren i svenskan", PILUS nr 50, 1983.

Källgren G. "Automatisk excerpering av substantiv ur löpande text", IRI-rapport 1984:1, Institutet för Rättsinformatik, Stockholms Universitet 1984.

RÄTTSDATA, utveckling, nuläge och framtid, Justitiedepartementet 1984 (ref: Ds Ju 1984:3).

Seipel, P. "Informationssystem för Rättsväsendet: Projekt Index", Stockholm 1976 (SARI).

Walker, D. & Karlgren, H. & Kay, M. (Editors): "Natural Language and Information Science - Perspecitvie and Direction for Research", FID, publ. 551, Stockholm 1973 (Skriptor).