DE NORDISKE DATALINGVISTIKKDAGENE 1981

Foredrag fra en konferanse
på Universitetssenteret på Dragvoll
22.-23. oktober 1981,

utgitt av Eirik Lien

EDB-tjenesten for humanistiske fag
Universitetet i Trondheim

INNHOLD:

Forord

Det tredje arrangementet i serien Nordiske datalingvistikk-dager holdes ved Universitetet i Trondheim 22. og 23. oktober 1981, med Edb-tjenesten for humanistiske fag som arrangør. Initiativet til konferansen er - som tidligere - tatt av Den nordiske samarbeidsgruppen for datamaskinell språkbehandling. Konferansen har vært planlagt av styret for edb-tjenesten, og den praktiske gjennomføringa har vært gjort av Kirsten Strømmen og Eirik Lien.

Til konferansen er det påmeldt ca. 80 deltakere fra Danmark, Sverige og Norge. Deltakerliste pr. 7. oktober er tatt inn bakerst i denne publikasjonen.

Denne publikasjonen inneholder den skriftlige versjonen av alle foredragene som blir presentert på konferansen, og de er trykt i den rekkefølgen de blir presentert. Foredragsholderne er bedt om å gjøre dokumentasjonen fyldig slik at det kan settes av desto mer tid til diskusjon på selve konferansen. En del foredragsholdere vil derfor forutsette at de viktigste synspunktene er kjent når de presenterer stoffet sitt.

Det fjerde arrangementet i denne serien skal holdes høsten 1983, og det vil bli bestemt under møtet i Trondheim hvor det skal arrangeres og hvem som skal være ansvarlig for det.

Trondheim 7. oktober 1981

Eirik Lien

Per Vestbøstad
NORSK TEKSTARKIV
UNIVERSITETET I BERGEN

NORSK TEKSTARKIV ETTER HALVTANNA ÅRS DRIFT

1. Litt historikk:

Tanken om eit arkiv for maskinlagra norsk tekst vann tilslutning på konferansen om "eit norsk datamaskinelt tekstkorpus" i Bergen, oktober 1978.[1] Etter konferansen vart det skipa ei plangruppe med representantar frå fagmiljøa i Bergen, Oslo og Trondheim og frå Norsk Språkråd. Arbeidet i denne gruppa førte fram til at Rådet for humanistisk forskning (i NAVF) gav løyving til ein prosjektmedarbeidar og driftsmidlar for 1980 med lovnad om løyvingar for åra 1981-1984. Prosjektet var då formalisert som eit samarbeid mellom PDS ved Nordisk inst. i Bergen og NAVFs edb-senter samme stad. Planleggingsgruppa vart omgjort til fagleg referansegruppe, under namnet Fagleg råd.
I planleggingsfasen arbeidde også ei gruppe med å avklara eit standardformat for tekstlagring.

2. Oppstartinga:

1. mars 1980 tok underskrivne til som prosjektarbeidar ved Norsk tekstarkiv, som fekk kontorplass ved NAVFs EDB-senter for humanistisk forskning ved Universitetet i Bergen. Når det gjeld datamaskinkraft, har Tekstarkivet fungert som eit prosjekt under PDS (Nordisk inst., UiB).

Den mest nærliggjande arbeidsoppgåva var å skaffa seg oversyn over kva for maskinlagre tekstar, som alt fanst i forskningsmiljøa landet over. Etter ei grundig rundspørjing (med god hjelp frå EDB-tenestene ved HF-fakulteta i Oslo og Trondheim) kunne oversynet spreiast i septembar 1980. Oversynet er sjølvsagt halde ajour og skal sendast ut på nytt denne hausten.

3. Inntak av avistekst

Som ei vidareføring av eit initiativ teke av Norsk leksikografisk inst. (Oslo) og PDS for å samla inn nynorsk avistekst, kontakta eg sommaren 1980 alle nynorskavisene i landet og laga eit oversyn over den satstekniske utrustninga deira. Ein del tekst vart samla inn, men avvikande diskettstandard i den grafiske industrien har til no hindra tekstinntak frå dei fleste mindre nynorskavisene, og frå mindre trykkeri elles. Ingen redaktørar hadde noko imot at Tekstarkivet fekk kopiera maskinlagra avistekst.

4. Inntak av bok-tekstar:

Den Norske Forleggerforening har rådd medlemmene sine til å gje Tekstarkivet høve til å kopiera maskinlagra tekstar frå bøker som dei har i produksjon, vederlagsfritt så lenge det gjeld ikkje-kommersiell bruk. Arbeidet med å få oversyn over kva bøker som finst maskinlagra, og kvar dei finst, har likevel vore komplisert.

Førebels har det vore relativt få bøker som har lagt godt til rette for å overføra til Univac-anlegget i Bergen. Vi har berre kunna teke inn frå større trykkeri, som kan levera magnetbandkopiar, men vi arbeider med å finne overføringsmåtar for dei vanlegaste fotosetjarane, som berre har diskettlager.

Velviljen hos trykkeria har vore stor. Når leveringstida likevel kan verta 3 månader, heng det truleg saman med at slike oppdrag fell utanom dei normale arbeidsrutinene. Prisen på slike kopiar ser ut til å kunna variera frå kr 300,- til kr 1.000,- pr. bok!

Teksten som kjem frå sats-systemet, er full av typogra-fiske kodar. Ein del av desse skal følgja teksten vidare som spesialtegn eller format, medan andre berre kan lesast forbi. Ved hjelp av programsystemet CROMWELL (utvikla ved PDS) styrer vi omsetjing og utval av trykkerikodar. Det er sjølvsagt tidkrevjande å spesifisera alle kodealternativ i eit sats-system, men har ein først gjort det, er det lett å ta imot fleire tekstar frå same systemet.

I samarbeid med Bjørn Eide, PDS, har eg gjort ferdig spesifikasjonane til Comtecs CRT-13 og Bergens Tidendes sats-system, og arbeidet med Nortext-systemet er godt i gang.

Sjølv om vi etter kvart vil kunne standardisera tekstar frå enno fleire sats-system, vil Tekstarkivets eine konsulent ha av-grensa kapasitet og fleire oppgåver. Vi reknar difor med å lagra "rått" mykje av det vi får tak i, og først når det vert konkret spørsmål etter ein slik tekst, vil vi "foredla" han til standardformat.

5. Tekstproduksjon i eigen regi

For å auka tilgangen på skjønnlitterære tekstar, har Tekstarkivet i eit samarbeid med Norsk språkråd gått i gang med å registrera 10 bokmålsromanar frå 1977, som Språkrådet skal nytta til å studera påverknaden frå offisiell norsk skriftnorm. Det er meininga å registrera like mange nynorskromanar frå same året, og dessutan ønskjer ein å undersøkja romanar utkomne i 1937 og 1917 på liknande måte.

6. Tekstformidling

Inntak, standardisering og produksjon av maskinlagra tekst har såleis stått sentralt i Tekstarkivets arbeid denne perioden. Etter kvart som tekstmengda veks, og Tekstarkivets tenester vert vidare kjente, reknar eg med at formidlingsarbeidet vil krevja meir tid. Til no har det særlig vore spørsmål etter det som finst av ferdige konkordansar. Dette bør få oss til å vurdera om ikkje Tekstarkivet også bør foredla nokre sentrale tekstutval til konkordans- og listeprodukt, slik ein t.d. ved PDS har gjort med bokmålsaviser frå 1968-73.

7. Oppsummering

Norsk tekstarkiv har ikkje til no makta å skaffa fram større mengder ny maskinlagra tekst, men ein har gjort viktig grunnlagsarbeid, som truleg har gjort det monaleg lettare å finna ut kva som finst, og å få tak i det som finst av slik tekst. Det er også oppretta kontaktar med forlag og trykkeri, som vil gje oss innsyn i den årlege produksjonen på eit tidspunkt då dei fleste bøkene enno ligg maskinlagra i trykkeriet. Ein har også gjort seg godt kjent med den satstekniske utrustninga hjå små og store trykkeri. Komande arbeidsår vil dette nokså sikkert gje utbytte i form av ein større auke i tilgangen på moderne norsk tekst for datamaskinell bruk i forsknings- og utviklingsarbeid.

Roald Skarsten
EDB-seksjonen v/HF
Univ. i Bergen

PRESENTASJON OG KOMMENTARER TIL NORSK UTGAVE AV C. MULLERS BOK "LINGVISTISK STATISTIKK"

Det er en glede å kunne presentere for deltakerne på de nordiske datalingvistikkdagene den norske utgaven av C. Mullers bok: Initiation aux méthodes de LA STATISTIQUE LINGUISTIQUE, Paris 1973. Den norske utgaven er et resultat av samarbeid mellom flere parter. Den er tilrettelagt ved NAVF's EDB-senter for humanistisk forskning i samarbeid med EDB-tjenestene for humanistiske fag ved universitetene i Oslo, Bergen og Trondheim. NAVF's EDB-senter for humanistisk forskning har i hovedsak dekket utgiftene. Undertegnede har organisert og ledet arbeidet med tilrettelegging og utgivelse av boken.

Boken er oversatt fra fransk av cand. philol. Kari Fonnes med assistanse fra amanuensis Ivar Fonnes som har hatt ansvaret for statistisk terminologi. Dessuten har universitetslektor Elsa Quale vært statistisk konsulent. Med utgangspunkt i E. Quales konsulentuttalelse har Ivar Fonnes og jeg bearbeidet oversettelsen på enkelte punkter, og noen merknader er satt inn i noter som "oversetters anmerkning" (o.a.). Konsulent Eirik Lien har lest igjennom det endelige manuskript og gitt språklige kommentarer.

Vi kan nesten si at den norske boken faktisk er bedre enn den franske originalen, fordi den er bearbeidet og utstyrt med noen oppklarende noter og fordi en del småfeil er rettet opp. Den fagstatistiske konsulenten hadde lyst til å rette opp mere enn det som er skjedd, men vi har måttet balansere mellom de ideale krav og den beskyttelse som åndsverkloven gir bokforfatteren.

Jeg antar at de av deltakerne som ikke har noen matematisk bakgrunn vil glede seg med oss over boken fordi den vil dekke et behov som tidligere har vært vanskelig å få tilfredsstilt for oss. Forfatterens intensjoner, slik han formulerer de i sitt forord, viser dette: Framstillingen er begrenset til å presentere og forklare både prinsipielt og praktisk hvordan statistiske metoder kan brukes på språklige og stilistiske fenomener. Hvilke spørsmål vi kan stille til lingvistisk statistikk og svar som statistikken kan gi er hele tiden en del av framstillingen. Slik gir boken elementære statistiske kunnskaper som kan brukes i praksis, og venner leseren til å bruke statistiske resonnementer og et algebraisk språk når det er snakk om å underbygge slike resonnementer.

At et slikt udekket behov for en slik elementær innføring i bruk av statistiske metoder innen språk og litteratur ikke er noe spesielt norsk fenomen ser vi ved at boken også er oversatt til tysk. I 1971 oversatte Lothar Hoffmann en tidligere versjon (fra 1968): Initiation á la statistique linquistique. I sin anmeldelse av denne tidligere versjonen skrev Hoffmann bl.a.: "Mullers Darstellung des "Begriffs - und Methodenarsenals der Statistik entspricht dem neuesten Stand. Sie ist vorläufig der einzige systematische Versuch dieser Art auf dem Gebiet der Linguistik. Sie besticht durch Einfachheit und Klarheit.""

Da vi så søkte etter en bok som kunne dekke behovet var der i tillegg til Muller kommet en bok med et lignende formål fra Barron Brainerd, Weighing Evidence in Language and Literature (Toronto 1974), men den hadde ikke de pedagogiske fortrinn som Muller's bok har. Konferer Bruce A. Beatie i artikkelen, "Measurement and the Study of Literature", Computers and the Humanities, july/sept. 1979, "The Communication problem is even more obvious in an effort that purports to bridge the gap: Barron Brainerd's Weighing Evidence in Language and Literature.[8] Although the reviewer in Computers and the Humanities assured us that it was "clearly written", the literary colleagues to whom I showed it found Brainerd's book as incomprehensible as a textbook on tensor calculus or thin-layer chromatography." (s. 191)

Der er etterhvert en ganske omfattende forskningslitteratur på feltet, men denne har vært vanskelig tilgjengelig uten de nødvendige forkunnskaper, og nettopp dette angir Hoffmann som et argument for sin oversettelse. "Besonders mangelt es an einer verständlichen und systematischen Einführung in die Sprachstatistik, die der Zugang zu der im Ausland Zahlreich publizierten Forschungsergebnissen erschliesst. Die übersetzung des Buches von Ch. Muller soll diese Lücke schliessen helfen." (s. 5 i forordet)

Det kan forøvrig nevnes at vi tok kontakt med både Sture Allén og H. Spang-Hanssen som begge støttet tanken om en oversettelse av Mullers bok. Når vi allikevel ikke alltid har vært like sikker på riktigheten av det vi gjorde, så var det fordi den statistiske konsulent vi brukte ikke umiddelbart var begeistret for boken, ut fra et fagstatistisk synspunkt. Jeg skal komme tilbake til hennes synspunkter i kommentaravsnittet. Som et praktisk problem, bokutgivelse eller ikke bokutgivelse, sto vi imidlertid i den situasjon som karakteriseres fortrinnlig ved det engelske ordtaket: "Beggers can't be choosers."

Noen lurer kanskje på hvorfor jeg kommer inn på disse forholdene. Boken foreligger der, og ferdig med det! En god grunn er at vi ønsker å ha en mest mulig positiv bakgrunn for de faglige kritiske synspunkter som vi skal komme tilbake til, og en annen god grunn for å presentere boken er at vi ønsker salg på den!

Jeg iler tilmed å forsikre om at de forannevnte personer ikke har noen økonomisk gevinst av salget. Et eventuelt tap på boken bæres av Universitetsforlaget. Hvis der imidlertid blir et visst salg på boken, så har vi en mulighet til å få utgitt fortsettelsesboken: Principes et méhodes de statistique lexicale. (Paris 1977) I den første versjonen fra 1968 besto boken av to deler som hver svarer til disse to nye bøkene, bortsett fra at de sistnevnte er blitt betraktelig utvidet og forbedret. I en anmeldelse av den siste boken skriver Michel Dubrocard om den første versjonen fra 1968, "The work was soon out of print, which was a measure of its success."

Om den boken som danner grunnlaget for den norske oversettelsen skriver han forøvrig at den er "considerably improved," og om fortsettelsesboken: "this new book is not merely a serviceable introductory manual indispensible for all newcomers in the field of statistical linguisties. It is also required reading for specialists and advanced scholars,",Computers and the Humanities,January - March 1979, s. 84.

Det er også verdt å nevne at det er utgitt et eget øvelseshefte til bøkene, som det også kunne være aktuelt å vurdere i oversettelsessammenheng, etter som statistikerne stadig vekk fremhever at praktiske statistiske øvelser er et nødvendig ledd i læringsprosessen.

Boken og forfatteren skulle hermed være presentert og anbefalt på det sterkeste. Det skal bare nevnes at den har vært benyttet som kursbok i et nasjonalt sommerkurs, og at boken i Norge allerede har vist sin fortreffelighet. (Hvis dere finner forbausende få trykkfeil i boken så skyldes det bl.a. kursdeltakernes innsats). I tiltro til at Norge er et representativt utvalg som gir grunnlag for en positiv slutning om bokens skjebne i den nordiske populasjon våger jeg å gå over fra presentasjonen til kommentarene.

Den alvorligste innvendingen mot Mullers bok fra den fagstatistiske konsulenten gjaldt presentasjonen av hypotesetesting. Hun skrev bl.a.: "Det jeg kritiserer hos Muller er hans slurv og nonchalanse i den manglende formulering og presisering av dette utgangspunktet for flere av signifikanstestene.

Først i avsnitt 13.8 kommer han inn på modellbegrepene språkytring. Da har han allerede gjennomgått flere eksempler på signifikanstesting og brukt sannsynlighets-teoretiske begreper som stokastisk variabel, sannsynlighetsfordeling, forventning etc. etc. - uten denne rammen som skulle gi begrepene mening.

I avsnitt 16.3 kommer han igjen inn på denne tankegangen som, så vidt jeg kan se, danner grunnlaget for og gir mening til det meste av analysene i de siste to tredjedelene av boka (kap. 9 og utover)."

Først må jeg her si at Muller allerede i kap. 3.8 med overskriften SPRÅK OG YTRING kommer inn på dette fundamentale forhold i forbindelse med det generelle spørsmål om POPULASJON OG UTVALG som er hovedoverskriften for kap. 3. "På den annen side kan vi introdusere den klassiske distinksjon mellom språk og ytring, altså mellom den potensielle og den realiserte. Da må vi betrakte enhver ytring som en realisering, altså som et utvalg av språket til den talende eller skrivende. For å observere språk må vi gå omveien om ytringene. Enhver statistikk baserer seg nødvendigvis på tekster, dvs. på utvalg av språket. Derav følger at hver gang vi mener å trekke en konklusjon om språk ut fra en statistisk analyse, så resonnerer vi ved induksjon; vi gjør oss opp en mening ut fra et utvalg. Vi må derfor benytte regneoperasjoner som er utviklet for slike analyser."

Dette er nå allikevel bare en detalj, hovedpoenget for E. Quale er at Muller ikke alltid klarer å fastholde sine i utgangspunktet riktige formuleringer når det kommer til konkrete eksempler i boken, og hun ville ha gitt dette forhold en bredere omtale i boken. Til nye lesere vil jeg derfor understreke nødvendigheten av å være særdeles oppmerksom på kap. 3.8, kap. 13.8 og kap. 16. Jeg synes Mullers synspunkter her er så viktige for forståelsen av boken og for å sette kritikken i sitt rette perspektiv at jeg vil sitere litt grundig fra kap. 16. "Vi går ut fra at moderpopulasjonen ikke består av selve teksten, men av språket, i Saussures betydning av ordet; et "språk" der egenskaper estimeres ut fra det store utvalg som stykket utgjør. Det er altså ikke det franske språk, heller ikke århundrets språk eller det språk som ble brukt i komedier på vers på den tiden, heller ikke det språk som ble brukt i Pierre Corneilles komedier på vers. Det er en bestemt språklig tilstand, nemlig Corneilles mens han skrev L'Illusjon; en latent mulighet som ikke bare er bestemt av språkets egne lover, men også av forfatterens person, av den genre han brukte (stilistiske årsker) og av det emne han hadde valgt (tematiske årsaker); et "språk" hvorav stykket og utdraget bare er utvalg av forskjellig størrelse.

Og språket, eller den språklige tilstand, kan pr. definisjon ikke observeres umiddelbart."

Det som så skjer i en del eksempler er at Muller, når han formulerer nullhypotesen, knytter denne til utvalget (teksten) istedetfor til den bakenforliggende populasjon (modell). Hypotesetestingens hensikt eller funksjon er jo å undersøke om forskjellen mellom f.eks. to gjennomsnitt i to utvalg er stor nok til at vi kan konkludere med at de er utvalg fra forskjellige populasjoner, og at nullhypotesen om at de er fra samme populasjon kan forkastes. Et eksempel fra Mullers bok kan illustrere Quales påstand. "Le cid har 4,01% adjektiver, Phédre har 5,99%. L'Illusion comique har 18% substantiver mens Matamore's rolle i stykket har 20%. I begge tilfeller må man gå ut fra nullhypotesen: "de to tragediene er utvalg av samme populasjon hvor proporsjonen av adjektiver er stabil. Forskjellen kommer av tilfeldige variasjoner" - "proporsjonen av substantiver i stykket er stabil. Det avvik som observeres i rollen bygger på tilfeldige variasjoner i de utvalg som er trukket fra populasjonen". (s. 108) Dette eksemplet viser hvordan det i det første tilfellet er brukt en riktig formulering av nullhypotesen. "De to tragediene er utvalg av samme populasjon...", mens det i det andre tilfellet "proporsjonen av substantiver i stykket er stabil...". Her er altså nullhypotesen knyttet til utvalget (til stykket) og da blir det meningsløst med hypotesetesting. Det er nærliggende å tale om slurv i dette tilfellet, men der er flere slike tilfeller, f.eks. i kap. 14.6 hvor nullhypotesen igjen formuleres i tilknytning til det aktuelle utvalg av et gitt språklig fenomen."(Den er holdbar hvis vi kan forkaste nullhypotesen:"tilfeldig fordeling av (a) i hele teksten.")", men det er ikke bare formuleringen av nullhypotesen dette gjelder, det kan også gjelde konklusjonene som trekkes, de knyttes til utvalget istedetfor til populasjonen, dvs. "språket". Et annet eksempel kan man bl.a. finne i kap. 18, med følgende konklusjon: "...det stilistiske fenomen (spesielt mange substantiver i dette utdraget) er reelt." (s. 142) Et annet eksempel finnes i kap. 21.4. Generelt savnes ofte modellformuleringen, leseren må selv ha den i tankene.

E. Quale finner at den sannsynlighetsteoretiske grunntanken blir uklar fordi forbindelsen mellom modell og virkelighet fremstilles for mangelfullt og til dels uriktig, som vi har sett. Det eneste vi kan gjøre med dette er å henstille til leserne å sette de uheldige formulerte eksempler inn i sin rette sammenheng, slik den er presentert i de nevnte Viktige avsnitt om språk og ytring. Når disse kritiske kommentarer fra fagkonsulenten er nevnt vil jeg også ta med noen av hennes positive kommentarer. De første kapitlene karakteriserer hun som preget av pedagogisk nennsomhet og omhu, og anbefales på det varmeste, og den deskriptive statistikken får god omtale . Ved fornyet lesning av den induktive statistikken (i lys av språkytring-modellen) hadde hun fått et mere positivt syn på denne delen. Hennes kommentar tildet oppsummerende avslutningskapittel består av bare ett ord: Nydelig.

"Det er respektabelt og oppløftende at en lingvist har påtatt seg det krevende arbeide å forsøke å formulere og introdusere det statistiske begrepsapparat blant språkforskere, hvis forhold til tall og formler man må formode ikke er av det mest fortrolige slaget." Hun synes "Det er viktig at boka er skrevet, at den er oversatt til norsk, og jeg synes den bør utgis." Hun ville dog ha foretrukket at de problemer som er påpekt i den induktive delen kunne vært kommentert i et tillegg, slik at studentene venner seg til presise oppstillinger av nullhypotese og alternativ hypotese i tilknytning til den aktuelle modell (populasjon).

Mullers bok skulle hermed være presentert og kommentert med sine pro et contra på behørig måte, ut fra dens egne forutsetninger.

Rolf Gavare
SPRÅKDATA

LEXIKOGRAFISK ALFABETISERING

Inledning

Föredraget behandlar de principfrågor som aktualiseras vid krävande datamaskinell alfabetisering av lexikaliska och encyklopediska material och ger förslag för den praktiska implementeringen. Även om särskild hänsyn tas till svenska språkets konventioner, så har grundprinciperna relevans även för andra språk med alfabetisk skrift.

Vad menar vi då med alfabetisering? Frågan kan i förstone tyckas överflödig. När man kommer in på detaljfrågor visar det sig emellertid ofta att man inte har någon etablerad praxis eller inte är medveten om vilka kriterier man använder för det alfabetiska ordnandet.

Med ett alfabet brukar vi mena ett språks uppsättning av bokstäver med en viss hävdvunnen intern ordningsföljd. Dessa konventioner är språkberoende. Alfabetisering är alltså, lite vagt formulerat, ordnandet av ord eller ordförbindelser i alfabetisk följd. Problemet är nu bland annat att ord inte enbart består av alfabetiska skrivtecken utan även av siffror, interpunktionstecken, diverse symboler, främmande skrivtecken, etc., utan självklar plats i alfabetet. Dessutom tillmäts inte alla skrivtecken lika stor betydelse — somliga tecken påverkar inplaceringen i den alfabetiska ordningsföljden mindre än andra.

Alfabetiskt ordnade register behöver ofta hanteras av datorer, vilket ställer särskilda krav i fråga om konsekvens, entydighet och formaliserbarhet på principerna för det alfabetiska ordnandet. De standardprogramvaror som finns för alfabetisk filering uppvisar emellertid stora brister; ofta blir det till och med problem att jämställa versaler och gemena skrivtecken eller v och w eller att få den svenska bokstavsordningen ...zåäö. De diakritiska tecknen och junkturerna klaras inte alls av på något tillfredsställande sätt. För en del användningsområden kan man acceptera vissa ofullkomligheter i alfabetiseringen, men för tillämpningar med större krav på lexikologisk stringens (såsom olika slag av konkordanser, ordlistor, ordböcker, encyklopedier, personnamnsregister och bibliografiska kataloger) krävs formaliserade alfabetiseringsprinciper som motsvarar vetenskapliga krav.

Alfabetiseringsreglerna skall uppfylla den grundläggande förutsättningen att kunna ge en algoritmisk beskrivning som

kan implementeras i datorprogram. Tyvärr saknas ännu en tillräckligt specifik standard för lexikografisk alfabetisering. Denna översikt kan förhoppningsvis bidra med synpunkter inför ett sådant standardiseringsarbete.

Allmänna överväganden

Ett fundamentalt krav på principer för alfabetisering är att slumpmässig ordning mellan (grafiskt) olika enheter inte skall tolereras. Ordningsföljden skall alltid vara förutsägbar och entydig. Detta innebär att man skall få exakt samma resultat om samma material alfabetiseras på nytt. En delvis slumpmässig interfilering vållar i många sammanhang komplikationer.

Då de traditionella alfabetiseringsprinciperna är otillräckliga och man behöver generalisera och utvidga dem, bör man sträva efter att finna lösningar som har lingvistisk och psykologisk relevans.

Rent allmänt kan sägas att alfabetiseringsregler först skall ge besked i två förberedande steg: vilka element som alfabetiseringen skall ta hänsyn till och vilka teckenföljder som skall jämföras då ordningen mellan enheterna skall fastställas. Man skall sträva efter att åstadkomma en uttömmande och särskiljande uppsättning termer för sorteringen.

I det första steget gäller det således att etablera de s.k. rangord som skall vara sorteringsgrundande. Olika tillämpningsområden har mycket olika konventioner för vad som skall utgöra rangord. Som ett exempel kan nämnas den bibliografiska fileringen, där bland annat vissa prepositioner och konjunktioner samt bestämda och obestämda artiklar (oavsett språk) inte får användas som första rangord; vissa artiklar kan dock vara homografa med exempelvis pronomen eller räkneord, vilka kan bli rangord. Ur den stora mängden bibliografiska regler kan också nämnas att tecknet & skall alfabetiseras som och, og, and, und, et, osv. alltefter titelns språk och att siffertal ordnas i stigande numerisk ordning. (Det förekommer också att man inordnar siffertalen enligt en alfabetisk form på det aktuella språket.)

Ett annat viktigt tillämpningsområde rör namnregister över personer, företag, etc. Här gäller ofta på likartat sätt att vissa led i personnamn negligeras (af, van, van der, von, etc.) och man använder inte attribut av typen Allmänna, Rikets, Statens, Svenska, Sveriges, osv. som första rangord.

I den lexikografiska alfabetiseringen är normalt alla ord som ingår i en ordboksartikels uppslagsform (och ev. ordklassbeteckning) rangord. (Underordnade uppslagsord eller -fraser samt hänvisningar kan dock vara primärt inplacerade med huvudordet som rangord i stället för förstaordet.) Eftersom vi här koncentrerar oss på de lexikografiska tillämpningarna, så behöver vi inte närmare gå in på hur rangorden etableras.

I det andra förberedande steget vid alfabetiseringen måste vi alltså bestämma i detalj vilka tecken som jämförelsen skall grunda sig på. Annorlunda uttryckt gäller det att etablera en ordningsform ur de rangord man tidigare valt ut. Detta led i alfabetiseringsproceduren har direkt koppling till det implementeringstekniska, vilket gäller genereringen av de sorteringsnycklar som datorn behöver för att styra ordnandet.

Ordningsformen etableras i lexikografiska sammanhang i allmänhet enligt teckenprincipen (stavningsprincipen). Denna innebär att enheternas inplacering i den alfabetiska ordningen helt grundar sig på stavningen, inte uttal, betydelse, vanligaste stavningsform eller något annat.

Andra principer för etablerandet av rangordens ordningsform, som t.ex. efter ordens fonetiska form eller normalform (kanonisk form), kräver i allmänhet att varje element manuellt förses med en lämplig ordningsform. Häri ligger också en stor risk för osäkra eller inkonsekventa bedömningar. Personnamnsregister är ofta alfabetiserade enligt normalformsprincipen (t.ex. telefonkatalogen) eller fonetisk form (exempelvis variantregister över släktnamn). Stavningsprincipen kommer här att ligga till grund för de fortsatta resonemangen.

[Beroende på materialets art, omfattning och presentationsform (internt arbetsmaterial, ordböcker för publicering, databaser, etc.) kan det finnas anledning att göra andra bedömningar än dem som fastslås i de följande avsnitten. Avsikten har emellertid i första hand varit att komma fram till ett så konkret förslag som möjligt vad gäller behandlingen av stora lexikaliska material.]

Ordvis eller teckenvis jämförelse?

Grundprincipen för sorteringsgången är att jämförelsen sker ord-för-ord eller tecken-för-tecken och vid likhet upprepar man samma procedur för nästa element osv. När två enheter skiljer sig från varandra endast genom att det ena har extra tecken i slutet anser vi den kortaste komma först — "ingenting före någonting".

Skall man då alfabetisera teckenvis eller ordvis? Eller, annorlunda uttryckt: I vilken mån skall man ta hänsyn till ordgränserna då man alfabetiserar? Här går det inte att ge något självklart och allmängiltigt svar — det beror på materialets art och, inte minst, på det aktuella språket. Rent allmänt kan sägas att om språkbruket vacklar mellan särskrivning och sammanskrivning (med eller utan divis) så är tecken-för-tecken-metoden (där alltså ordmellanrummen är av underordnad betydelse) oftast att föredra därför att man då får samlat de förekommande varianterna på ett ställe; då man inte är säker på om ett uttryck är sär- eller hopskrivet slipper man alltså att söka på flera ställen. I engelskspråkiga register tillämpas därför i de flesta fall teckenvis alfabetisering.

För svenskans vidkommande torde i allmänhet ord-för-ord-principen ge bäst resultat. Flerordiga uttryck behandlas då inte som hopskrivna utan förs in direkt efter det första ordet som simplex och ordnas sinsemellan i andra hand på följande ord.

ord-för-ord	tecken-för-tecken
god	god
god frejd	godartad
god man	god frejd
god tro	godhet
godartad	godkänna
godhet	god man
godkänna	godmodig
godmodig	gods
gods	godtaga
godtaga	god tro

Ett viktigt problem i samband med sorteringsgången gäller frågan om på vilket stadium man skall ta hänsyn till accentuerade tecken, skiljetecken, osv. Vid alfabetiseringen sker, som vi strax skall få se, ordnandet i flera steg där man successivt tar hänsyn till bland annat variantformer av bokstäver, diakritiska tecken, versaler och junkturer. Vid den teckenvisa alfabetiseringen kan dessa modifikatorer emellertid endast beaktas då hela ordningsformen i övrigt är identisk med en annan och inte på ordnivån, vilket är mer naturligt från lingvistisk synpunkt. Även detta faktum talar alltså för den ordvisa gången.

När alfabetiseringen sker ordvis är det således viktigt var ordseparatorer förekommer i de element som skall ordnas. Ett fall där bruket i svenskan kan vackla är abbreviationer och akronymer, vilka betraktas som flerordsuttryck om de skrivs med ordmellanrum: osv. respektive o. s. v. placeras därför in på olika platser, likaså SAS och S A S. (Problemen känner de flesta igen från telefonkatalogen.)

I vissa fall kan man vilja ge även andra tecken än ordmellanrum denna ordsepararande status, exempelvis logogram av typen & och – (s.k. till-minus; ev. även divis). I alfabetiska register över ämnens kemiska formler kan man göra en "ord-för-ord"-sortering där versalitet fungerar som separator (C CO CO_2 $CaCO_3$, ej C $CaCO_3$ CO CO_2).

Frågan om ordvis eller teckenvis alfabetisering måste alltså avgöras från fall till fall med hänsyn till språk och materialets art. För de fortsatta resonemangen (som i första hand gäller svenskan) utgår vi i allmänhet från att alfabetiseringen sker ordvis.

Vad som tidigare sagts beträffande ordningsgången i en alfabetisk sortering — att man jämför tecknen genom orden från ordbörjan till ordslut — gäller den normala, initialalfabetiska sorteringen. Ord som börjar på likartat sätt kommer då att stå intill varandra i den sorterade listan.

I en finalalfabetisk sortering jämförs orden i stället från ordslutet och åt vänster, så att alla ord med likartat slut (samma sista sammansättningsled, avledningssuffix, böjningsändelse, etc.) grupperas samman. För speciella ändamål (såsom korsordsordlistor och morfemordnade ordlistor) kan man naturligtvis införa speciella regler för i vilken ordning jämförelserna skall ske. För att kunna lösa problemen med behandlingen av diakritiska tecken, versaler, junkturer, osv. permuterar man helt enkelt tecknen i ordningsformen på önskat sätt. Sorteringsnyckeln genereras sedan på normalt vis och några andra åtgärder behövs inte.

Alfabetisk sortering av bokstäver

Vad som ligger till grund för en alfabetisk sortering är naturligtvis bokstävernas konventionella ordningsföljd i alfabetet. Såväl uppsättningen grundbokstäver som deras inbördes ordning varierar som bekant från språk till språk. Svenskan har aà* b c d eé f g h i j k l m n o p q r s t u vw x yü z å ä ö, danskan och norskan har a b c d eé f g h i j k l m n o p q r s t u v w x y z æ ø å, isländskan aá b dð eé f g h ii j k l m n oó p r s t uú v x yý z þ æ ö, engelskan a b c d e f g h i j k l m n o p q r s t u v w x y z, tyskan aä b c d e f g h i j k l m n oö p q r s t uü v w x y z, franskan aàâ b cç d eéèêë f g h iîï j k l m n oô p q r s t uùû v w x y z, spanskan aá b c ch d eé f g h ii j k l ll m n ñ oó p q r s t uúü v x y z, osv.

Om vi återgår till svenskan, kan vi notera att det förekommer bokstäver som egentligen bär diakritiska tecken — i, j, å, ä, ö — men som konventionellt betraktas som egna, självständiga bokstäver. När vi i fortsättningen talar om bokstäver med diakritiska tecken avses inte denna grupp av etablerade grundbokstäver i det traditionella alfabetet.

Vi kan vidare se att där förekommer ligaturer och digrafer av typen æ (da., no., isl.), ch och ll (sp.) och ij (holl.), vilka i respektive språk, av fonematiska skäl, betraktas som enkla bokstäver i alfabetiskt hänseende, men som av andra språk oftast betraktas som likvärdiga med de särskrivna bokstäverna: æ = ae etc. (Tyskt ß = sz behandlas emellertid som ss och franskans œ behandlas alltid som oe.)

I vilken mån skall då variantformer av bokstäver (i svenskan exempelvis w eller ü) och diakritiska tecken påverka den alfabetiska inplaceringen av ett ord? Skulle det vara acceptabelt att sätta w efter v och é efter e? I så fall skulle vi exempelvis få ordningen kilovara kilovis kilovolt kilovoltampere kilowatt kilowattimme respektive ide ideal ..., ideologi idé idéassociation ... idévärld idiom. Detta är inte tillfredsställande. Vid den alfabetiska sorteringen måste hänsyn primärt tas endast till grundbokstaven, utan hänsyn till om den i själva verket förekommer i en variantform eller är accentuerad. Först om de jämförda orden i övrigt är lika, skall man beakta bokstavsvarianter och diakritiska tecken.

---

*) Bokstäver som här står grupperade har samma primära sorteringsvärde.

Ett likartat problem gäller distinktionen mellan gemena och versala tecken. Skulle gemen alltid gå före versal (dvs. a A b B c C ...) så skulle t.ex. näve komma före Na och sten långt före Sten. På motsvarande sätt som nyss nämnts skall man därför alfabetisera primärt på en form där grundbokstaven inte ger någon distinktion mellan gemen och versal och först om orden i övrigt är lika bör man låta gemen, som språkligt sett får anses vara det omarkerade fallet, gå före versal.

Sammanfattningsvis kan alltså sägas beträffande alfabetisering av ord med enbart alfabetiska tecken att sorteringen i första hand skall grunda sig på en normaliserad form, där variantformer jämställs med grundbokstaven, diakritiska tecken negligeras och gemener och versaler betraktas som likvärdiga. Vid ordlikhet tas sedan hänsyn till variantformer, diakritiska tecken och versalitet, i nämnd ordning. Att det är lämpligt att testa i just denna ordning beror på att dessa distinktioner i de flesta fall specificerar en följd med avtagande semantisk differentieringsförmåga; den semantiska skillnaden mellan ordpar som sving / swing, tvist / twist, vatten / watten eller Myller / Müller är oftast större än den mellan exempelvis a / à, filen / filén, jubileum / jubiléum, supen / supén eller Rosen / Rosén, som i sin tur är större än skillnaden gemen—versal: göteborgskonstnär / Göteborgskonstnär, tv / TV eller Saab / SAAB (i synnerhet om även "oäkta" versaler finns med i materialet, t.ex. från ord som stått först i en mening eller först i en propriell syntagm). Det räcker således inte med en linjär ordningsföljd för alfabetets grundbokstäver, utan även de underordnade nivåerna av variantformer, diakritiska tecken, osv., måste beaktas och i sin tur få en intern rangordning.

Med tanke på att man i möjligaste mån vill gruppera ihop semantiskt närstående ord, är det viktigt att även jämförelsen med avseende på varianter, accenter och versaler sker i ordning från vänster till höger på ett sådant sätt att ord med störst likhet från ordbörjan räknat får stå tillsammans.

Behandlingen av abbreviationer och logogram

En speciell status bland orden har de olika typerna av förkortningar. De alfabetiska ordförkortningarna, abbreviationerna, är mycket ofta flertydiga (typen el., f., fr., min.). Vissa abbreviationer har också fått samma status som normala, självständiga ord och böjs som sådana: kolla, bil, stins, Saab, laser. Det är därför rimligast att abbreviationerna inordnas på samma sätt som övriga ord, dvs. enligt stavningsprincipen, oavsett vad förkortningarna härstammar från.

För logogrammen finns liknande skäl mot en inplacering grundad på en tänkt alfabetisk transkription. Men eftersom dessa tecken (av typen §, &, °, %, osv) inte har någon hävdvunnen plats i alfabetet får de ordnas i en särskild följd, lämpligen före alfabetet (så att de lätt uppmärksammas; ord som står

efter alfabetets slut glöms lätt bort). I undantagsfall kan logogrammen även ingå som led i hybrida grafordı 96%-ig, $-kursen, 273°K, 24x36.

Vissa logogram kan, som tidigare nämnts, anses vara ordseparerande, t.ex. &, — ("till"; om man vill betrakta detta tecken som logogram) och matematiska tecken.

Enstaka tecken från främmande alfabet kan också betraktas som logogram. Så exempelvis bruket av grekiska teoken i naturvetenskap: γ-strålning, β-karoten, Ω, μm. I speciella fall kan man överväga möjligheten att i stället sortera på en translittererad form. Vi berör detta i avsnittet om supplering av information för sortering.

Junkturgrafemens behandling

Till kategorin junkturer räknar vi i detta sammanhang alla tecken som inte är bokstäver, diakritiska tecken, logogram eller siffror. Specialtecken och symboler utan uttalsform räknar vi således hit. Förutom de vanliga skiljetecknen (med agglutinerande, abbrevierande eller syntaktisk funktion) kan nämnas tryckaccenter, segmentgränser, parenteser och citationstecken av olika slag.

Då man skall bedöma hur junkturerna skall behandlas i alfabetiseringen måste man bland annat ta hänsyn till att junkturerna normalt inte utläses, att bruket av t.ex. divis i sammansatta ord ofta är vacklande (all-rum / allrum, icke-religiös / ickereligiös, u-land / uland, Väst-tyskland / Västtyskland, etc.), likaså användningen av punkt i ordförkortningar (U.S.A. / USA, d.v.s. / dvs., kr. / kr) och kolon, apostrof och vissa andra skiljetecken vid förkortning, böjning och avledning (typen sta'n, så'n, Lars', '82, TV:n, SSU:are, H:son), för att nu ta några exempel.

Junkturerna bör därför påverka den alfabetiska sorteringsordningen endast då man på grundval av den rent alfabetiska delen av orden inte kan avgöra ordningen och man följer då en intern rangordning mellan junkturgrafemen. Också i detta fall bör man sträva efter att gruppera samman enheter som börjar på likartat sätt.

Vid teckenvis alfabetisering bör ordmellanrum och annan utslutning betraktas som junkturer.

Det kan ibland vara önskvärt att även ge vissa junkturer ordseparerande status (exempelvis ' och - i franska material). Detta beaktas i så fall lämpligen redan då ordningsformen skapas.

Problemen med siffertalen

Hur stort inflytande skall siffertalen ha på den alfabetiska ordningen? Vid alfabetiseringen bör målsättningen vara att,

inom ramen för ett formaliserbart förfarande, försöka gruppera samman semantiskt närstående ord. Man har därför större skäl att hålla samman grupperna

00-talet	2-åring	2-sidig
10-talet	21-åring	4-sidig
1000-talet	50-åring	8-sidig
1800-talet	75-åring	96-sidig
1930-talet		
80-talet		

än en grupp med större semantiska skillnader, som

50-lapp
50-metersbassäng
50-minutersprogram
50-talslåt
50-åring
50-årsdag.

När man ser detta ur språkvetenskaplig synvinkel bör slutsatsen bli att siffrorna inte skall vara primärt sorteringsgrundande, som de alfabetiska tecknen, utan endast påverka ordningen då den alfabetiska delen av orden är identisk, på samma sätt som junkturerna. Att på något sätt väga om junkturerna eller siffrorna påverkar den semantiska uttolkningen mest, är nog att gå för långt; den vinst som eventuellt skulle kunna uppnås genom införandet av ytterligare en nivå i sorteringen uppvägs säkert inte av att användaren då kan få svårt att orientera sig i förteckningen. Det förefaller därför rimligt att utföra sorteringen på två huvudnivåer — först på den alfabetiska delen av ordet (inklusive logogram) och därefter på övriga tecken, dvs. junkturer och siffror. Ord av typen 491, 0,5, 10:50, 1981/82, etc., som saknar alfabetisk del kommer då följaktligen att inordnas först.

I en lexikologisk ordförteckning finns det inga skäl att frångå teckenprincipen och ordna de numeriska leden i talordning. Skulle talordning införas måste rimligen även decimalbråk, allmänna bråk, romerska tal m.m. ordnas efter samma princip. Problemet är att en mycket stor andel av enheterna kan vara homonyma eller polysema: 1,250 kan vara "ett komma ..." eller "ett tusen ...", 2:10 kan vara priset 2 kronor och 10 öre eller 1/5, 2/3 kan vara 2 mars eller en musikalisk taktangivelse eller ett matematiskt förhållande, 19.30 kan vara ett klockslag eller ett pris osv. osv. Ofta har också den numeriska delen av ett ord en proprieliknande funktion, som i årtal eller varubeteckningar (1981, 80-talet, 343, 4-2-4-systemet, 4711, 5A, 8:an).

Att generellt inordna numeralerna enligt en fullständig alfabetisk form är självfallet i allmänhet ogörligt, bland annat på grund av de nämnda homonymiproblemen och problem med böjda och avledda numeraler (skall 1 inordnas som en, ett, första eller förste; skall 1800 tolkas som ettusenåttahundra, ett tusen åtta hundra, artonhundra eller möjligen adertonhundra?).

Den enda rimliga lösningen, för våra syften, är alltså att strikt tillämpa teckenprincipen på siffrorna och junkturerna då den alfabetiska delen av orden är lika.

För speciella tillämpningar finns alltid möjligheten att sortera efter en manuellt supplerad ordningsform. I t.ex. bibliografiska sammanhang, där man har tillgång till ordens kontext i titeln, kan detta vara genomförbart. Man bör emellertid vara mycket restriktiv med inordning på en form som ej är explicit och fullständigt entydig, förutom att man även av andra skäl vill minimera den manuella insatsen.

Supplering och negligering av information för sortering

I vissa sammanhang kan det finnas skäl att åsätta en enhet en ordningsform som kanske inte kan genereras automatiskt, med formaliserbara kriterier. Som tidigare berörts kan detta gälla exempelvis inordnandet av romerska tal (i t.ex. regentlängder vill man kanske ha ordningen I, II, III, IV, V ... och inte I, II, III, IV, IX, V), behandlingen av ord med icke-latinska skrivtecken, samsortering av olika stavningsvarianter under en normalform eller bibliografisk filering där bland annat vissa ord ej får vara rangord. Det är därför ibland av vikt att man har möjlighet att markera att vissa tecken skall negligeras i rangorden, och omvänt, att man skall kunna modifiera rangord eller supplera med ytterligare rangord för att åstadkomma en ordningsform som ger den avsedda ordningen. Dock bör man alltid se upp med manuella ingrepp som innebär en risk för inkonsekvent behandling av enheterna.

Sammanfattning

För att kort sammanfatta denna översikt över de viktigaste problemen vid automatisk alfabetisering av lexikografiskt material kan sägas följande. I ett första steg utväljes de rangord (uppslagsform, ev. ordklass etc.) som skall alfabetiseras. I nästa steg etableras en ordningsform på grundval av rangorden. Ordningsformen har exempelvis spjälkat æ till ae, eventuellt translittererat þ till th, ß har blivit ss, i spanska material har ch och ll fått unika koder och ordseparatorer kan ha inskjutits kring vissa logogram och junkturer. Är sorteringen en annan än initialalfabetisk permuteras sedan denna ordningsform. Därefter genereras den sorteringsnyckel som skall användas vid den maskinella jämförelsen. Sorteringsnyckeln skall vara konstruerad på sådant sätt att hänsyn i en första omgång endast tas till alfabetiska tecken (och logogram) och först då de alfabetiska tecknen överensstämmer beaktas övriga tecken (junkturer och siffror).

För att återgå till den alfabetiska delen, tillgår jämförelsen där på fyra underordnade nivåer: först jämförs grundbokstäverna i orden, därefter (dvs. vid likhet mellan orden) tas hänsyn till eventuella varianttecken, diakritiska tecken och

versaler, i nämnd ordning. Inom varje nivå skall således finnas en intern rangordning mellan tecknen (mellan grundbokstäverna, variantformerna, de diakritiska tecknen, gemener/versaler respektive övriga, icke-alfabetiska tecken). Jämförelsen bör ske ordvis och på ett sådant sätt att ord som börjar på likartat sätt ordnas intill varandra.

Avslutningsvis bör framhållas att det är viktigt att man, åtminstone i mer omfattande alfabetiska register, tydligt beskriver de alfabetiseringsprinciper man tillämpat. Åtminstone tills dess att en standard på området har etablerats.

[En något utförligare presentation med bibliografiska referenser och en modell för implementering (med flödesplan och kodtabeller) finns i Rapport nr 14 från Språkdata. Den kan beställas från Språkdata, Göteborgs universitet, Norra Allégatan 6, S-413 01 Göteborg.]

Benny Brodda
Institutionen för Lingvistik
Stockholms Universitet

81-09-30

"THE TAGGER"
==========

The TAGGER är ett programsystem med vars hjälp man kan skapa och/eller uppdatera taggfiler associerade med en given text på ett semiautomatiskt och interaktivt sätt. The TAGGER är ett för taggningsändamål specialdesignat editeringsprogram där grundideen är att alla inoch utgående filer skall ha enkla, "raka" textformt utan konstiga superstrukturer i form av pekare eller dylikt. I avsnitt 1, nedan, ges en beskrivning av de olika format man har att välja mellan när det gäller taggfilernas utsende.

Den fundamentala frågan (ur teknisk synpunkt) om hur systemet håller reda på associationen mellan tagg och taggat objekt diskuteras i avsnitt 2.

När det gäller själva editeringsförfarandet har The TAGGER ett speciellt kommandospråk, med vars hjälp man enkelt kan ändra eller på annat sätt modifiera sina taggfiler. Detta kommandospråk beskrives i avsnitt 3.

Taggning med hjälp av The TAGGER kan ske på ett semiautomatiskt sätt, genom att man kan foga till ett lexikon till systemet (med regler av typ PÅ => PREP); detta lexikon kan interaktivt ändras under en taggninssession och sedan lagras undan; systemt kan alltså successivt "lära" sig den specifika användarens ideer om hur just hans taggning skall vara. Lexikonets användning och struktur behandlas i avsnitt 4.

Som ovan nämnts så förutsättes att alla input- och outputfiler är i rent textformat (detta gäller även lexikonet). Filerna kan ges godtyckliga namn (dvs olika användare kan använda sina egna filer), men i det följande kommer jag använda följande namn för de olika filerna; dessa namn är också systemets "defaultnamn": Originaltexten, dvs den text som man avser tagga, benämnes INFIL.TXT. Om man redan har taggat texten tidigare men vill ytterligare korr- igera de åsatta taggarna, så utgår systemet från att man har en redan existerande taggfil, OLDFIL.TAG kallad. Den nya (tagg)fil man skapar benämnes NEWFIL.TAG, och har man ett lexikon med "färdiga" taggningsförslag så heter det TAGGER.LEX.

Man bör observera att den ursprungliga originaltexten, INFIL.TXT, förblir alltid oförändrad, det är bara taggfilerna som ändras.

Jag vill avslutningsvis i denna inledning betona vad The TAGGER är och vad det inte är. Det är ett editeringsprogram med vars hjälp man bekvämt kan tagga en text, och den taggade filen som därvidlag åstadkommes har en vettig och enkel struktur, användbar, hoppas jag, för många olika syften. Jag vill dock betona att taggning i sig själv inte ger ett vetenskaplig resultat, det är vad man gör med sina taggade texter som (eventuellt) kan ha vetenskaplig värde. Det man gör med The TAGGER är alltså inte forskning i sig. Rätt använt kan dock program av den typ The TAGGER representerar utgöra värdefulla forskningsinstrument.

1. INTAGGNING OCH UTTAGGNING

Med en "tagg" menar jag fortsättningsvis en grammatisk, syntaktisk och/eller semantisk flagga eller etikett (eng. "tag") associerad med ett ord eller annat lingvistiskt objekt i en textmassa av något slag. Denna textmassa kan vara en vanligt löpande text, men den kan också utgöras av en textmassa som på ett eller annat sätt varit utsatt för någon annan typ av processning före själva taggningen, exempelvis en konkordans. "Taggning" är själva processen att åsätta sådana taggar, och The TAGGER är då ett programsystem som är avsett att underlätta denna process.

Som nästan alltid i datasammanhang är själva nyckeln till framgång det att man har genomtänkta och "rena" datastrukturer. I The TAGGER har jag valt att använda textformat i alla in- och utgående filer. Textfiler kan listas, ändras, lagras och flyttas med vanliga enkla standardprogram, och en välstrukturerad textfil kan alltid användas som utgångspunkt för, tex, ett databassystem; omvändningen är inte alltid sant. Vissa begränsningar ligger i en sådan filosofi; vilka jag strax tänker beröra; jag anser ändå att fördelarna med en sådan filosofi är helt överväldigande och jag anser att det skall bra starka skäl till i det enskilda fallet för att överge det enkla texformatet som primärt lagringsformat.

I det vanliga, enkla fallet utgör taggning egentligen inget större problem, nämligen i det fallet att man behöver sätta på enkla grammatiska taggar (el dyl) på de enskilda orden i en löpande text. Antag, t ex, att vi vill sätta på ordklassbeteckningar på orden i följande mening:

(1) KATTEN SITTER PÅ MATTAN.

På ett eller annat sätt vill vi alltså associera (1) med något i stil med

(2) NOM VBPRES PREP NOM

Den datamaskinella frågan vi då måste lösa är hur vi skall lagra taggsekvensen (2), i förhållande till (1), så att maskinen helt entydigt "ser" att det första NOM:et hör ihop

med KATTEN, att VBPRES hör ihop med SITTER, etc. Hur man väljer att lägga sina taggar beror ju delvis på vad man skall göra med sina analys. Kanske vill man göra en ren ordklass-statistik, och då är ju representationen i (2), där taggarna är helt lösgjorda från ursprungstexten, alldeles utmärkt som den är. Man kan betrakta en fil av poster av typ (2) och betrakta den som en text vilken som helst, och på den göra vanlig ordstatistik. Presto.

I andra sammanhang behöver man kanske ha kvar associationen ord-tagg, t ex om man vill kunna fråga sig vilken preposition som är den vanligaste i den text man håller på att analysera. Man behöver då snarare en representation av typ

(3) KATTEN/NOM SITTER/VBPRES PÅ/PREP MATTAN/NOM

Om man i en text, där "orden" ser ut som elementen i (3), gör en enkel konkordans (t ex) så får man just svar på frågor av den sistnämnda typen.

De två nyss angivna sätten att lagra taggarna utgör två av de grundkonventioner för lagring av taggar som The TAGGER utnyttjar. Format (2), där taggarna är lösgjorda från sina resp. ord, kallar jag för "uttaggning", och användes format (3), där taggarna är direkt fasklistrade på sina resp. ord, kallar jag för "intaggning".

Användes uttaggning som lagringsformat, kommer taggarna att existera i en slags "spökfil", parallell med den ursprungliga texten. Denna spökfil blir på sätt och vis en trogen kopia, eller rättare sagt en spegelbild, av ursprungsfilen, därigenom att radfallet bibehålles och att övriga skiljetecken (punkt, komma o dyl) läges på motsvarande ställen. Detta underlättar ganska mycket för läsaren (och delvis också för The TAGGER) när han/hon skall orientera sig i tagg-texten. Jag återkommer strax med en beskrivning över hur The TAGGER mer i detalj klarar av associationen ord-tagg, när uttagning användes.
De nyss angivna lagringsformaten är säkert ganska goda när det gäller att utföra den slutgiltiga analysen, men bägge är definitivt dåliga ur en synpunkt, nämligen ur presentationssynpunkt, antingen medan man håller på att jobba med sin taggning eller för den slutgiltiga presenta- tionen. The TAGGER har då ytterligare ett lagringsformat, som jag kallar (horisontell) display-taggning, som är mycket väl lämpat för sådan presentation:

(4)
```
KATTEN SITTER PÅ   MATTAN
NOM    VBPRES PREP NOM
```

Användes display-taggning så ser The TAGGER till att ord och tagg alltid matchas rakt ovanför varandra vid en utskrift. Detta underlättar korrekturläsning o dyl.

I The TAGGER kan man fritt växla mellan de olika taggnings-konventionerna för indata och utdata. Man kan t ex mata in en

display-taggad OLDFIL.TAG (taggningskonvention 4) och generera en uttaggad NEWFIL.TAG (taggningskonvention 2). På så vis har man inte mindre än 12 olika kombinationsmöjligheter: man kan ha ottaggad, uttaggad, intaggad och displaytaggad fil som input, och man kan generera en uttaggad, intaggad ller displaytaggad outputfil.
Hitinitlls har jag enbart berört enkel ordtaggning, där man hela tiden har att varje ord alltid tilldelas en tagg och att varje tagg alltid hör precis till ett ord. Tyvärr är det inte alltid så enkelt, och platsen här räcker inte för att idetalj gå genom hur man tekniskt kan lösa vidhängande problem. Problemet är, kortfattat, att man inte alltid kan vidmakthålla detta enkla ett-till-ett förhållande mellan ord och tagg. Antag att att man vill ha både syntaktisk och semantisk analys av sin text. Ordet KATTEN, t ex, i (1), kan associeras med en hel uppsättning taggar, av typ NOM+DEF+SUBJ, orden PÅ MATTAN tillsammans kan behövas åsättas beteckningen ADV+PLACE, och att ovanför kan sekvensen SITTER PÅ MATTAN åsättas beteckningen VP. Allt detta gör att vi kan komma att behöva kunna ange många-till-många relationer mellan taggar och ord på ett ganska intrikat sätt.

2. ASSOCIATION ORD-TAGG

I detta avsnitt är det lättast att rent tankemässigt utgå från ett uttaggat system. Som jag antydde tidi gare är själva grundideen i ett sådant system att åstakomma en fil som är så trogen kopia som möjligt av ursprungsfilen vad beträffar det rent yttre; i någon mening skall den "se" lika dan ut. Utgår man från en rättfram ordtaggning, är ju detta väldigt enkelt att åstadkomma, nämligen helt enkelt genom att använda ordliknande beteckningar för sin taggar (PREP, NOM, VERB, o dyl) Gör man så, så åstadkommer The TAGGER verkligen en trogen "spökfil" parallell till den ursprungliga texten.

Denna enkla grundprincip har jag försökt gene ralisera därhän att resultatet ur The TAGGERs synpunkt blir exakt på det sättet, dvs till varje "ord" i ursprungsfilen skall det finnas ett ordliknande objekt i taggfilen och tvärtom. För att åstadkomma detta måste jag dock precisera mer noggrant vad jag menar med ett "ord"; det är genom att generalisera det begreppet som man kan åstakomma mer komplicerade saker än den rena ordtaggningen.

I "vanliga" sammanhang så utgör ett (text)ord helt enkelt en sammanhängande sekvens av bokstäver, på ömse sidor omgiven av ordavkiljare (punkt, komma, mellanslag o dyl). Var och en som sysslat det minsta med datalingvistik är dock helt bekant med det faktumet att i datasammanhang är det inte alltid så klart vad som menas med "bokstav", det hör ju till den datorlingvistiska vardagen att få texter som innehåller de mest mystiska "diakritiska" tecken instoppade i orden (accentmarkering o dyl). Redan detta gör att man i varje generellt system för hantering av texter måste ha mekanismer

att ha ett kommando med innebörden "dagens alfabet utgöres av ...)

I The TAGGER är detta löst på det viset, att varje tecken i maskinens teckenförråd (i den aktuella versionen av The TAGGER ASCII-alfabetet) tilldelas ett typvärde, ett värde i intervallet 0, 1, 2, 3 och 4. "4:or" är det som definieras som "bokstav". (Graf)ord i The TAGGERS mening är då varje maximal, sammanhängande sekvens av "4:or". Grundideen vid uttaggningen (ja, alltid) är att textfilen och taggfilen skall ges exakt samma grafordsstruktur, med bibehållande av alla andra tecken (inkl vagnret.) på sina ursprungliga platser

I The TAGGER kan man ge ett kommando, "DEFTYP 4=" och där räkna upp "dagens alfabet". Det finns ett defaultalfabet i The TAGGER, nämligen

DEFTYP 4= 'A'-'Z', '0-9' (bokstävr o siffror)

dvs varje sammahängande sekvens av bokstäver och/eller siffror (t ex K2R) utgör "ord" i grundsystemet. Vill man betrakta engelska ord av typ CAN'T som ett enda ord så räcker det att föra till "'" till mängden "4:or", alltså med kommandot "DEFTYP 4= '''' (systemt förutsätter att man sätter apostrof kring de tecken man anger, även kring apostrofen).

Anledningen till att siffrorna också är "defaultade" som bokstäver är att det är mycket naturligt att använda taggbeteckningar av typ VB1, VB2, ADV1, ADV2,... för olika underklasser av verb, adverb o dyl. Vill man inte ha siffrorna behöver man bara typa om dem till, t ex, "2:or", vilket är normal värdet för "icke-bokstäver".

Genom att utnyttja denna teckentypning på ett övertänkt sätt kan man nu åstadkomma mycket generella taggningningssystem. Antag att man vill sätta på mer en än tagg på varje enskilt ord, dvs ge den till ordet hörande taggen en inre struktur. Säg att vi vill tilldela ordet KATTEN analysen NOM+DEF+SUBJ+ ANIM. Ja, det kan vi göra precis på det sättet bara genom att typa plustecknet som en 4:a. The TAGGER kommer då att uppfatta hela den nyss angivna sekvensen som ett enda graford, och kan alltså lätt hålla ihop associationen ord-tagg. Varje enskilt tecken i ASCII-alfabetet kan användas som "bokstav" i den här meningen.

Om man vill kunna åsätta hela grupper av ord en skilda taggar blir det genast lite mer kom plicerat, men i princip kan man göra på det sättet att man editerar in paranteser i ursprungsfilen kring de ord man anser utgöra en enhet. Dessa paranteser typas sedan som "4:or", och blir, om de editeras in med luft runt omkring, ur The TAGGERs mening självständiga ord, vilka alltså också blir åtkomliga för taggning. Den således åstadkomna taggningen svarar då precis mot en "labelled bracketing".

3. KOMMANDOSPRÅKET

Själva påförandet eller korrigerandet av taggarna till en text med hjälp av The TAGGER sker med hjälp av ett kommandospråk. Kommandona i detta språk utgöres dels av lokaliseringskommandon, sådana som innebär att man går en fram och tillbaka i den aktuella filen, och dels de rena edite ringskomandona, sådana kommandon som direkt ändrar (eller låter bli) att ändra taggar. I den mån man för på eller ändrar saker så är det uteslutande object (taggar) i taggfilen som ändras, själva ursprungsfilen, INFIL.TXT, påverkas inte av The TAGGER.

Kommandona i The TAGGER har följande syntax

(5) (n) OP (arg)

där n är antalet ggr operationen OP skall utföras, OP är en operator, ett tecken ur den specifika uppsättningen TAGGER-operatorer som systemet ut nyttjar, (f n tecknen /, :, ;, <, =, >, ?, @), och som har det defaultade typvärdet DEFTYP 3. Normalt kan man ge godtyckligt många kommandon på en gång, så länge de ryms inom en rad (om kombi nationskommandon skall användas bör inte typningen av operatorerna ändras). Argumentet <arg> till en operator bör vara ett "graford" enligt avsn. 2. Varje kommando(rad) som innehåller mer än ett tecken totalt lagras automatiskt undan, och kan refereas till genom kommandot "=", vilket kommando alltså innbär "repetera senaste något så när komplexa kommando".

När man kör systemet presenteras posterna en efter en på skärmen i display-format (4), enligt avsnitt 1. Om taggningssessionen innebär en uppdatering av en existerande taggfil (OLDFIL.TAG), så presenteras däri förekommande taggar en efter en, och man kan antingen låta dem förbli som de är, eller ändra dem (med komandot "ändra aktuell tagg"; se nedan). Om taggfilen man håller på att ändra till väsentliga delar är OK kan man hoppa antingen framåt i posten eller ta längre kliv i hela filen med hjälp av lokaliseringskommandona (väsentligen ":").

Om taggningen rör en "fräsch" textfil, dvs en som man inte har någon taggfil till, så presenterar systemet antingen tillverkade taggar (a1, b1, c1, ..., a2, b2,...) vilka fungerar som "dummies" - ur The TAGGERs synpunkt är det enbart viktigt att de rent formellt ser ut som taggar - eller så presenterar systemet förslag till förmodat mer vettiga taggar, enligt det lexikon, TAGGER.LEX, som man optionellt kan ha fogat till systemet. Dessa föreslagna taggar kan man antingen acceptera (kommandot ">"), eller förkasta ("@"), varvid systemet föreslår andra taggar för samma ord, så länge alternativ finnes i lexikonet. Man kan också manuellt editera in det rätta alternativet, om inget av de presenterade dög.

Här följer en kort översikt över de kommandon som f n är implementerade i The TAGGER (version 8110):

OP	ARGUMENT (typ)	BETYDELSE
/	lexikonregel	För in lexikonregeln i TAGGER.LEX
:	ID-begrepp	Lägg ut post, och stega fram till angiven post (förutsätter rad-id)
;	sträng	Ändra akt. tagg till "sträng"
<	tagg	Backa till angiven tagg
=		Repetera senast komplexa instruktion.
>	tagg	Acceptera akt. tagg, och stega ett steg eller till angiven tagg.
?		Displaya akt post en gång till
@		Förkasta föreslagen tagg och begär fram en ny (om sådan finnes)
<CR>		Stega en tagg utan att ändra akt.

Obs! Argumenten är i samtliga fall optionella. Om man vid lokaliseringskommandona inte ger en sträng, "går" systemet ett steg. Ett rent ":", t ex, betyder "Skriv ut akt. post och tag in nästa". Om man vid "/" avstår från argument, sättes en regel in i lexikonet med innebörden "Ge i fortsättningen akt tagg som förslagsalternativ till akt. ord".

Det är många operativa finesser med systemet som jag här inte alls har plats att beröra. Det finns speciella körinstruktioner som tar upp allt det mer i detalj. Låt mig bara nämna en facilitet, som har visat sig mycket användbar, nämligen att man kan köra The TAGGER i vad man kan kalla "run off mode". Antag att man har en "fräsch", otaggad text och ett taggningslexikon. Run off mode betyder att man låter The TAGGER köra igenom hela texten i ett enda svep, varvid den sätter på taggar på de ord den har i sitt lexikon (alltid första alternativet), och låter "dummy"-taggarna stå kvar på övriga ord. Utfilen, som då lämpligen ges display-format, kan man sedan ta ut som en radskrivarutskrift, vilken man i lugn och ro kan sitta hemma och korrekturläsa.

4. LEXIKONET

Jag har ovan upprepade gånger refererat till att man optionellt kan foga ett lexikon till The TAGGER, ett lexikon som systemet utnyttjar till att komma med mer eller mindre vettiga förslag till taggar. Default-namnet på detta lexikon är TAGGER.LEX.

Formatet på lexikonet är att under ordet "REGLER" kommer lexikonreglerna en efter en. Lexikonreglerna skall ha följande utseende:

(6) xxxx; yyy; C

där "xxxx" markerar det ord (el motsvarande) i originalfilen som skall taggas av lexikonet, "yyy" är den tagg man vill

asätta "xxxx", och "C", slutligen, är ett enkelt kontextvillkor, vars betydelse jag strax skall ge.

Samma element "xxxx" kan förekomma flera gånger i lexikonet. Vid en aktuell taggningssituation föreslår systemet taggarna i den ordning man lagt in dem i lexikonet; det är alltså viktigt att försöka Lägga in lexikonorden i rätt ordning.

Exempel: ordet PÅ är säkert i 80 % av alla förekomster en vanlig preposition, och i återstående fall en partikel, och kanske i något enstaka fall något slags adverb (kan vi antaga för exemplets skull). Då bör man i lexikonet skriva

```
(7)   PÅ; PREP;  3
      PÅ; PTCL;
```

och inte tvärtom. Det tredje, lågfrekventa fallet behöver man inte alls ta med, efter som man hela tiden har möjlighet att helt förkasta systemets inbyggda förslag och i stället för hand föra in egna taggar.

Talet "3" som står i exemplet är ett exempel på hur kontextvillkoren skrives, och "3" betyder helt enkelt att det är ord i sin helhet som lexikonet tittar på. Mer specifikt betyder "3" "Ordavgränsare både till vänster och till höger", eller, med andra ord, ord i sin helhet. Övriga tillåtna kontextvillkor är "1", som betyder "Ordavsklijare till vänster", (dvs i ord början) och "2" = "ordavskiljare till höger" (dvs ord slut). Med denna konvention kan man också ha regler av typ:

```
(8)   ARNA; SBPLUR;  2
      AR; SBPLUR;
      AR; VBPRES
      BE; VB;        1
      PÅ; VB;
```

etc. Med den konventionen så skulle ordet POJKARNA få ett förslag SBPLUR, likaså ordet POJKAR. Verbet HOPPAR skulle först ges samma förslag, men efter ett "@" (förkasta förslag) ges man det korrekta VBPRES. Ett ord som SOMMAR finge man tagga för hand, t ex genom kommandot ";SBSG" (ändra föreslagen tagg till SBSG). Kon villkoren ("2" resp. "1" i exemplet) behöver inte upprepas om de är samma.

Ordbörjankontexten "1" kanske är mindre användbar, men i princip kan det ju vara en god gissning att ord som börjar på BE eller PÅ är verb. Sådant får man prova sig fram till.

Jag vill påpeka att man med hjälp av programsystemet BETA kan göra betydligt mer avancerad analys än den man kan åstadkomma inom ramen för The TAGGER, och därför finns det knappast någon anledning att bygga ut själva taggningsprogrammet med än mer sofistikerade analysmekanismer. Observera också att man mer sofistikerade inbyggda analysmekanismer. Med Beta kan man nämligen också se till att utfilen får antingen ett

uttagat eller ett intaggat format, dvs utfilen från Beta kan ges sådant format att den fungerar som input till The TAGGER, i vilket program man sedan manuellt kan göra slutjusteringarna.

Gunnar Thorvaldsen
Registreringssentral for historiske data
Universitetet i Tromsø

NORMALISERING AV PERSONNAVN.

Historiske navnedata som folketellinger og kirkebøker blir brukt til mange forskningsformål. Ofte består mye av kildestrevet i å finne igjen personer fra den ene kilda til den andre for å få oversikt over individers livsløp. Skal dette arbeidet bli overkommelig, er det nesten en forutsetning at kildene fins i maskinleselig versjon. Gjenfinning kan tenkes foretatt automatisk ved hjelp av EDB-programmer, men hittil har den vanlige metoden vært å sortere kildene etter ulike nøkler, og finne fram i datalistene manuelt.

Det viktigste kriterium for identifikasjon er gjerne for- og etternavn. Den som prøver metoden, finner snart ut at på 1800-tallet var skrivemåten av navn langt mindre konsekvent enn nå. Vi kan si at skrivemåten snarere fulgte skriveren enn den som bar navnet. Presten eller folketelleren skrev Niels eller Nils etter eget hode.

Dette medfører at arbeidet med identifikasjon tar svært mye tid. Skal vi se etter Anne eller Ane? Heter hun Ni(e)lsdatter til etternavn kan vi i alle fall tenke oss 4 kombinasjoner. En form for normalisering eller standardisering vil dermed kunne spare oss for mange oppslag i listene. Og skal automatisk lenking ha noe for seg, må EDB-programmene bli fortalt hvilke navneformer som sannsynligvis er synonyme. Siden kildene brukes bl.a. i navneforskning er det ikke akseptabelt at navneformene endres før de legges inn i maskinen.

Her har vi emnet for denne artikkelen: Hvordan kan vi rasjonalisere identifikasjon ved å redusere antall navneformer uten å ta bort noe vesentlig av den informasjon som ligger i at navn skal være ulike? Siden siktemålet er praktisk, skal vi ta forholdsvis lett på fonetisk teori omkring navneformer. Vi legger heller til grunn en mer intuitiv forståelse av hvilke skrivemåter av navn som er synonyme, og konsentrerer oppmerksomheten om det tekniske. Det betyr på ingen måte at kommentarer til denne artikkelen bør følge samme oppskrift!

Vi kan i utgangspunktet tenke oss å utføre normaliseringen på to ulike hovedmåter. Enten ved å formulere formelle kriterier for hvilke tegn og tegnsekvenser som skal forandres (alle forekomster av -ie- forandres til -i-?). Eller vi lar EDB-programmet slå opp i ei ordliste hvor det

for hvert navn finner hvilken form som er standard. Jeg har selv prøvd begge disse opplegg på et par folketellinger. En vurdering av resultatene følger her.

Utgangspunktet var folketellinga for Alta i 1875. Her fins navn og andre opplysninger om 2419 individer. Et EDB-program ble skrevet som går igjennom tellinga, plukker ut alle fornavn, sorterer og oversetter hvert av dem ifølge formelle kriterier etter førstnevnte modell.

Hvilke kriterier var innbakt i programmet?

1. To like tegn reduseres til ett. (Anne blir Ane)
2. h fjernes, men ikke i begynnelsen av ord. (Alethe blir Alete)
3. c blir til k, foran e og i til s. (Carl blir Karl, Cecilie blir Sesilie)
4. aa blir til å (unntak fra regel 1).
5. ou blir til au. (Poul blir Paul)
6. w blir til v.
7. ph blir til f. (Philip blir Filip)
8. I endelsene -na, -ia, -ta, -ka, -la blir a erstattet med e. Mia blir Mie.
9. I endelsene -rd, -id, -ld, -nd blir d fjernet. Sigrid blir Sigri.
10. æ blir til e (Bæret blir Beret)
11. Endelsen -af og -au blir til -av. (Olaf blir Olav)

Kriteriene er resultat av mange navnelister og mye prøving og feiling. En nyttig metode var å sortere navna baklengs for å få oversikt over endelsene. Flere algoritmer måtte fjernes, f.eks. hadde det liten hensikt å erstatte ie med i i Niels når Daniel samtidig ble Danil.

Resultatet ble ei liste med ustandardiserte og standardiserte navneformer hvor programmet etter mitt syn har gjort mer nytte enn ugagn. Dvs. at for hvert kriterium er antall vellykkede standardiseringer større enn antall mislykkede. For hvert navn har programmet også satt på antall forekomster av navnet i kilda og et merke for type standardisering.

Den oppmerksomme leser vil forstå at lista inneholder en rekke navn som har blitt forandret uten grunn. F.eks. vil det bare forvirre når Johan finnes som Joan. Dessuten fins det mange standardiseringer som det er tungvint å skrive algoritmer for fordi de er vanskelige å innpasse i noe mønster.

Dette gjelder f.eks. fjerning av den siste e'en i Henery. For å gi en ide om problemenes omfang har jeg laget en tabell over forekomsten av de ulike typer standardisering (vertikal variabel) og hvor vellykkede de ble (horisontal variabel). Verdiene vertikalt tilsvarer standardiseringstypene ovenfor, blank er ingen standardisering.

Horisontalt står "-" for mislykket standardisering, "." for manuell standardisering og "&" for ukurant term ("udøpt", "barn" o.l. kan knapt kalles fornavn).

		. MANUE	- MISLY	& UKURA	INGEN	SUM
1	DOBBEL	7	6	2	48	63
2	H BORT	15	3	0	58	76
3	C>S/K	1	0	0	10	11
4	AA> Å	2	0	0	5	7
5	OU>AU	0	1	0	1	2
6	W > V	1	0	0	2	3
7	PH> F	0	1	0	1	2
8	-NA>E	10	8	0	81	99
9	-RD>R	3	0	0	26	29
10	Æ > E	1	0	0	3	4
	INGEN	68	0	5	272	345
SUM		108	19	7	507	641

Vi ser at de 2419 menneskene i 1875-tellinga for Alta hadde 641 ulike fornavn. (Doble navn er delt opp, ett-tegns forkortelser er kuttet ut.) Når det gjelder opptelling av de ulike typer standardisering må jeg understreke en viktig begrensning. For hvert navn er det bare plass til en kode. Mange navn blir standardisert etter flere kriterier, og da er siste tilslag avmerket.

En form for maskinell standardisering er foretatt i nesten halvparten av tilfellene. Som regel dreier det seg om forandring av endelser etter regel 8 og 9, fjerning av h eller forenkling av doble tegn (regel 1 og 2). Langs den vertikale variabelen ser vi at det varierer hvor vellykket programmet arbeidet, men i under 10% av tilfellene gjorde det vondt verre. Dette er rettet med editor.

Verdien "manuelt" gjelder noe annet. Dette er tilfeller hvor navna ikke var standardisert maskinelt, men hvor vi allikevel kunne ha nytte av et redusert antall skrivemåter. I mange tilfeller gjelder det også navn som allerede var forandret i programmet, men ikke nok. Dette er gjort med editor og gjelder ca. 17% av navna. Med tillegg for de mislykkede, måtte 20% endres manuelt. Det viktigste er allikevel at de 641 navneformene er redusert med vel en en tredjedel til 420.

Resultatene blir noe anderledes for andre kilder eller regioner. Men det kan neppe rokke ved den konklusjon at mange navn kan standardiseres ved hjelp av forholdsvis enkle algoritmer. Disse vil imidlertid lett gripe inn der det er uønsket, samtidig som de ikke tar seg av en rekke navn som vi ønsker å standardisere. Problemene kan møtes med mer

kompliserte regler for konvertering, men antagelig vil en annen strategi være mer rasjonell: Søking i navnelister som nevnt innledningsvis.

Ei slik liste må inneholde to former av navna, den originale og den standardiserte. Og det var jo ei slik liste vi nettopp produserte! Denne lista inneholder da alle fornavn i den kilda vi tok utgangspunkt i, slik at navna der nå kan standardiseres ut fra den. Søking i slike lister tar nødvendigvis en del maskintid siden vi får et oppslag pr. navn, men ved såkalt binær søking går det forholdsvis raskt.

Et annet problem er at når vi går løs på nye kilder vil det dukke opp navn som ikke finnes i lista. Vi må derfor ha en prosedyre som standardiserer dem, først maskinelt og deretter manuelt. Når de er kontrollert kan de inkorporeres i navnelista slik at den gradvis bygges opp. For at den ikke skal bli for stor og uhåndterlig kan vi sette grenser for hvilke navn som kommer med, f.eks. kutte ut alle med forekomst lik en.

Ved å trekke inn folketellinga fra år 1900 for Harstad har jeg prøvd ut et slikt opplegg. Her bodde 2109 personer med tilsammen 3306 fornavn. 811 var ulike, og av dem var det 511 som ikke fantes i 1875-tellinga for Alta. Altså en betydelig restkategori, men hele 344 av disse forekom bare en gang. Dette tyder på at det vil gå forholdsvis raskt å bygge opp et register over frekvente fornavn med standarder.

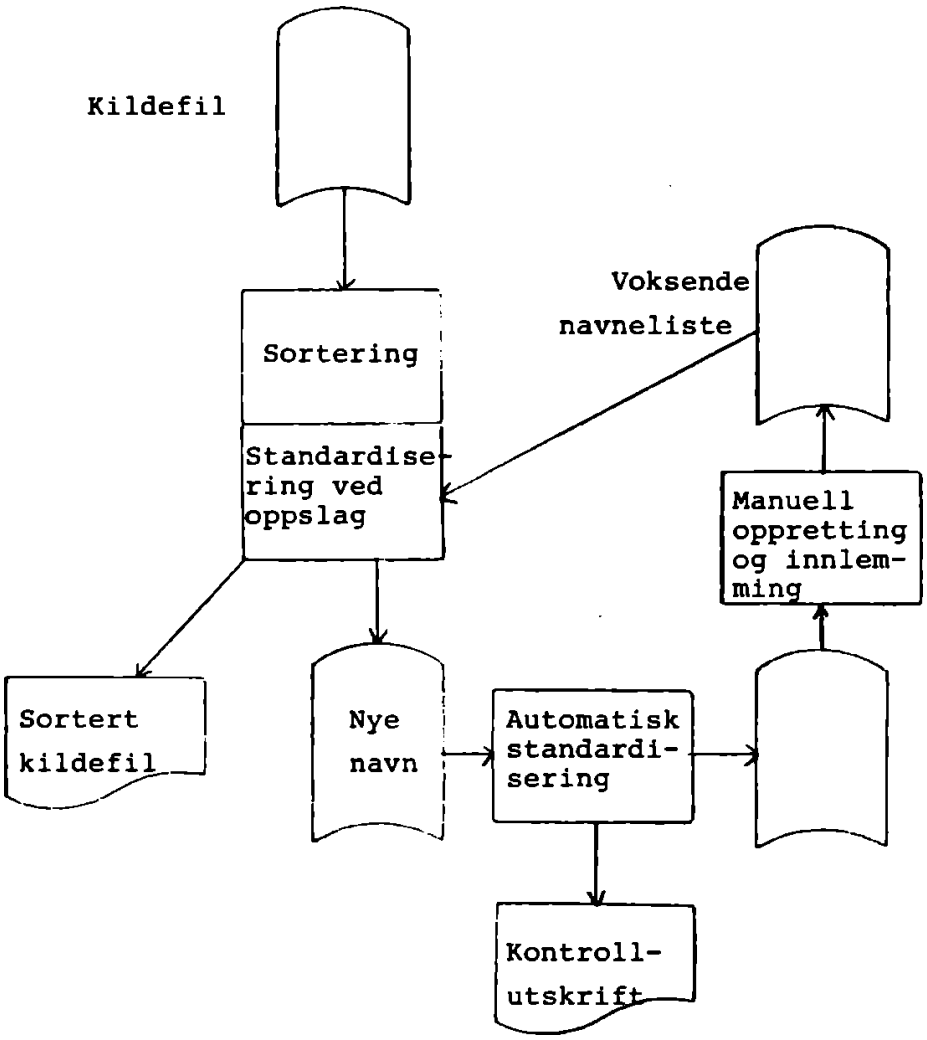
Det kan gjøres med utgangspunkt i følgende flytskjema:

Hver gang vi sorterer ei kilde på navn og ønsker normalisering, slår programmet opp i lista over navneformer med standarder. Når et navn ikke fins her, kommer det med i lista over nye navn. Denne krympes slik at hvert navn bare forekommer en gang sammen med en frekvens. Lista gjennomgår deretter en automatisk normalisering, og på grunnlag av kontrollutskriften foretas en manuell oppretting. Eventuelt kan man utelate de minst vanlige navna før den nye lista innlemmes i den gamle, voksende navnelista.

Hvis det var mange nye navn, vil det lønne seg å kjøre sorteringsprogrammet om igjen for den aktuelle kildefila slik at alle navn i utskriften er sortert etter sin normalversjon.

Slik det er skissert her tar systemet seg bare av fornavn. Det vil imidlertid være en enkel utvidelse å ta med etternavn. Og når det gjelder patronymika, kan man basere seg på fornavnlista.

FLYTSKJEMA FOR NORMALISERING AV PERSONNAVN



ALLMENN NYTTE AV DATA FRA RHD I NAVNEGRANSKING

Registreringssentral for historiske data ved Universitetet i Tromsø har som oppgave å databehandle folketellinger og kirkebøker fra 1800-tallet for utvalgte deler av Norge. Siden kildene skrives mest mulig bokstavrett av, mener vi EDB-utgavene også vil være til nytte i navnegransking.

Navnetilfanget i materialet omfatter både person- og stedsnavn. Når det gjelder personnavn skal de nevnte kildene inneholde hele befolkninga og dermed utgjøre en bortimot fullstendig navnesamling. (Et viktig unntak gjelder etniske minoriteter).

Når det gjelder stedsnavn gir nok de nevnte kildene bare en grovere oversikt. De aller fleste gårds- og bruksnavn er med, ofte med ulike skrivemåter i de ulike kildene. Det er noe mer tilfeldig i hvilken grad tellerne har fått med navn på mindre (husmanns) plasser og jordlapper.

Imidlertid avspeiler kildene godt hvor mange måter man kunne skrive (og uttale) et navn på. Dessuten åpner de store muligheter for forskere som vil sette navn i sammenheng med egenskaper som yrke, fødested o.l.

Hvordan kan vi så presentere data for brukerne? For det ene kan vi levere maskinskrevne kopier av kildene. Videre kan vi alfabetisere personer og bosteder etter de ulike navnetyper før utskrift. Og endelig kan vi levere magnetbånd med navnetilfanget til forskere som selv ønsker å bearbeide kildene med EDB.

Registreringssentralen vil i første omgang konsentrere seg om kilder fra utvalgte deler av landet. En foreløpig oversikt er vedlagt. Både Nord-, Midt-, Vest- og Øst-Norge blir representert. Vi tilstreber også en næringsgeografisk representativitet. Helt avgjørende er allikevel forskernes behov. Meld derfor fra om konkrete forskningsprosjekt hvor det er behov for kildematerialet.

På lenger sikt er det aktuelt å starte registrering av materiale som vil være mer detaljert mht. stedsnavn. Vi tenker da f.eks. på matrikkelforarbeide og pantebøker.

Eric Grinstead
Centralinstitut for nordisk asienforskning
København

A METHOD FOR THE STUDY OF COMPOUND WORDS IN CHINESE

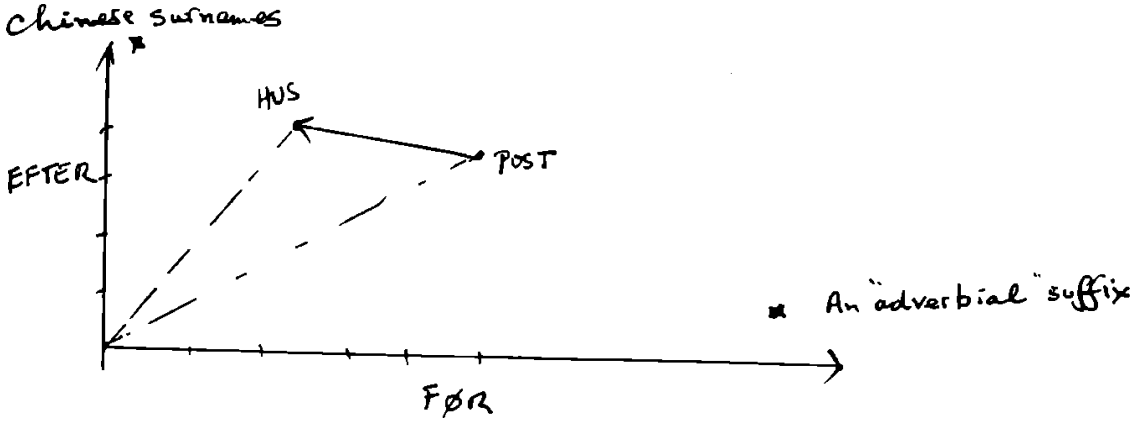
Lexicostatistical work usually treats script letters as the basic unit. This is not suitable for Chinese, where the basic unit contains a meaning, a pronunciation, and a form.

The "word" we study is actually of two signs, so we must study two-element words. I have tried to find properties that are independent of form, meaning, and pronunciation, but incline to consider the meaning important when analysis is possible.

In a given vocabulary, an element can occur first or second. The number of occurrences before and the number after form a pair that can be represented as a Cartesian point, or as prefer it, a vector.

A two-element compound is defined by the line joining POST to HUS, in vector terms POST minus HUS.

I am only beginning to see what the consequences are, but so far a sample of the less common characters gives a good correlation with the rare characters of a new dictionary published in China, even though the criteria for selection are based only on compounds in my list, and on occurrences in Classical Chinese in the dictionary.



Milan Bílý
Department of Linguistics
University of Lund, Sweden

EXPERIENCE WITH COMMENTATOR, A COMPUTER SYSTEM SIMULATING VERBAL BEHAVIOUR

0. The project "COMMENTATOR" at the department of general linguistics at the university of Lund is intended to test ideas about language production. The system implemented in BASIC on the ABC 80 micro-computer generates a scene on the monitor where two persons, Adam and Eve, move randomly around a gate. Not only the present positions of Adam and Eve are shown on the screen but even the positions before the last "jump". This setting is also used for presenting human subjects the same sort of stimuli as the computer. The moves are generated randomly but the operator can choose the lenght of jumps. The initial placement of Adam and Eve can be determined by the operator, too, as well as the instruction for the machine concerning the "focus of attention" (Adam or Eve) and the primary goal of the focused actor (the gate or the other actor). On the operator's command the computer makes written comments on the development happening om the monitor screen. (The present version of COMMENTATOR comments in Swedish but it isintended to use the same set of abstract semantic presications "percieved" by COMMENTATOR for production in several languages, all according to the operator's choice. As COMMENTATOR is a research tool, it does not use any ready-made sentences describing foreseeable situations.)

1. The system works roughly as follows: From the primary information (the coordinates of the gate and the two actors) some more complex values are derived (distances, relations "to left", to right" etc). Then the topics and their "goals" are determined. After that the conditions are tested for the use of the abstract predicates in the given situation - the so-called question menu. This results in positive or negative abstract propositions. The abstract sentence constituents are ordered as subjects, predicates, and objects. Connective elements are added if possible. These connect the last propositions to the previous ones, i.e. conjuctions or connective adverbs are inserted in the proposition. The use of proper names, pronouns, or other NPs is chosen on the basis of reference relations to the preceding proposition. The abstract propositions are substituted by surface phrases and words. The assembled structure is printed. When the whole repertoir of comments is exhausted, a new situation is generated on the screen and the process is repeated. (For a more extensive description of the program and one version of the program itself see Sigurd 1980.)

2. To my knowledge , COMMENTATOR is the only system of its sort in Sweden, if not in the whole of Scandinavia, but there exist some related projects in other countries implemented on larger computers, such as SUPP described in Okada (1980). (SUPP is primarily aimed at recognition of picture patterns.) However, a lot of linguistic research has been done in recent years that will appear useful for the further development of automatic systems of this sort. Badler (1975) is one example of descriptions relevant for COMMENTATOR;

for additional bibliography see Sigurd (1980), Viberg (1981), Okada (1980).

3. The text produced by COMMENTATOR may look like this:
   Eva är till höger om Adam. /Eve is to the right of Adam./
   Han är till vänster om henne. /He is to the left of her./
   Han är till vänster om porten också. /He is to the left of the gate, too./
   Han närmar sig den. /He is approaching it./
   Han närmar sig Eva också. /He is approaching Eve, too./
   Hon är närmast porten dock. /lit. She is closest to the gate, however./
   Hon är inte nära den. /She is not close to it./
   Adam är inte nära den heller. /Adam is not close to it, either./

As can be seen from the sample, the commentary does not vary very much, owing to the limited vocabulary available to the program at the present stage of development. However, to enlarge the vocabulary would not be difficult. A real problem is to instruct the computer how to avoid unnecessary repetitions of redundant information. The second sentence of our proof exemlifies this. The first sentence already implies what is repeated (from "Adam's point of view") in the second sentence. There must be a cartain ordering of propositions in the generating subsystem that guarantees that the amount of redundancy is limited. This ordering must also exclude correct but quite misleading commentaries such as the following sequence describing the situation after Adam's last jump:
   Han är nära porten. /He is near the gate./
   Han är i den. /He is in it./

Thus what is needed is a better question menu than the existing one. The questions must be ordered according to their importance starting from the most relevant question down to less important question. The importance of this ordering is obvious when we consider the use of negation. In the preliminary versions of COMMENTATOR there were unproportionally many negated sentences. Now negated sentences are limited to "answers" to the natural continuations down the question menu. In other words, the first statement about a certain actor and a certain "goal" cannot be a negated sentence. E.g., "Adam närmar sig Eva. Han är inte nära henne dock." /Adam is approaching Eve. However, he is not close to her./

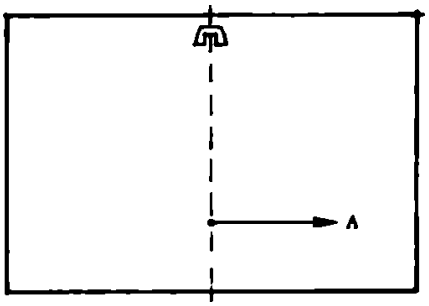
4. Another question that has to be solved satisfactorily is the question of correct reference. COMMENTATOR today is able to do some simple pronominalization, such as substituting the full subject or object of the previous sentence by a pronoun, correctly specified as to grammatical case and genus in a following sentence. It cannot yet refer via a pronoun to both the subject and object of the previous clause. ("Adam närmar sig Eva. De är nära porten"./Adam is approaching Eve. They are near the gate./) What is more important, the program can't observe the restrictions put on pronominalization by the rules of Functional Sentences Perspective. This is not so obvious in the present version of COMMENTATOR, where all NPs are pronominalized if the same (coreferential) NP occurs in the previous sentence. However, intersentential pronominalization is not always obligatory and it remains to motivate pronominalization or

its absence. The necessity of discovering explicit rules of coreference formulated in FSP terms would become obvious if the produced text were in a language distinguishing between degrees of Communicative Dynamism expressed by various sorts of pronouns, such as Czech, Polish, Russian or Chinese, which differentiate between the "most given" subjects with zero pronouns versus thematic, though to a lesser degree, pronominal subjects, or Czech and Polish, which make the same differentiation as to pronouns in oblique cases (the "most thematic" objects have reduced, enclitic forms while the "less thematic" objects have full forms). (For the discussion of this problem see Bílý 1981a, Chapter 3.)

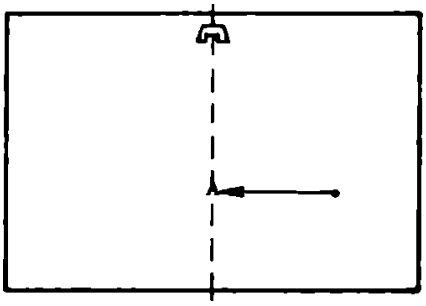
5. As may have been expected, the exact, explicit meanings of the predicates are crucial for the succes of COMMENTATOR. In its present version, all such predicates as "is approaching", "is moving away from", "is to the left of", "is to the right of" etc. are based on measuring the absolute, physical distances in terms of rows and columns on the screen. This is, of course, hardly satisfactory. The result is that if Adam is twenty rows below the gate and one column to the left from the center of the gate, "Adam är till vänster om porten."/A. is to the left of the gate./ will be one of the comments produced, which is hardly compatible with comments of human speakers. Similarly, it is stated that "Adam närmar sig porten."/A. is approaching the gate./ when the distance between Adam and the gate after the last jump is smaller than the distance before this jump, all measured in absolute inits via Pythagoras' theorem. In many cases this may be correct, but at times it feels completely wrong because the direction of the jump points to the conclusion that Adam is bound to miss the gate with his present direction, the shorter distance notwithstanding. The boundary for "approaching" depends obviously on the distance from the object approached. The lesser the distance the less derivation from the right course is accepted. It also depends on the speed of the approaching object (judged in our case by the length of the jump). The longer jump in the wrong (or not quite correct) direction, the harder restriction on "approaching". The insufficiency of measuring the "reality of the screen" in physical units only also becomes obvious when both objects are moving. It is not correct to say that A and E are approaching each other if the distance between them has diminished after the jump but they have passed the point of minimal distance and are, in fact moving away from each other again. A third case which has not yet been taken into consideration is a movement circumvening unpassable obstacles blocking the path. Even then it is not possible to express the conditions for the use of the predicate "is approaching" in terms of physically measurable distance in the present simple way.
It seems necessary to distinguish between states ("is near to", "is to the left of" etc), results ("has approached", "has moved away from" etc), and processes (" is approaching", " is going away from" etc).

Thus the picture of the usual predicates used by different versions of COMMENTATOR becomes quite complicated. What is quite interesting is that certain predicates that one would be inclined to consider symmetric poles in a contrary opposition appear to be asymmetrical- for example, "is approaching" and "is moving away from" show quite

different restrictions. "Is moving away" is possible in certain cases where movement in the opposite direction couldnot be called "is approaching":

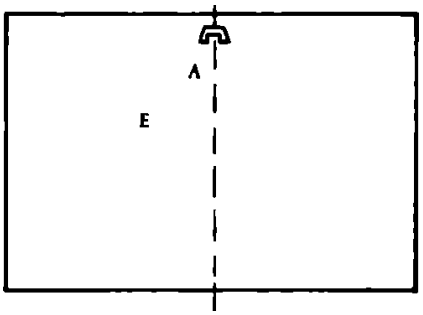


Adam is moving away from the gate.

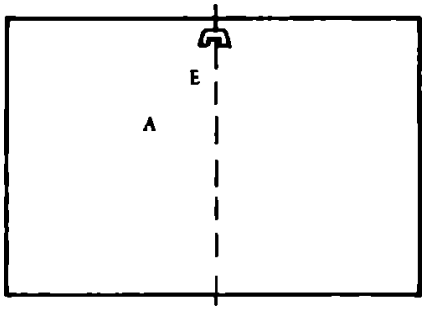


?? Adam is approaching the gate.

It is well-known that comparatives and superlatives of quantitative adjectives express differences of comparable in some absolute units, while positives are relative values, non-measurable in any absolute units. (The best poet in the town does not have to be a good poet. If Adam is nearer to the gate than Eve, he still does not have to be near to the gate etc.) Therelativity of the positives becomes obvious when, e.g. the predicate "is near to" is chosen the first time in a "microtext". The utmost boundary for nearnes is thus established which cannot be passed:

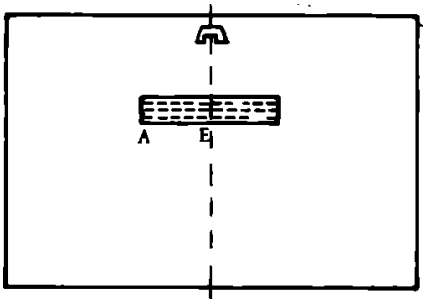


Adam is near the gate.
?? Eve is also near the gate.



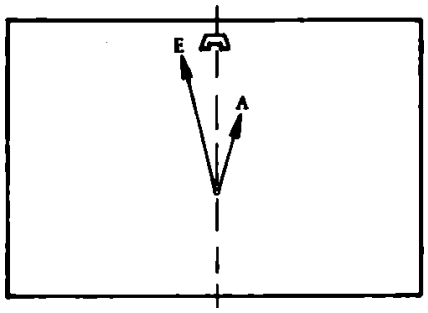
Adam is near the gate.
Eve is even near/Eve is also near the gate.

At least some informants differentiate between "being nearer" and "standing nearer""



A is standing nearer the gate (than E)
E is nearer tha gate (than A)

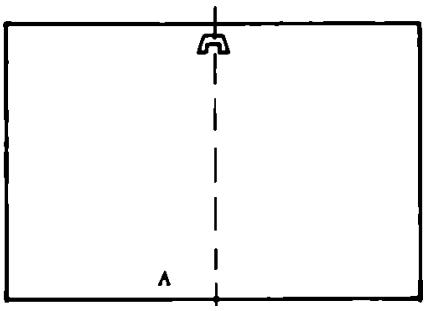
Thus "standing nearer" behaves in a way similar to "approaching". If Adam moved from the place where Eve is now, he would be approaching the gate, too. "Standing nearer" is thus perceived as something like "the result of a movement at the speed of zero", while "being" is purely static. Thus "being nearer/nearest" etc is a state expressed absolutely in the difference of physical measurement units, "being near" is a state measured relatively (in some relation to some reference point th size of the object one is near to, the frame of the background etc.),
Some more examples may illustrate my point:



A is nearer to the gate now.
A is approaching the gate.
A has approached the gate.

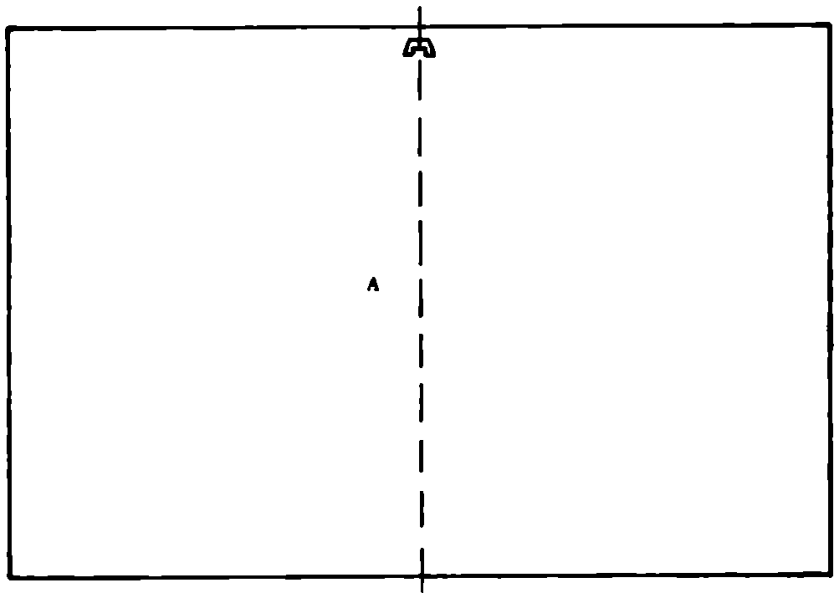
E is nearer to the gate now.
*E is approaching the gate.
E has approached the gate.

As I have alreadymentioned, "to the left of" is a function of the vertical and horizontal distance. Another factor that must be taken into consideration is the size of the referential frame:



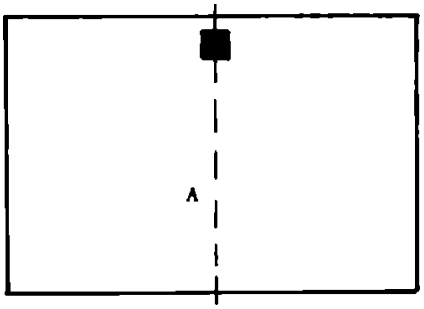
?? A is to the left of the gate.

If the frame is sufficiently enlarged (the more, the better) the same predication becomes quite okay with the distances between A and the gate kept unchanged:

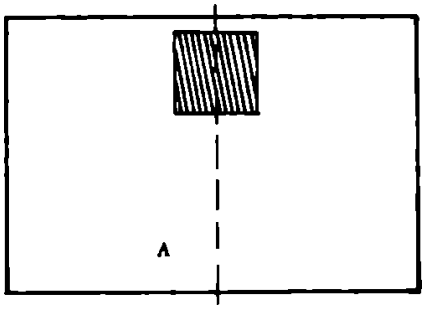


A is to the left of the gate.

The size of the objects related is of equally importance:



??A is to the left of the square.



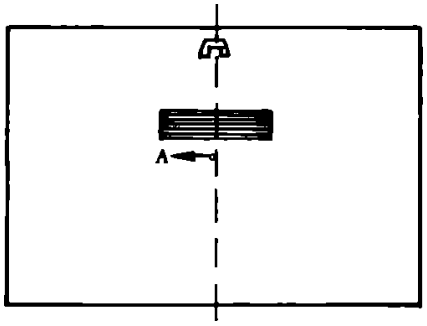
A is to the left of the square.

To sum up, it seems that the usual (and in my opinion rather boring) semantic analyses of the sort "a bull is a male cow" are hopelesly static. What is needed are dynamic semantic descriptions, i.e. descriptions in terms of algorithms applied in the process of mental computation when the choice of the appropriate lexical items is carried out.

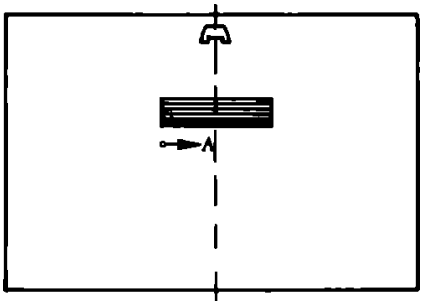
6. When describing positions and movements in my COMMIE, the revised COMMENTATOR program, we obtain something like following set of programeters (The descriptions must be seen as first approximations only, we can hardly expect that they would hold even for a more advanced description of a larger vocabulary.):

1. States: +Relative ("being near to" etc)
2. States: -Relative (being nearer to" etc) - meazurable absolutely
3. Results: -Relative, +Directional ("standing nearer to", "having approached" etc)
4. Processes: +Relative, +Directional ("be approaching" etc)

5) Processes: -Relative, +Directional ("be moving away from" etc)
The difference between 1) and 2) is obvious from what has beeb said above. The difference between 3) and 4) lies in 3) being measurable in absolute units (considering the direction of the movement to a certain extent), while the direction becomes more prominent in 4), where the distance cannot be generally measured in absolute units. 3) differs from 5) by being a result, while 5) is a process. 4) and 5) are both processes and depend on the direction of the movement, but while 4) is relative, unmeasurable simply in the physical distance on a bee-line, 5) the direction of the movement taken into consideration, can be measured absolutely. The predicates that contain the feature +Directional are at times ambiguous, or to be more exact, they can be used for descriptions of movements in opposite directions, depending on the expectation of the future movement:



A has approached the gate
(provided that we expect a further movement "round the corner")



A has approached the gate
(provided that we expect the movement to stop in the present position)

7. The new version of COMMENTATOR called COMMIE is meant to take into consideration the problems discussed here. To achieve a greater variation of comment three persons and two gates are generated. In order to assist test persons in their interpretations of the stimuli as movements, the present localization of the actors is completed by the last two states.

COMMENTATOR is primarily intended to be an instrument of linguistic research. "Making a computer talk meaningfully" demands an explicit description of the meanings of the words used in the program. (cf 5.) Beyond the level of the sentence two additional important demands have to be met: The program ought to curb repetitions of redundant information to the "human level", where "unnecessary" repetitions do occur but are chiefly limited to an occasional reconfirmation of validity of an earlier proposition that may have already slipped from the speaker's/listener's memory, or where the repetitions are used to convey difficult, complex information from another point of view in the interest of promoting understanding. (cf 3.) Secondly, the program has to cope with a far from simple and easy to understand human language better. Beside the purely theoretical aspects of COMMENTATOR there are many more practical ramifications on the horizon. It can help us to simulate, understand and cope with various disorders in speech production. Another application would be in language teaching. An explicit, better understanding of

language typology based on the comparison of how certain mental concepts are conveyed via different languages would be of a great help. And, of course, an "intelligent" machine verbally commenting variouns processes could substitute and/or complement various traditional dials and screens. An automatic radar surveillance commenting on changes of interest would be one of the most obvious applications of a future, more advanced version of the presently existing COMMENTATOR.

Bibliography

Badler, N.I. (1975): "Temporal Scene Analysis: Conceptual Descriptions of Object Movements", Techn. report no.80, Univ. of Toronto.

Bílý, M. (1981): "Intransentential pronominalization and Functional Sentence Perspective (in Czech, Russian, and English)", Lund Slavonic Monographs 1, Lund 1981.

Okada, N. (1980): "conceptual Taxonomy of Japanese Verbs and Sentence Production from Picture Pattern Sequences", Oita University,Japan

Sigurd, B. (1980): "COMMENTATOR. A Computer System Simulating Verbal Behaviour", Dept. of Linguistics, Univ. of Lund, mimeographed.

Viberg, Å. (1981): "En typologisk undersökning av perceptionsverben som ett semantisk fält", SSM Report 8, Dept. of Linguistics, Univ. of Stockholm.

Gregers Koch
Datalogisk Institut, Københavns Universitet

EN PROBLEMORIENTERET PROGRAMMELUDVIKLINGSMETODE I LINGVISTISK DATABEHANDLING

1. Indledning

Fortolkning af prædikatkalkyle som et programmeringssprog udgør en ny og lovende datalogisk metode, som ofte kaldes logikprogrammering. Prædikatlogiske notationer kan betragtes som højniveau, menneskevenlige programmeringssprog som kan anvendes til praktisk programmering såvel som til teoretiske undersøgelser. Specielt i forbindelse med datalingvistiske problemstillinger synes metoden lovende. Dette kommer skriftet her nærmere ind på, og desuden diskuteres nogle forsøg på udvidelse af metoden.

Prædikatkalkyle synes at være af stadigt stigende interesse for datamatisk orienterede lingvister [Charniak og Wilks 76]. Samme tendens synes at gøre sig gældende inden for kunstig intelligens [Nilsson 80].

Med fremkomsten af logikprogrammeringssprog som Prolog [Bowen 79] kan man se nye perspektiver i denne udvikling, tildels på grund af muligheden for effektiv udførelse af inferenser som nødvendigvis knytter sig til sådanne systemer.

2. Metoden

Definitte klausuler (også kaldet Hornklausuler) [Colmerauer 78, Kowalski 74, 79, Mayoh 80] er formler af formen

$$C_0 \leftarrow C_1, C_2, \ldots, C_n$$

hvor alle Cerne er prædikatudtryk der indeholder

- variable $X, Y, Z, \ldots, X_1, Y_1, Z_1, \ldots$
- konstanter
- funktionsnavne.

Idet universel kvantificering er underforstået, kan en sådan formel betragtes som ækvivalent til prædikatkalkyleformlen

$$\forall X_1, X_2, \ldots, X_m: ((C_1 \wedge C_2 \wedge \ldots \wedge C_n) \rightarrow C_0)$$

hvor Xerne netop udgør sættet af samtlige variable i Cerne. Specielt kan betingelserne være tomme ($n = 0$) svarende til simple påstande, eller konklusionen C_0 kan mangle svarende til en negation (også kaldet en målklausul), eller begge dele

svarende til den tomme klausul eller umulige påstand.

En kontekstfri grammatiks produktionsregler af formen

$$\text{Nonterminal} \rightarrow B_1\ B_2\ \ldots\ B_n$$

kan omformes til følgende formel fra første ordens prædikatkalkyle

$$\forall S_0,S_1,\ldots,S_n\colon ((B_1(S_0,S_1) \wedge B_2(S_1,S_2) \wedge \ldots \wedge B_n(S_{n-1},S_n)) \rightarrow \text{Nonterminal}\ (S_0,S_n))$$

med følgende mening:

- hele teksten fra position S_0 til position S_n kan fortolkes som et objekt tilhørende kategorien Nonterminal, såfremt
- teksten fra position S_0 til position S_1 kan fortolkes som et B_1-objekt, og
- teksten fra position S_1 til position S_2 kan fortolkes som et B_2-objekt, og

. . . . . .

- teksten fra position S_{n-1} til position S_n kan fortolkes som et B_n-objekt.

Et lille eksempel er følgende kontekstfrie grammatik:

$$\begin{aligned} &\text{Sentence} \rightarrow \underline{\text{the}}\ \text{Noun Verb} \\ &\text{Noun} \rightarrow \underline{\text{woman}} \\ &\text{Verb} \rightarrow \underline{\text{lives}} \\ &\text{Verb} \rightarrow \underline{\text{smells}} \end{aligned} \qquad (1)$$

som kan omformes til følgende prædikatlogiske formler:

$$\begin{aligned} &\forall S_0,S_1,S_2,S_3((C(\underline{\text{the}},S_0,S_1) \wedge \text{Noun}(S_1,S_2) \wedge \text{Verb}(S_2,S_3)) \rightarrow \text{Sentence}(S_0,S_3)) \\ &\forall S_0,S_1 : (C(\underline{\text{woman}},S_0,S_1) \rightarrow \text{Noun}(S_0,S_1)) \\ &\forall S_0,S_1 : (C(\underline{\text{lives}},S_0,S_1) \rightarrow \text{Verb}(S_0,S_1)) \\ &\forall S_0,S_1 : (C(\underline{\text{smells}},S_0,S_1) \rightarrow \text{Verb}(S_0,S_1)) \end{aligned} \qquad (2)$$

Hvis vi ønsker at se om

the	woman	smells
1	2	3 4

er en sætning fra den lille grammatik kan vi tilføje følgende påstande

$$C(\underline{\text{the}},1,2)$$
$$C(\underline{\text{woman}},2,3)$$
$$C(\underline{\text{smells}},3,4).$$

Problemet er nu om systemet er konsistent, og om det er muligt at deducere Sentence (1,4) som et teorem inden for systemet.

Den lille grammatik (1) kan også udtrykkes som definitte klau-

suler i stil med [Pereira & Warren 80]:

$$\begin{aligned}
&\text{Sentence}(S_0,S_1) \leftarrow C(\underline{\text{the}},S_0,S_1),\ \text{Noun}(S_1,S_2),\ \text{Verb}(S_2,S_3)\\
&\text{Noun}(S_0,S_1) \leftarrow C(\underline{\text{woman}},S_0,S_1)\\
&\text{Verb}(S_0,S_1) \leftarrow C(\underline{\text{lives}},S_0,S_1)\\
&\text{Verb}(S_0,S_1) \leftarrow C(\underline{\text{smells}},S_0,S_1)\ .
\end{aligned} \tag{3}$$

Som et lidt større eksempel i samme retning kan vi kigge på syntaksanalyse i henhold til følgende lille grammatik

$$\begin{aligned}
&\text{S} \rightarrow \text{NP VP [ADVP]}\ .\\
&\text{NP} \rightarrow \text{[DET] ADJ* N}\ .\\
&\text{NP} \rightarrow \text{PRON}\ .\\
&\text{ADVP} \rightarrow \text{PREPP}\ .\\
&\text{ADVP} \rightarrow \text{ADV}\ .\\
&\text{PREPP} \rightarrow \text{PREP NP}\ .\\
&\text{VP} \rightarrow \text{V [NP] [ADVP]}\ .
\end{aligned} \tag{4}$$

Denne grammatik omformes til definitte klausuler ved simplificering (som her vil sige eliminering af valgfrie elementer [...] samt repetitive elementer ...*) samt ved tilføjelse af positionsangivelser (x,y,z,w):

$$\begin{aligned}
&\text{S}(x,y) \leftarrow \text{NP}(x,z),\ \text{VP}(z,w),\ \text{ADVP}(w,y)\\
&\text{S}(x,y) \leftarrow \text{NP}(x,z),\ \text{VP}(z,y)\\
&\text{NP}(x,y) \leftarrow \text{DET}(x,z),\ \text{ADJLIST}(z,w),\ \text{N}(w,y)\\
&\text{NP}(x,y) \leftarrow \text{ADJLIST}(x,z),\ \text{N}(z,y)\\
&\text{NP}(x,y) \leftarrow \text{PRON}(x,y)\\
&\text{ADJLIST}(x,y) \leftarrow\\
&\text{ADJLIST}(x,y) \leftarrow \text{ADJ}(x,z),\ \text{ADJLIST}(z,y)\\
&\text{ADVP}(x,y) \leftarrow \text{PREPP}(x,y)\\
&\text{ADVP}(x,y) \leftarrow \text{ADV}(x,y)\\
&\text{PREPP}(x,y) \leftarrow \text{PREP}(x,z),\ \text{NP}(z,y)\\
&\text{VP}(x,y) \leftarrow \text{V}(x,z),\ \text{NP}(z,w),\ \text{ADVP}(w,y)\\
&\text{VP}(x,y) \leftarrow \text{V}(x,z),\ \text{NP}(z,y)\\
&\text{VP}(x,y) \leftarrow \text{V}(x,z),\ \text{ADVP}(z,y)\\
&\text{VP}(x,y) \leftarrow \text{V}(x,y)\ .
\end{aligned} \tag{5}$$

En inddatastreng som "De kommer på skadestuen" kan analyseres ved tilføjelse af følgende leksikalinformation

```
PRON(x,y)   ←  C(de,x,y)
V(x,y)      ←  C(kommer,x,y)
PREP(x,y)   ←  C(på,x,y)
N(x,y)      ←  C(skadestuen,x,y)
C(de,1,2)   ←                                    (6)
C(kommer,2,3)    ←
C(på,3,4)        ←
C(skadestuen,4,5) ←
←S(1,5) .
```

Bemærk at vi intetsteds specificerer hvilken analysealgoritme der ønskes anvendt. Vi specificerer kun problemet, så finder systemet selv ud af, hvordan problemet skal håndteres.

3. Kasussystemer

Sagt ultrakort agiteres der her for en datalogisk metode som går ud på at udsætte datalingvistiske problemer for en datamatisk behandling som om de var logiske problemer, og der søges argumenteret for det fordelagtige i denne metode fra et datalogisk synspunkt.

Påstanden er således at så at sige enhver datalingvistisk teori eller strategi ville profitere af at benytte denne metode. Som eksempler har jeg beskæftiget mig med Schanks "Conceptual Dependency" [Schank 75] og Parker-Rhodes' "Inferential Semantics" [Parker-Rhodes 78, Jørgensen 80].

Fremstillingen her ligger nærmest Parker-Rhodes, medens fremstillingen i [Koch 80/16] har flere lighedspunkter med Schanks teorier.

Det må understreges at fremstillingen her kun skal ses som et eksempel der belyser mulighederne ved at anvende denne metode til realisering af givne datalingvistiske teorier. (Således udelades her flere aspekter bl.a. tempusangivelser og numerusangivelser).

Da begge forfattere vedkender sig en vis gæld til [Fillmore 68], kan disse to teorier med nogen ret betragtes som kasussystemer.

Lad os først behandle en række små eksempler fra [Schank 75]:

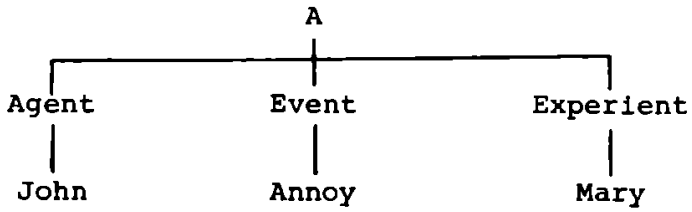
Eksempel 1

John annoyed Mary.

Den forudsatte leksikalske information kan være

(Annoy Agent (Experient)).

Et rimeligt syntakstræ kan være



I så fald kan det forventede resultat af syntaksanalysen være følgende listestruktur

[A [Agent John]
[Event Annoy]
[Experient Mary]].

Et rimeligt resultat af oversættelsen kan være

Event(A,Annoy)
Agent(A,John)
Experient(A,Mary). □

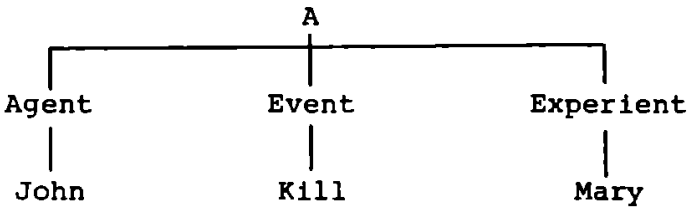
Eksempel 2

John killed Mary.

Den forudsatte leksikalinformation kan være

(Kill Agent (Experient) (by Instrument)) .

Et rimeligt syntakstræ kan være



Det forventede resultat af syntaksanalysen kan være

[A [Agent John]
[Event Kill]
[Experient Mary]].

Et rimeligt resultat af oversættelsen kan være

Event(A,Kill)
Agent(A,John)
Experient(A,Mary). □

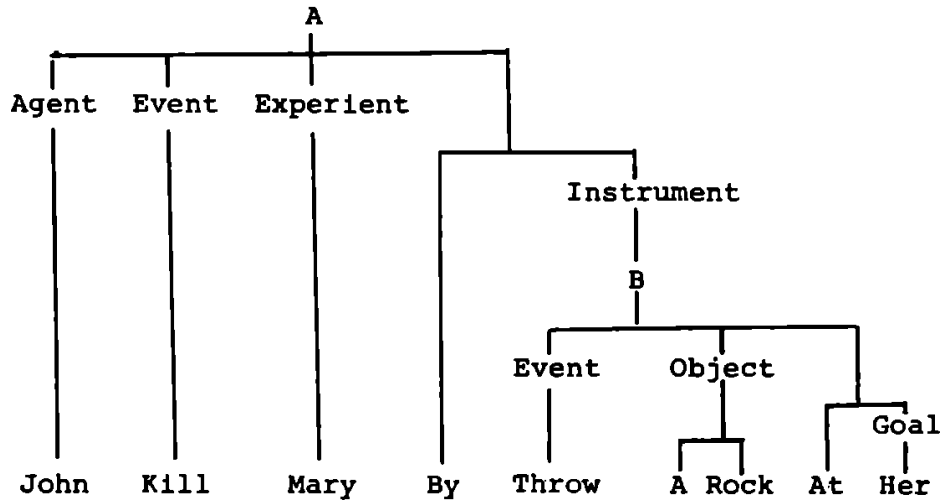
Eksempel 3

John killed Mary by throwing a rock at her.

Den forudsatte leksikalske information kan være

(Throw Agent (Object) (at Goal))
(Mary Person Female)
(John Person Male).

Et rimeligt syntakstræ kan være



Det forventede resultat af syntaksanalysen kan være

```
[A [Agent John]
   [Event Kill]
   [Experient Mary]
   [By [Instrument [B [Event Throw]
                      [Object [A Rock]]
                      [At [Goal Her]]]]]].
```

Et rimeligt resultat af oversættelsen kan være

```
Event(A,Kill)
Agent(A,John)
Experient(A,Mary)
Instrument(A,B)
Event(B,Throw)
Object(B,Rock)
Goal(B,Mary)
Agent(B,John) .
```

□

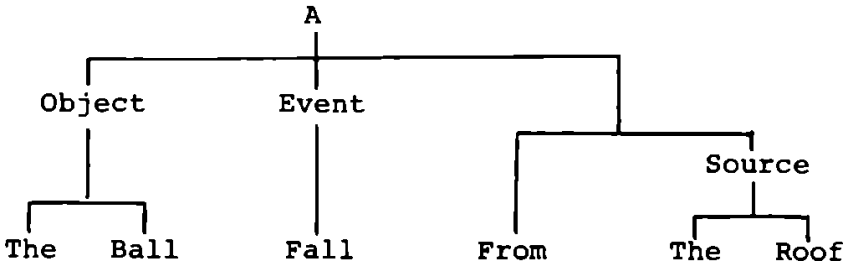
Eksempel 4

The ball fell from the roof.

Den forudsatte leksikalske information kan være

(Fall Object (from Source)).

Et rimeligt syntakstræ kan være



Et rimeligt resultat af oversættelsen kan være

Event(A,Fall)
Object(A,Ball)
Source(A,Roof).

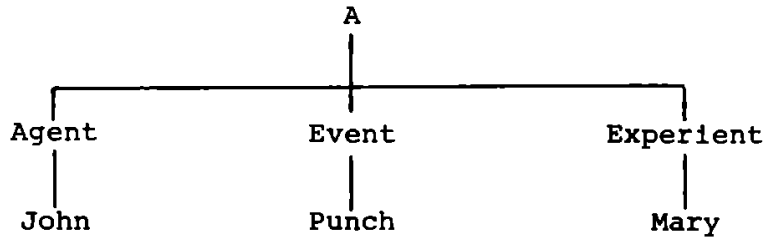
Eksempel 5

John punched Mary.

Den forudsatte leksikalske information kan være

(Punch Agent (Experient)).

Et rimeligt syntakstræ kan være



Et rimeligt resultat af oversættelsen kan være

Event(A,Punch)
Agent(A,John)
Experient(A,Mary).

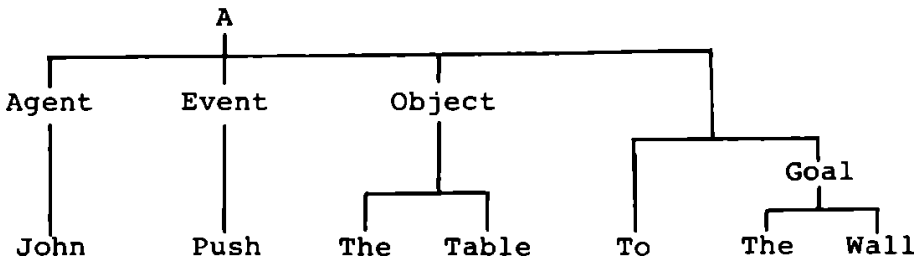
Eksempel 6

John pushed the table to the wall.

Den forudsatte leksikalske information kan være

(Push Agent (Object) (to Goal))

Det forventede resultat af syntaksanalysen kan være



Et rimeligt resultat af oversættelsen kan være

Event(A,Push)
Agent(A,John)
Object(A,Table)
Goal(A,Wall). □

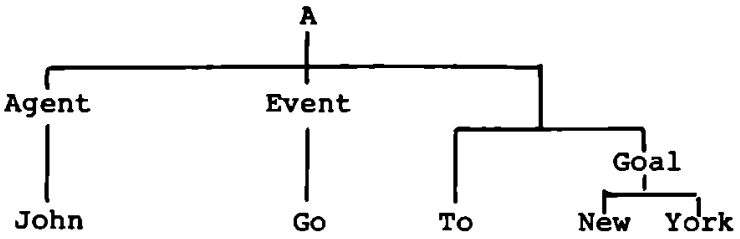
Eksempel 7

John went to New York.

Den forudsatte leksikalske information kan være

(Go Agent (from Source) (to Goal)) .

Et rimeligt syntakstræ kan være



Et rimeligt resultat af oversættelsen kan være

Event(A,Go)
Agent(A,John)
Goal(A,New York) . □

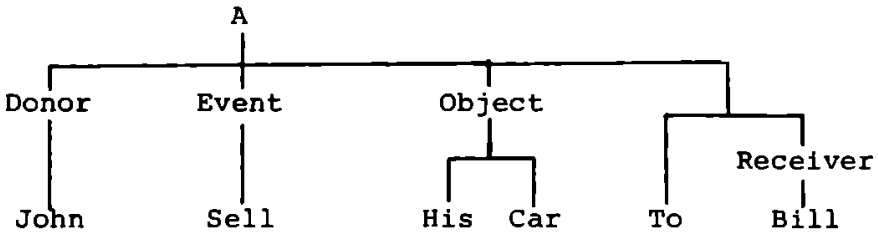
Eksempel 8

John sold his car to Bill.

Den forudsatte leksikalske information kan være

(Sell { ((Receiver) Object) / (Object) (to Receiver) }) .

Et rimeligt syntakstræ kan være



Et rimeligt resultat af oversættelsen kan være

Event(A,Sell)
Donor(A,John)
Object(A,Car)
Receiver(A,Bill). □

På baggrund af sådanne eksempler søger vi nu at realisere oversættelsen. Generering af syntakstræ er ganske nemt ud fra metoden i det foregående afsnit med brug af en grammatik som beskriver eksemplerne. Denne kan for eksempel udtrykkes noget i denne retning:

Event → { Agent (Timeadv) Event\A | NP is Adj | NP have NP | Donor (Timeadv) Event\D | Object (Timeadv) Event\O }

Event\A → { EventAE (Experient) | EventAEBYI (Experient) (by Instrument) | EventAOATG (Object) ({ at | to } Goal) | EventAFSTG (from Source) (to Goal) | EventAO (Object) | EventA }

Event\D → { EventDRO ((Receiver) Object) | EventDRO (Object) (to Receiver) | EventDRD (Receiver) (Datum) | EventDRD (Datum) (to Receiver) }

Event\O → EventOFS (from Source)

{ Experient | Object | Source } → NP

Adj → short

Timeadv → often

etc.

Systemet (3) gav kun svaret ja/nej. For at få et resultat af analysen, må man tilføje ét eller flere ekstra argumenter og i disse resultatargumenter angive hvordan konklusionens resultatarguments værdi skal være.

Tilbage står at realisere oversættelsen fra syntakstræ på listestrukturform til sættet af prædikater. Dette hverv kan udføres omtrent således:

```
Make((n . (x . y),r)← M((x . y),n,r)
M(([u w] . z),n, ([u "<" n "," w ">"] . z1))
     ← Member(u,Cases), M(z,n,z1)
M ([p [u [(n1 . w)]]] . z),n,
       ([u "<" n "," n1 ">"] . z1))
     ← Member(p,Prepositions), Member(u,Cases), M(z,n1,z1).
```

Lad os nu se på endnu et par eksempler (denne gang taget fra [Parker-Rhodes 78]):

Eksempel 9

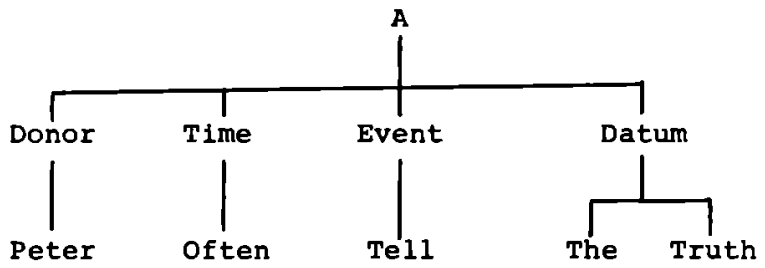
Peter often tells the truth.

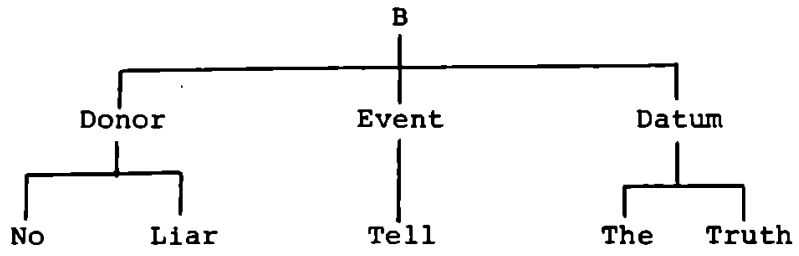
No liars tell the truth.

Den forudsatte leksikalske information kan være noget i denne retning

(Tell Donor { (Datum) (to Receiver) / (Receiver) Datum })

Syntakstræerne kan være





Resultatet ved algoritmen skulle så blive noget i denne retning

Event(A,Tell)

Donor(A,Peter)

Time(A.Often)

Datum(A,Truth)

← Event(y,Tell),Donor(y,x),Datum(y,Truth),Isa(x,Liar).

□

Eksempel 10

Peter eats garlic.

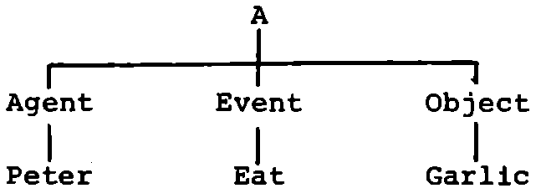
Everyone who eats garlic smells.

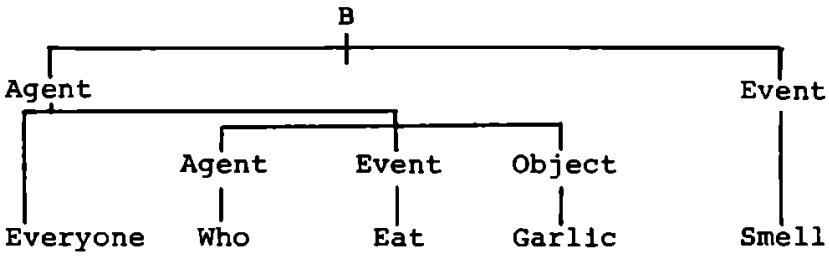
Den forudsatte leksikalske information kan være

(Eat Agent (Object))

(Smell Agent).

Syntakstræerne kan være





Resultatet bliver så noget i denne retning

Event(A,Eat)

Agent(A,Peter)

Object(A,Garlic)

Event(F(x,y),Smell) ← Event(y,Eat),Object(y,Garlic),
Agent(y,x)

Agent(F(x,y),x) ← Event(Y,Touch),Object(y,Garlic),
Agent(y,x).

□

Fordelene fra et datalogisk synspunkt er blandt andre af følgende art:

- Kompleksitetsteoretisk: Simple grammatikker (for eksempel af type LL1) giver effektiv (lineær) analyse, og grammatikker som "næsten" har disse egenskaber giver også forholdsvis effektiv analyse.
- Brugervenlighed: Den lingvistiske bruger skal kun udtrykke egentlige datalingvistiske relationer, endda i en notation som ligger tæt på de normalt anvendte.
- Systemkonstruktionsmæssigt: Som datastyret programmel benyttes den samme algoritme (inferensalgoritmen) hver gang. Man kan sige at der kun kræves en (ganske vist temmelig omstændelig) problemspecifikation. Såsnart problemet er logisk entydigt, kan programmellet overtage behandlingen. I denne forstand kan et sådant system betragtes som et problemorienteret system, og denne metode at konstruere systemer på kan betragtes som en problemorienteret programmeludviklingsmetode.

Vi arbejder på at benytte metoden her i forbindelse med datamatformidlet undervisning.

Vi er også ved at udvide systemer af denne art til at omfatte nogle intensionelle logiske systemer à la [Montague 74a] og [Koch 79].

4. Databaseforespørgsler

En metode til håndtering af databaseforespørgsler i humansproglige vendinger går ud på undervejs at oversætte forespørgslen til definitte klausuler.

Med samtlige oplysninger fra databasen formuleret som definitte klausuler ville svaret kunne genereres inden for det deduktive system i kraft af den indbyggede deduktionsmekanisme.

Fra et databasesynspunkt ville denne fremgangsmåde imidlertid være utilfredsstillende af effektivitetshensyn. Langt bedre ville det være at oversætte de definitte klausuler til et egentligt forespørgselssprog for et databasesystem. Som en realistisk mulighed har vi især undersøgt sproget QUEL tilhørende systemet INGRES. [Stonebraker et al 76].

Her eksemplificeres med en simpel forespørgsel til QUEL. En simpel database for "The Happy Valley Food Cooperative" består af tre databaserelationer

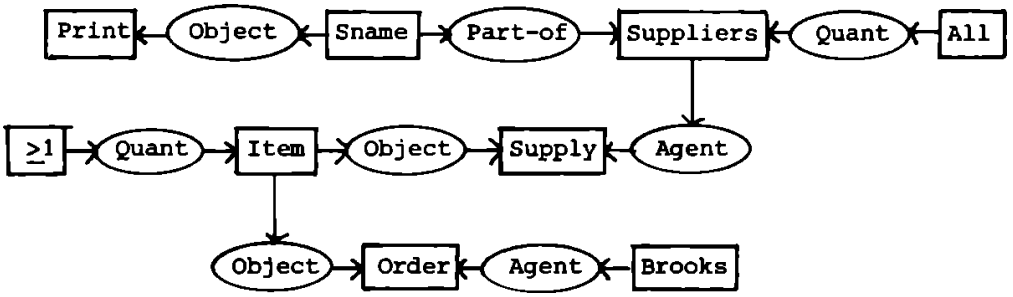
MEMBERS(NAME, ADDRESS, BALANCE)

ORDERS(NAME, ITEM, QUANTITY)

SUPPLIERS(SNAME, SADDRESS, ITEM, PRICE)

således at alle medlemmer har en adresse og en saldo, nogle medlemmer har bestilt forskellige varer i bestemte mængder, og nogle varer kan leveres af leverandører fra deres forretningsadresse til bestemte priser [Ullman 80].

Forespørgslen "Udskriv navnene på alle leverandører, som leverer mindst een vare bestilt af Brooks" kan i et logikprogrammeringssystem analyseres til en konceptuel graf [Sowa 76, 79, Pedersen 78] af følgende form



som igen af en relativt simpel algoritme kan transformeres videre til et logisk program af følgende form

```
Event(A(y),Print)       ← Isa(y,Suppliers),
                          Event(E(y),Order),
                          Agent(E(y),Brooks),
                          Object(E(y),Item(y))
Object(A(y),Sname(y)) ← Isa(y,Suppliers),
                          Event(E(y),Order),
                          Agent(E(y),Brooks),
                          Object(E(y),Item(y)).
```

Dette logiske program kan automatisk oversættes videre til et program i et egentligt databaseforespørgselssprog som QUEL med følgende resultat

```
RANGE OF y IS Suppliers
RANGE OF z IS Orders
RETRIEVE y . Sname
WHERE z . Name = Brooks  ∧
      z . Item = y . Item
```

som er et udmærket program til at besvare det stillede spørgsmål. [Jørgensen & Koch 81].

5. Udvidelser

Det simple databaseeksempel i foregående afsnit gik godt med brug af definitte klausuler. Men vi kunne også kigge på et eksempel som går ud på at undersøge om der findes medlemmer af Alpinistklubben som er bjergbestigere men ikke skisportsmænd, idet følgende vides:

Tony, Mike og John er i Alpinistklubben. Ethvert medlem af Alpinistklubben er skisportsmand eller bjergbestiger. Ingen bjergbestiger kan lide regn, og alle skisportsmænd elsker sne. Mike hader alt det Tony holder af og holder af alt det Tony hader. Tony kan godt lide regn og sne.

I sædvanlig prædikatkalkyle kan vi skrive:

$$\begin{aligned}
&\text{Tony} \in \text{Alpinists}\\
&\text{Mike} \in \text{Alpinists}\\
&\text{John} \in \text{Alpinists}\\
&\forall x \in \text{Alpinists}\ [\text{Skier}(x) \vee \text{Climber}(x)]\\
&\forall x\ [\text{Climber}(x) \rightarrow \text{Dislikes}(x,\text{Rain})]\\
&\forall x\ [\text{Skier}\ (x) \rightarrow \text{Likes}(x,\text{Snow})]
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\forall y\ [\text{Likes}(\text{Tony},y) \rightarrow \text{Dislikes}(\text{Mike},y)]\\
&\forall y\ [\text{Dislikes}(\text{Tony},y) \rightarrow \text{Likes}(\text{Mike},y)]\\
&\text{Likes}(\text{Tony},\text{Rain}) \wedge \text{Likes}(\text{Tony},\text{Snow})\\
&\forall x \in \text{Alpinists}\ [\text{Climber}(x) \wedge \neg\, \text{Skier}(x) \rightarrow \text{Print}(x)]
\end{aligned} \tag{8}$$

(7) og (8) giver os problemer fordi negation er nødvendig.

De vanskeligheder vi her er løbet ind i beror væsentligt på at kun definitte klausuler er tilladte. Definitte klausuler er åbenbart for restriktiv en notation, og specielt at negation mangler synes at volde problemer. Så metoden her skulle helst generaliseres ud over definitte klausuler.

Den mest oplagte udvidelse er nok følgende:

Vi kan supplere hvert prædikatnavn P med et tilsvarende navn NP som symboliserer benægtelsen af prædikatet P . Altså vi forsøger så at sige at indføre negationen bag om ryggen på systemet (at programmere den ind i systemet). For at kunne udnytte sammenhængen mellem P og NP bliver vi så også nødt til at mangedoble hver regel

$$A_1(\underline{x}), \ldots, A_n(\underline{x}) \leftarrow B\ (\underline{x}), \ldots, B_m(\underline{x})$$

ved omskrivningen

$$A_1(\underline{x}) \vee \ldots \vee A_n(\underline{x}) \vee \neg\, B_1(\underline{x}) \vee \ldots \vee \neg\, B_m(\underline{x})$$

eller

$$\neg NA_1(\underline{x}) \vee \ldots \vee \neg\, NA_{i-1}(\underline{x}) \vee A_i(\underline{x}) \vee \neg\, NA_{i+1}(\underline{x}) \vee \ldots \vee \neg\, NA_n(\underline{x})$$

$$\vee \neg\, B_1(\underline{x}) \vee \ldots \vee \neg\, B_m(\underline{x})$$

eller

$$A_i(\underline{x}) \leftarrow B_1(\underline{x}), \ldots, B_m(\underline{x}), NA_1(\underline{x}), \ldots, NA_{i-1}(\underline{x}), NA_{i+1}(\underline{x}),$$

$$\ldots, NA_n(\underline{x})$$

for hvert $i \in \{1, \ldots, n\}$.

Tilsvarende laves reglen

$$NB_j(\underline{x}) \leftarrow B_1(\underline{x}),\ldots,B_{j-1}(\underline{x}),B_{j+1}(\underline{x}),\ldots,B_m(\underline{x}),NA_1(\underline{x}),\ldots,NA_n(\underline{x})$$

for hvert $j \in \{1,\ldots, m\}$. Vi kan sige at vi udnævner hvert af de indgående prædikater til konklusion i en definit klausul. Endelig tilføjes reglen

$$\leftarrow p(\underline{x}),\ Np(\underline{x})$$

for hvert prædikat p .

Bruger vi denne metode på Alpinist-eksemplet fås følgende system som løser problemet:

```
Alpinist(Tony).
Alpinist(Mike).
Alpinist(John).
Dislikes(x,Rain) ← Climber(x).
Likes(x,Snow) ← Skier(x).        Nskier(x) ← Dislikes(x,Snow).
Dislikes(Mike,y) ← Likes(Tony,y).
Likes(Mike,y) ← Dislikes(Tony,y).
Likes(Tony,Rain).
Likes(Tony,Snow).
Skier(x) ← Alpinist(x), NClimber(x).
Climber(x) ← Alpinist(x), NSkier(x).
← Climber(x), NClimber(x).
← Skier(x), NSkier(x).
Print(x) ← Climber(x), NSkier(x).
```

Ulempen er at vi ender med at simulere traditionelle resolutionsstrategier med den deri liggende fare for ineffektivitet af såvel pladsmæssig som tidsmæssig art.

Altså den oplagte metode med at inkludere negationer i prædikatnavnene fører til en forholdsvis ineffektiv variant af resolutionsmetoden. I et kommende skrift [Koch 81] søges udviklet nogle alternative og mere begrænsede udvidelser af definitte klausuler, hvor disses effektive udførelse i det væsentlige synes bevaret.

6. Litteraturhenvisninger

Bowen, K.A. [1979]. Prolog, Proc. of the 1979 Annual Conf. ACM, Detroit, Michigan.

Charniak, E., Wilks, Y. (eds.) [1976]. Computational Semantics, pub. North-Holland.

Colmerauer, A. [1978]. Metamorphosis Grammars, in L. Bolc (ed.) Natural Language Communications with Computers, Springer, Berlin.

Fillmore, C. [1968]. The Case for Case, in Bach and Harms (eds.) Universals in Linguistics Theory, Holt, Rinehart and Winston, New York.

Jørgensen, P.H. [1980]. Inference and Semantics of Natural Language, master thesis, Institute of Datalogy, Copenhagen University.

Jørgensen, P.H., Koch, G. [1981]. Two New Methods of Natural Language Database Queries (in Danish), Proc. Norddata Conf., Copenhagen 1981, 227 - 232.

Koch, G. [1979]. Experimental Formalization of Danish. DIKU report 79/19 (in Danish), Institute of Datalogy, Copenhagen University.

Koch, G. [1980]. A Prolog Way of Representing Natural Language Fragments, DIKU report 80/16, Institute of Datalogy, Copenhagen University.

Koch, G. [1981]. Ulemper ved og udvidelser af definitte klausuler. Forthcoming DIKU report, Institute of Datalogy, Copenhagen University.

Kowalski, R. [1974]. Predicate Logic As Programming Language, Proc. IFIP 74, Stockholm.

Kowalski, R. [1979]. Logic for Problem Solving, New York, North-Holland, New York.

Mayoh, B.H. [1980]. The Meaning of Logical Programs, DAIMI PB-126, Aarhus University.

Montague, R. [1974a]. The Proper Treatment of Quantification in Ordinary English. [In Montague 74b].

Montague, R. [1974b]. Formal Philosophy, Yale University Press.

Nilsson, N.J. [1980]. Principles of Artificial Intelligence, Tioga Publ. Comp., California.

Parker-Rhodes, F. [1978]. Inferential Semantics, Harvester, Sussex, England.

Pedersen, G.S. [1978]. Conceptual Graphs I. DIKU report 78/9, Institute of Datalogy, Copenhagen University.

Pereira, F.C.N., Warren, D.H.D. [1980]. Definite Clause Grammars for Language Analysis - a Survey of the Formalism And a Comparison with Augmented Transition Networks. Artif. Intell. 13,3,231 - 278.

Schank, R. (ed.) [1975]. Conceptual Information Processing, North-Holland, Amsterdam.

Sowa, J.F. [1976]. Conceptual Graphs for a Database Interface. IBM Journ. Research. Devel. 20, 336 - 357.

Sowa, J.F. [1979]. Definitional Mechanism of Conceptual Graphs, in V. Claus et al. (eds.) Graph-grammars And Their Application to Computer Science And Biology, Springer, Berlin.

Ullman, J.D. [1980]. Principles of Database Systems. London.

Stonebraker, M. et al. [1976]. The Design and Implementation of INGRES. ACM Trans. on Database Systems 1,3, 189-222.

Tor Stålhane
RUNIT

LEKSIKALSK ANALYSE I MJUKE SYSTEM

0. INNLEDNING

Mjuke System er et RUNIT-prosjekt som blir finansiert av Norges Teknisk-Naturvitenskapelige Forskningsråd. Prosjektet har som mål å lette tilgange til EDB-systemer. Målgruppa består av to deler:

- folk som bare sporadisk bruker EDB
- folk som ofte bruker EDB, men som ofte bytter maskin. Dette gjelder særlig brukere som benytter datanett.

Hovedproblemet for begge disse brukergruppene er at de må huske en mengde nøkkelord og konvensjoner, så som plassering av punktum og komma o.l. Både nøkkelordene og konvensjonene varierer sterkt fra maskin til maskin.

Det er vår mening at dette problemet løses best ved å la brukerne benytte et subsett av norsk til kommunikasjon. Det systemet som skal stå mellom brukeren og EDB-systemet, har vi kalt et Mjukt System eller Mjukt Grensesnitt.

Dette systemet består av tre hoveddeler:

- leksikalanalysator
- syntaks- og semantikk-analysator
- system-grensesnitt

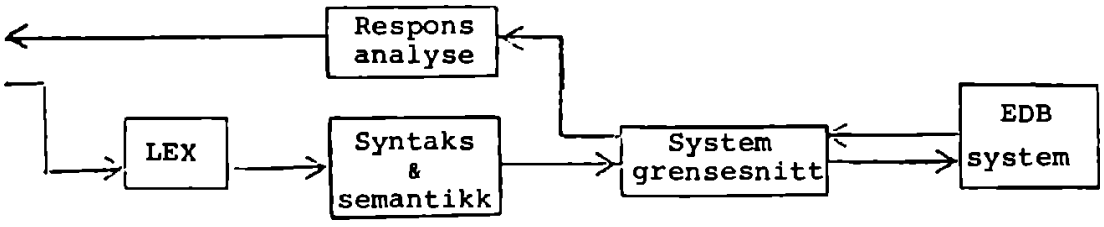


Fig. 1

1. VOKABULAR

Systemet har for øyeblikket et relativt lite vokabular, bare ca. 150 ord. Ordutvalget er bestemt ut fra ei rekke protokollforsøk (1). Andre undersøkelser som er gjort har konkludert med at et generelt vokabular på ca. 400 ord pluss et fagvokabular på ca. 100 ord er tilstrekkelig for de fleste anvendelser (2).

1.1 Ordlister i Mjuke System

Vi har valgt følgende retningslinjer for ordliste-strukturen i prosjektet:

- notasjonen skal være enkel å forstå
- det skal være enkelt å legge inn nye ord og nye skrivemåter for ord som allerede er i ordlista
- endringer og tillegg skal kunne gjøres med en vanlig tekst-editor

Ut fra disse betingelsene har vi valgt følgende løsning:

Vi har en symbolsk fil som inneholder hele ordlista i direkte lesbar form. Lista inneholder to typer informasjon:

- generelle bøyningsmønster, som man kan referere til seinere i lista
- ord, med tilhørende bøyninger, eller med referanse til ei generell bøyning

Et generelt bøyningsmønster har forma

navn = <serie av bøyningsregler>.

Hver bøyningsregel har følgende form:

$$(\langle\text{ending}\rangle\{(\langle\text{utinfo}\rangle\langle\text{ordklasse}\rangle\langle\text{ordform}\rangle)\}_1^*)$$

Eksempel

```
SUBST_1=(- (- SUBST UBEST_SING))
        (EN(- SUBST BEST_SING))
        (ER(- SUBST UBEST_PL))
        (ENE(- SUBST BEST_PL)).
```

Dette bøyningsmønsteret kan nå knyttes til et ord i ordlista. F.eks. ordet BOKSTAV, som følger denne reglen

```
BOKSTAV SUBST_1.
```

Et ord defineres i ordlista på én av to måter:

- Ved bruk av de generelle bøyningsmønstrene, f.eks.

  BOKSTAV SUBST_1.

- Ved å definere et bøyningsmønster spesielt for dette ordet. Dette skjer ved å sette

  <ord> <serie av bøyningsregler>.

  F.eks.:

```
NAVN (-(- SUBST UBEST_SING)
        (- SUBST UBEST_PL))
     (ET(- SUBST BEST_SING))
     (ENE(- SUBST BEST_PL)).
```

Den sistnevnte metoden benyttes også for å behandle uregelmessige verb. Dette vises lettest ved et eksempel:

```
ER   (-(VÆR VB PRES)).
VÆR  (E(- VB INF)) (-(- VB IMPT))
     (T(- VB PERF)).
VAR  (-(VÆR VB IMPERF)).
```

Vi benytter her ut-info delen til å gi rett informasjon videre i systemet. Dette blir også brukt dersom vi ønsker å bytte ut et ord med et annet, f.eks.:

```
PRINTER (-(LINJESKRIVER SUBST UBEST_SING))....
```

1.2 Ordlistas lagerstruktur

Internt i leksikal-analysatoren blir hvert ord lagra på følgende måte:

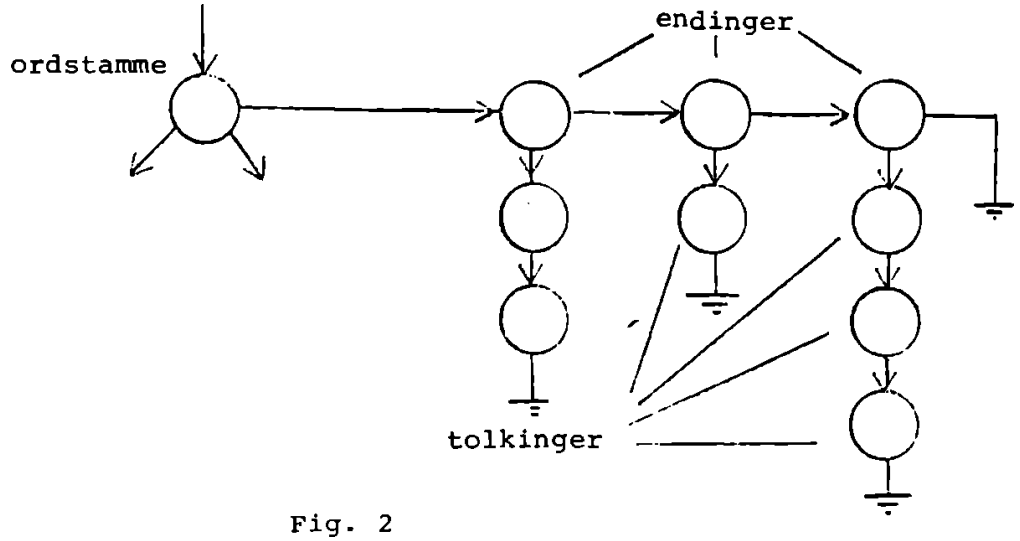


Fig. 2

Som ordstamme har vi valgt å bruker imperativsformen for verb og ubestemt form éntall for substantiver. Dette er valgt ut fra hensynet til en mest mulig kompakt struktur. Hadde vi f.eks. valgt infinitivsformen for verb ville vi vært nødt til ha ha ekstra innslag i lista for imperativsformen.

Eksempel på lagring:

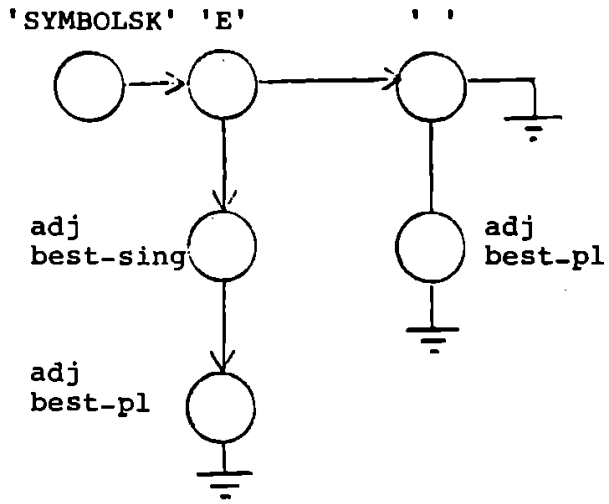


Fig. 3

Ordlista er delt opp i et sett av balanserte trær. Ett tre består av alle ord som begynner på 'A', ett tre består av alle ord som begynner på 'B' osv., til 'Å'.

Hvert tre er ordna etter tre kriterier:

- orda er sortert etter lengde før de settes inn i treet.
- hver ordstamme settes inn ved å leite fram et tomt lenkefelt på følgende måte:

  Viss ordet vi sammenligner med er lenger ute i alfabetet, følger vi høyre lenke, ellers følger vi den venstre. Dette fortsetter vi med til den lenka vi vil følge, er tom. Det nye ordet blir så hengt på der.

- treet er balansert for hver ordlengde (3)

Dette er gjort for å forenkle søkinga etter et ord og for å unngå å måtte gå igjennom treet flere ganger. Ved å sørge for at treet er balansert vil vi trenge gjennomsnittlig ca. $\frac{1}{3}\log_2(N)+1$ sammenligninger der N er antall ord i treet.

Uten balansering vil vi kunne risikere å måtte foreta N/2 sammenligninger. Det ferdige treet vil kunne se f.eks. slik ut

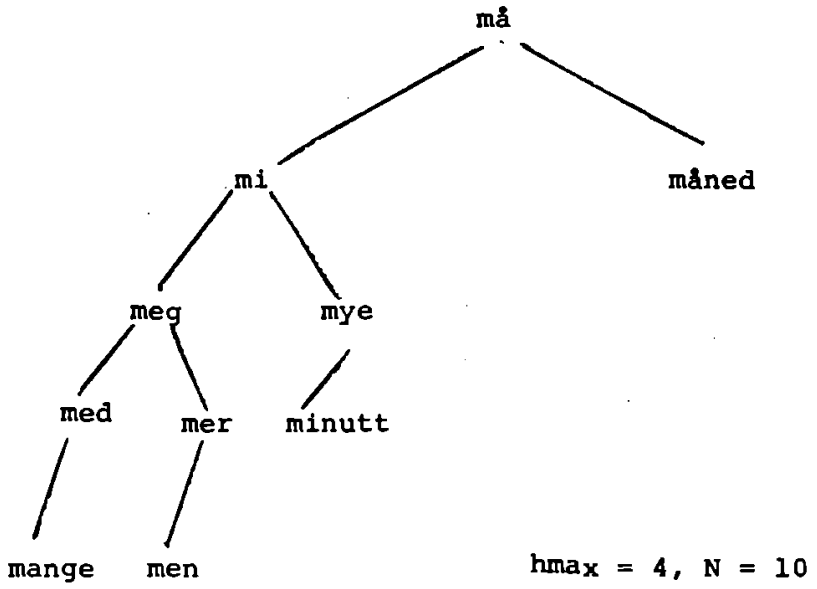


Fig. 4

1.3 Søkemetode

For å finne ut om et ord er i ordlista benytter vi følgende metode (4):

- Første bokstav viser hvilket tre vi skal søke i. Lengda av første ordstamme i treet viser hvor stor del av ordet vi må begynne med i søkinga. Resten av ordet blir behandla som ei ending.

- Deretter leter vi i treet til vi finner den aktuelle ordstammen eller vi finner en stamme som er lenger enn den vi har for øyeblikket.

  Dersom vi har kommet fram til en lengere ordstamme enn den vi søker med, forlenger vi stammen med neste bokstav i søkeordet. Endinga blir da forkorta tilsvarende.

  Viss det ikke er flere bokstaver igjen, finnes ikke søkeordet, ellers fortsetter vi søkinga til vi finner ordet, eller vi når slutten på treet.

- Når vi har funnet en ordstamme som passer, må vi sjekke om resten av ordet (endinga) finnes som ei lovlig ending til denne stammen. Dersom det er tilfelle, er alt ok og vi kan returnere den informasjonen vi har funnet.

  Dersom resten av ordet ikke er ei lovlig ending, må vi forlenge stammen og fortsette søkinga.

Et par enkle eksempler vil vise hvordan metoden fungerer i praksis:

1)

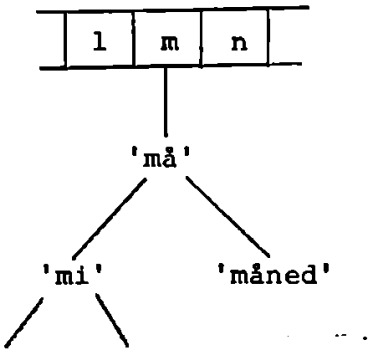


Fig. 5

Leter etter ordet 'måned'.

. begynner med 'må' - 'ned'

. finner 'må' først i treet, men denne stammen har ikke 'ned' som tillatt ending

. utvider til 'mån' - 'ed', men treet har ingen ordstammer langs denne greina med tre bokstaver. Det samme gjelder for 'måne'-- 'd'.

. først når vi utvider ordstammen til 'måned' - ' ' finner vi ordet og kan returnere

'måned subst ubest_sing'

2)

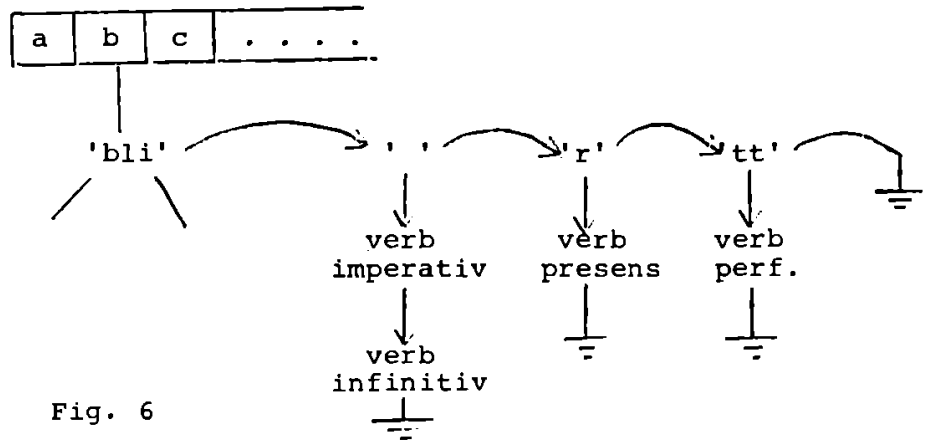


Fig. 6

Leter etter ordet 'blitt'.

Siden første ordstammen har tre bokstaver, starter vi med 'bli'-'tt'. Vi finner at 'bli' er en lovlig ordstamme og søker langs kjeda av lovlige endinger med 'tt'. Som vi ser i fig. 6, finner vi denne endinga sist i kjeda og returnerer derfor

'blitt bli verb perf.'

Som tidligere sagt under 1.1 blir uregelmessige verb og substantiver knytta til stammen via den delen vi kaller ut-info. Dersom vi f.eks. skulle leite eller 'blei' istedenfor 'blitt' ville vi ha funnet dette som et eget innslag i tabellen og fått returnert

'blei bli verb imperf'

2. HANDTERING AV STAVEFEIL

Eksperimenter og innsamla erfaringsdata indikerer at den alt overveiende delen av skrive/stave-feil tilhører en av følgende kategorier (5):

- én ekstra bokstav
- én manglende bokstav
- én feil bokstav

I tillegg har det vist seg at feilraten for alle disse feiltypene er langt større "inne i" ordet enn i starten og slutten. Disse forholda har gjort at vi har valgt følgende metode for handtering av stavefeil (6):

- for hvert ord som er lagra i ordlista blir det laga en bitvektor som forteller hvilke bostaver som finnes i ordet.

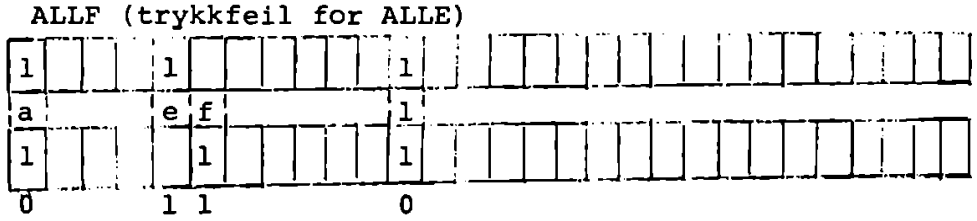
  Eks.

  ALLE

  I tillegg tar vi vare på ordlengden. Den samme informasjonen blir generert for et ord som ikke er blitt funnet i ordlista etter et vanlig gjennomløp.

- for hvert ord i det treet vi gjennomløper, sjekker vi så differansen mellom ordet i ordlista og det ordet vi tror er et feilstava ord. Denne differansen kommer fram ved å sammenligne de to bit-vektorene og summere alle bitene som ikke finnes i begge orda. Til dette tallet legger vi så absoluttverdien av forskjellen i ordlengdene.

  Eks.:



  Differansen blir 2, idet orda har samme lengde.

- for at et ord skal bli godtatt som en feilstaving av et annet, må differansen være mindre enn en forhandssatt terskel, dmax.

  Det er lett å forvisse seg om at de tre vanlige formene for skrivefeil alle vil gi en differanse på maksimum 2. Vi har derfor valgt dmax lik 2, unntatt når ordet har færre enn fire bokstaver. I de tilfellene benytter vi dmax lik 1.

  Metoden har så langt vist seg relativt tilfredsstillende. Det eneste problemet har vært at metoden ofte kommer opp med alt for mange tolkningsalternativer. Dette problemet vil kunne øke etter hvert som vi får flere ord i ordlista.

3. KOBLING TIL PROLOG

Den syntaktiske og semantiske analysen av inndata foregår i et PROLOG-program (7).

PROLOG benytter listestruktur til å representere inn- og ut-data. I Mjuke System kommer data inn som fri-format tekst. Leksikalanalysatoren deler opp teksten i ord, tall og skille-tegn og henger på en del informasjon om hver tekstenhet. Denne informasjonen har følgende format:

$$(\langle \text{inn-data} \rangle \left\{ (\langle \text{indikator} \rangle \langle \text{stamme} \rangle \langle \text{tolking} \rangle) \right\}_1^*)$$

indikator kan være:

- ! : ordet er funnet i ordlista
- ? : ordet er ikke funnet i ordlista, men vi har funnet ei rimelig tolking ut fra feilstavingsalgoritma

tolking ::= <ordklasse> <bøyning>

Dersom bøyning ikke er relevant, blir den erstatta med '-'.

Noen eksempler vil vise hvordan denne blir brukt:

"Hvilke filer har jeg?"

vil gi:

```
((HVILKE (!  HVILKE PRON   PL        ))
 (FILER  (!  FIL    SUBST  UBEST-PL ))
 (HAR    (!  HA     VB     PRES     ))
 (JEG    (!  JEG    PRON     -      ))
 (?      (!  "?"    TEGN   SPØRSMÅL ))
```

"Filler" (trykkfeil for Filer) vil gi:

```
 (FILLER (?  FILLER IDENT    -       )
         (?  FLER   ADJ     PL       )
         (?  FINN   VB      PRES     )
         (?  FEIL   SUBST   BEST-SING)
         (?  FIL    SUBST   BEST-PL  )
         (?  FIL    SUBST   UBEST-PL ))
```

4. OPPSUMMERING

Det er vår mening at den implementerte løsninga tilfredsstiller alle viktige krav til en leksikal-analysator for et naturlig språk:

. Det er lett å legge inn nye ord
. Det er lett å legge på nye tolkninger av ord som allerede finnes
. Det er lett å legge inn nye former og skrivemåter
. Systemet er rimelig tolerant overfor de vanligste skrivefeil

Det videre arbeidet med leksikalanalysen i Mjuke System vil konsentrere seg om to områder:

- utvidelse av vokabularet etter hvert som vi får mer bruker-erfaring

- forsøk på å dele vokabularet i én generell norsk-del som er brukeruavhengig, og en fagterm-del som må byttes ut når systemet skal handtere andre fagfelt/universer.

5. REFERANSER

(1) Amble, Tore
Mjuke System, Arbeidsnotat nr. 2, RUNIT 1981

(2) Kelly, Michael
Limited Vocabulary Natural Language Dialogue
Int. J. Man-Machine Studies, 1977 no. 9

(3) Wirth, Niklaus
Algorithms + Datastructures = Programs
Prentice-Hall, Englewood Cliffs NJ, 1976

(4) Stålhane, Tor
Mjuke System, Arbeidsnotat nr. 5, RUNIT 1981

(5) Morgan, H.L.
Spelling Error Correction in Systems Programs
Comm ACM 13, 1970

(6) Tenczar, P.
Spelling, Word and Concept Recognition
University of Illinois, Urbana, Ill. 1972

(7) Amble, Tore
Introduction to Logic Programming
RUNIT, 1981

Tore Amble
Regnesentret
Trondheim

Å VÆRE ELLER Å HA, DET ER SPØRSMÅLET

1. MYKE SYSTEMER

Naturlig språk brukes vanligvis når mennesker kommuniserer med hverandre, og er velegnet til dette formål. I moderne tid kommer stadig flere mennesker i kontakt med datamaskiner, enten de liker det eller ikke, og måten kommunikasjonen foregår vil være viktig for effektiviteten og trivselen til de som har med dem å gjøre.

For spesialoppgaver som skal utføres av en fast stab som har fått opplæring, vil spesielle styrespråk være tilstrekkelig og i noen tilfeller å foretrekke. (Det har bl.a. vært hevdet at opplæring i slike styrespråk gir utøverne en slags yrkes-identitet.)

En utvikling i retning av naturlig språk vil være ønskelig i situasjoner der

a) bruken er sporadisk
b) brukerne har kontakt med flere datamaskiner, f.eks. via datanett
c) brukerne mangler opplæring i spesielle systemer

Hittil har situasjonen ofte vært den at et datasystem er utviklet for å utføre en oppgave. Til slutt har man laget et bruker-grensesnitt for å ta seg en kommunikasjon med brukerne. Et slikt brukergrensesnitt vil ofte utgjøre en mindre del av det totale system.

Etter som kravene til brukervennlighet stiger (f.eks. via arbeidsmiljøloven) vil vi kunne komme i den situasjon at en vesentlig del av et datasystem utgjøres av brukergrensesnitt.

Myke systemer

For å møte utfordringer fra brukerne og gjøre det lettere å lage brukergrensesnitt, har Regnesentret ved Universitetet i Trondheim (RUNIT) startet et forskningsprosjekt "Mjuke System" med støtte fra NTNF. Formålet er å lage et generelt bruker-grensesnitt som kan tilpasses EDB-systemer etter behov.

Elementer i et slikt mykt system vil være naturlig språk og menyer (dvs. ja/nei-spørsmål eller nummererte alternativer).

Som et første målsystem har vi tatt for oss operativsystemet SINTRAN (2), og har konsentrert oss om å stille spørsmål til dette operativsystemet på norsk.

2. FORUTSETNINGER FOR Å BRUKE NATURLIG SPRÅK I DIALOG MED DATAMASKINER

Hvorfor er naturlig språk vanskelig for en datamaskin?

- Mangel på presisjon, kompleksitet.

Det er flere grunner til at naturlig språk er vanskelig å behandle.

Datamskiner er ekspert på å utføre algoritmer, dvs. et begrenset antall entydige regne- eller beslutningsregler, men de har hittil ikke vært flinke når beslutningsreglene er upresise eller avhengige av ytre sammenhenger.

Som et lite eksempel kan vi ta setningen

HVILKEN X' HAR Y'

Her kan X' være subjekt og Y' objekt som f.eks.

HIVLKEN ANSATT HAR TELEFON?

eller X' har vært objekt, og Y' subjekt som i

HVILKEN TELEFON HAR HANSEN?

- Stort ordforråd

Mens datamaskinspråk hittil har et meget begrenset ordtilfang (f.eks. et kommando-språk kan ha et par hundre), så vil naturlige språk ha et ordforråd som er gigantisk i forhold. (Det ekstreme tilfelle er "Complete and unabridged" OXFORD Dictionary med 400.000 innslag).

- Stor kunnskapsmengde

Et voksent menneske har minst 18 års intens opplæring i bruk av naturlig språk i vid forstand, og besitter en enorm kunnskapsmengde som han bare delvis er bevisst.

Hvilke restriksjoner må vi lage?

Den viktigste forutsteningen for å lage dialog-system i naturlig språk er å innføre begrensninger for at systemet skal være praktisk mulig å implementere. Begrensningene må være slik at

en ny bruker ikke trenger mer opplæring enn det som kan presenteres i begynnelsen av en dialog. Reglene må være lette å formulere, og lette å huske.

Avvisning og feilmeldinger må dessuten alltid være konstruktive, og referere til alment vedtatte normer som f.eks. norsk grammatikk eller vanlig betydning av ord.

Brukeren er den endelige dommer over et system. I praksis blir det en forutsetning for å lykkes at brukeren er takknelig over å få kunne uttrykke seg på sitt morsmål, selv med en del restriksjoner. Det vil si at naturlig språk er mest velegnet for uerfarne brukere.

Restriksjoner faller i 2 hovedgrupper:

- begrensninger i emnet for dialogen
- begrensninger i språket

Begrensninger i emnet vil være en grei restriksjon, som er ikke bare nødvendig men også ønskelig. Vi må i alle fall lage spesialanvendelser først, og la generaliseringene komme etterpå.

Begrensninger i språket er mer problematisk.

Det er flere typer regler for naturlig språk:

1) ordforråd (leksikalske regler)
2) bøyningsregler (morfologiske regler)
3) ordstillingsregler (syntaks)
4) betydningsregler (semantikk)

1) Ordforråd kan begrenses til opplisting av godkjente ord, eller eventuelt eksklusjon av forbudte ord. For eksempel kan ordene AT og Å forbys fordi de gir opphav til for abstrakte setninger. En annen restriksjon er å begrense bruk av verb. Dette kommer vi tilbake til.

   Restriksjoner på ordforrådet kan formildes noe ved å innlede en dialog når et ukjent ord påtreffes.

2) Når det gjelder bøynigsregler, så bør strategien være at dersom systemet finner en bøyningsfeil som ikke forandrer meningen i setningen, så skal den godtas. Vår erfaring hittil er at vi kommer langt ved bare å analysere ordstammene, og finne meningen ved hjelp av konteksten.

3) Ordstillingsregler
Det bør være akseptabelt at man i utgangspunktet forlanger grammatikalsk riktig ordstilling, siden alle brukere bør forutsettes å kunne det.

4) Betydningsregler
Betydningsregler eller semantikk er vanskelig å formalisere. Gode feilmeldinger kan alikevel gi en pekepinn om hvor misforståelsen er.

3. VERBFRITT SPRÅK

Datamaskiner som brukes til datalagring har tradisjonelt store mengder ensartede data. Situasjoner vil endre seg etter hvert som vi er i stand til å mestre mer komplekse informasjonsstrukturer. Vi kan alikevel forutsi noe om arten av informasjon som vil bli lagret:

- informasjon som er uavhengig av tid og sted og hvem som fortalte det, altså ikke

  "Han er 14 uker gammel"

  men

  "Halvard er født 13/7-1981".

- opprinnelige informasjon i motsetning til avledet informasjon

  Altså, ikke

  "Per er eldre enn Pål"

  men

  "Per er født i 1975"
  "Pål er født i 1976"

Mange fenomener kan beskrives ved tilstander og endringer i tilstander. Vi kommuniserer vanligvis ikke alt vi ser og hører, bare det som er nødvendig for at tilhøreren skal kunne oppdatere sin kunnskapsbase. Som oftest uttrykker vi informasjonen indirekte, slik at det som vi ønsker å meddele er en logisk følge av det som blir sagt, og det som er kjent allerede.

Meget grovt sagt bruker vi ofte verb til å uttrykke endringer i tilstander, mens vi bruker substantiver og adjektiver til å beskrive tilstander. Om vi eliminerer bruk av andre verb enn HA og VÆRE, så vil vi fokusere på beskrivelsene av fakta og

tilstander, og dette er nettopp hensikten med restriksjonen. Den samme restriksjon møter de som skal forfatte skjema som noen skal fylle ut. De må ofte "avverbisere" informasjonen.

Eksempler på verbfrie omformuleringer (tatt fra innbydelses-skjema til denne kongressen):

Med verb	Uten verb
Hva heter du?	Hvilket navn har du?
Hvor bor du?	Hvilken adresse har du?
Hva skal foredraget ditt hete?	Hva er tittelen på foredraget ditt?
Hvor lang tid trenger foredraget?	Hva er tiden til foredraget?

Ut fra disse betraktninger har vi satt som et mål å først lage en språklig og betydningsmessig komplett system for å forstå samtaler, dvs. opplysninger og spørsmål i et verbfritt språk. Dette system skal være forutsetningsfritt, og kunne tilegne seg kunnskap på norsk. (Et blankt system vil ha den bisarre egenskap å kunne snakke norsk uten å vite noe.)

Et lite eksempel som er kjørt, vil illustrere hovedidéen. En liten gutt jeg kjenner ble meget imponert og fornøyd da vi (B) kjørte følgende sesjon på datamaskin (S).

B: ALLE GUTTER ER PERSONER
S: OK
B: NOEN GUTTER ER SNILLE
S: OK
B: NOEN GUTTER ER SLEMME
S: OK
B: JAN MAGNUS ER EN GUTT
S: OK
B: JAN MAGNUS ER SNILL
S: OK
B: HVEM ER SNILLE?
S: JAN MAGNUS
B: ER JAN MAGNUS SNILL?
S: JA

Idéen med verbfritt språk er ikke ny. Den er funnet i (1) som behandler gjennomførbarheten av å bruke engelsk som kommando og spørrespråk i industrielle applikasjoner. Ved protokollforsøk har de funnet at spørsmål og kommandoer ble formulert eller kunne lett reformuleres med setninger som bare benyttet verbene BE, HAVE og DO.

4. SOFTRAN - DIALOGSYSTEM FOR OPERATIVSYSTEMER

SOFTRAN er et dialogsystem basert på verbfritt norsk, med predefinert informasjon om operativsystemet SINTRAN (2). Dessuten er det laget tilkopling til selve operativsystemet.

SOFTRAN har en leksikalsk preprosessor (3) som er skrevet i PASCAL (4). Resten av systemet er skrevet i PROLOG (5).

Beskrivelsen av spørresystemet representerer tilstanden pr. dags dato, og vil gi en del informasjon om angrepsmåte og forventninger, men er ikke representativ for ytelse til det tiltenkt ferdige system. Resultatene må ses i relasjon til at prosjektet er i en startfase.

Alle ord som brukes må i prinsippet være kjent med stamme og bøyninger dersom de skal kunne brukes fritt. Hvis ikke kan man definere synonymer selv (på norsk).

Alle endelser blir fjernet innledningsvis. Dette virker kanskje brutalt, men virker i praksis.

Det har to gunstige effekter:

1) Systemet er ettergivende over uvesentlige grammatikalske feil, f.eks.

   HVILKE TERMINAL ER TILKOPLET

2) Implisitte påstander om antall svar blir ignorert, f.eks.

   HVILKEN FIL HAR JEG?

   når svaret er mer enn én fil.

En tredje forenkling er å slå sammen ord (f.eks. TIL, AV, FRA, PÅ, FOR, I . Jfr. det engelske ordet OF.)

Filosofien bak disse forenklingene er at den informasjon som derved forsvinner og som ikke lar seg rekonstruere av sammenhengen er unødvendig å bry seg om.

Ordklasser

Syntaksanalysen foregår etter en noe annen oppdeling enn den vanlige (substantiv, verb etc.). Vi opererer med følgende 6 klasser:

1. Entitet — Egennavn, individ
   Eksempel JEG, TA-EVA2

2. Klasse — Fellesnavn for entiteter, herunder også perfektum partisipp som brukes som klasse.
   Eksempel FIL, TERMINAL, ANSATT

3. Attributt — Substantiver som betegner en egenskap som kan ha en verdi, f.eks.
   FARGE, TYPE, STØRRELSE

4. Egenskap — Adjektiv, herunder også perfektum partisipp som brukes som adjektiv.
   Eksempel SYMBOLSK, GUL, STOR, TERMINERT

5. Verb — HA, VÆRE, UTFØRE

6. Diverse — Ordklassen er i praksis videre oppdelt. Lista er ikke komplett. Ord i parentes blir gjenkjent, men avvist.

   ALL, ALLE, ALT, (AT), AV
   BARE, BÅDE
   DE, DEN, DENNE, DET, DETTE, (DIN), (DINE), (DU)

   ELDRE, ELLER, ET, EN
   FOR, FORSKJELLIG
   HAN, HUN, HANS, HENNES
   HVA, HVEM, HVILKE, HVILKEN, HVILKET, HVOR
   (HVORFOR), (HVORDAN)

   I, IKKE, INGEN
   JEG

   LIK, LITEN
   MED, MELLOM, MEN, MEST, MIN, MINDRE, MINST
   NOE, NOEN
   OG, OGSÅ, OVER

   PÅ
   SAMME, SIN, SOM, STOR, STØRRE, STØRST
   TIL
   ULIK, UNDER
   (VI)
   YNGRE
   (Å)

5. SYNTAKSANALYSE

Metoden som blir brukt for syntaksanalyse er hentet fra Pereira og Warren (6), der PROLOG blir foreslått som et alternativ til ATN (7).

Den illustreres best ved et eksempel

En tekst i formatet ENTITET kan bestå av en tekst i formatet KLASSE fulgt av en tekst i formatet NAVN

```
    FIL TA-EVA2
X'   Y'      Z'
```

Fig. 1

Som antydet på figur 1 er mellomrommene mellom ordene markert med variable (X', Y', Z').

I PROLOG lar denne syntaksdefinisjonen uttrykke ved

```
ENTITET(X',Z'):KLASSE(X',Y'),NAVN(Y',Z').
```

og kan leses slik

For alle X', Y', Z', er det en tekst i formatet ENTITET mellom X' og Z' dersom det er en tekst i formatet KLASSE fra X' til Y' og en tekst i formatet NAVN fra Y' til Z'.

Siden vi ikke bare skal analysere, men også viderebehandle tekster, lager vi en produksjon med et resultatfelt (etter "="). Vi bygger opp en struktur som internt er en trestruktur, men som eksternt kan presenteres med et parentes-uttrykk.

Eksempel:

```
ENTITET(X',Z')= EN(M',N'):

        KLASSE(X',Y')=M',
        NAVN(Y',Z')=N'.
```

Resultatet av å analysere frasen

```
FIL SOFTRAN
```

blir en komponent i et semantisk tre

```
EN(FIL,SOFTRAN)
```

som blir analysert i den semantiske analysen.

Av de komponenter som inngår i et semantisk tre, vil vi nevne følgende:

EN (klasse, identitet)

BÅDE (egenskap, klasse)

VERDIAV (attributt, smlgn, verdi)

ANTALL (kvantifikator, klasse)

DETIL (attributt, entitet)

IKKE (egenskap)

HVA

HVILKE

ER

HAR

HAS-AV

6. SEMANTISK ANALYSE

Semantisk tre

Den syntaktiske analysen produserer et semantisk tre. Eksempel til setningen

HAR FIL SOFTRAN TYPE SYMB?

bli oversatt til en trestruktur

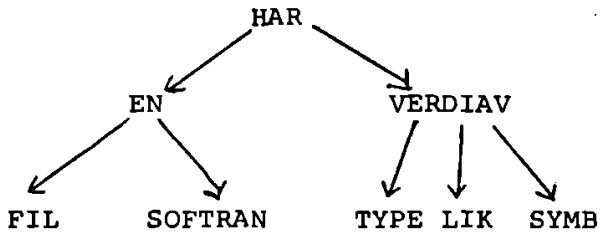


Fig. 2

som kan presenteres med parentesuttrykket

HAR (EN(FIL,SOFTRAN),VERDIAV(TYPE,LIK,SYMB))

Semantisk nett

Semantikk vil si betydningen av setningene, og vil alltid være avhengig av hvilken verden vi snakker om. (I forbindelse med datamaskiner er det ofte adekvat å snakke om mikroverden.) Det er flere filosofiske syn på hva egentlig mening er. Vi vil innta den pragmatiske holdning at et ord aldri har mening i og for seg, men bare i forbindelse med de relasjoner ordet har til andre ord. Disse forbindelser eller relasjoner kan vi ofte fremstilt grafisk i såkalte semantiske nett (9). Semantisk nett er ikke noen stringent formalisme, men en notasjonsform som kan ha mange varianter. Grunnelementene i et semantisk nett er navngitte noder og piler der nodene representerer entiteter og klasser, mens pilene representerer attributter og egenskaper.

Eksempel:

Fig. 3

som mekanisk oversatt kan leses

TORE har TELEFON 3016

Her er TORE og 3016 entiteter mens TELEFON er en relasjon.

Vi kan kople sammen opplysninger:

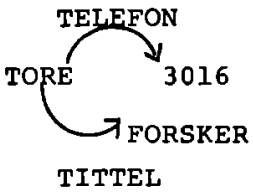


Fig. 4

I naturlig språk er det ikke bare nødvendig å vite de eksakte elementæropplysningene (hvilken telefon har Tore) men også opplysninger av mer generell art, som f.eks. hvilke klasser av ting har telefon.

Det er viktig for at spørsmål som

Hvilken tittel har telefon?

blir avvist som meningsløs, og ikke blir besvart med

"INGEN"

For å få til slike opplysninger, må vi utvide begrepsapparatet med noen spesielle relasjoner, som gis spesiell betydning.

1) ER-EN

uttrykker et element i en klasse

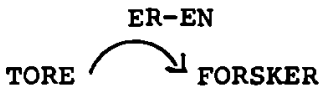


Fig. 5

Uttrykker at TORE er et medlem av klassen av forskere.

2) ER uttrykke subklasse - forhold mellom to klasser, f.eks.

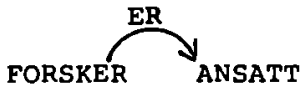


Fig. 6

som uttrykker at alle forskere er ansatt.

3) ER?

uttrykker at medlemmer av en klasse kan ha en egenskap (men uten verdi)
Eksempel:

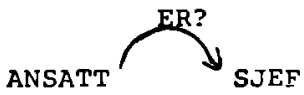


Fig. 7

4) HAR

uttrykk at alle medlemmer av en klasse har en verdi-avhengig egenskap (attributt)

Fig. 8

Uttrykker at alle ansatte har telefon, og at for hver ansatt har telefon er verdi (telefonnummer). Dersom en slik verdi mangler, er det adekvate svaret

"UKJENT".

5) HAR?

Uttrykke at noen medlemmer av en klasse kan ha en verdi-avhengig egenskap

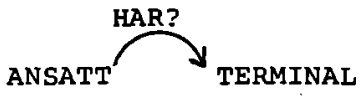


Fig. 9

HAR? uttrykker også ofte det vi forstår med et eier-forhold. Dersom en slik verdi mangler, er det adekvate svaret "INGEN".

Det semantiske nettet er på flere nivåer samtidig, både generelt og spesielt, og brukes både til å kontrollere semantikk og til å besvare spørsmål.

Følgende regler gjelder

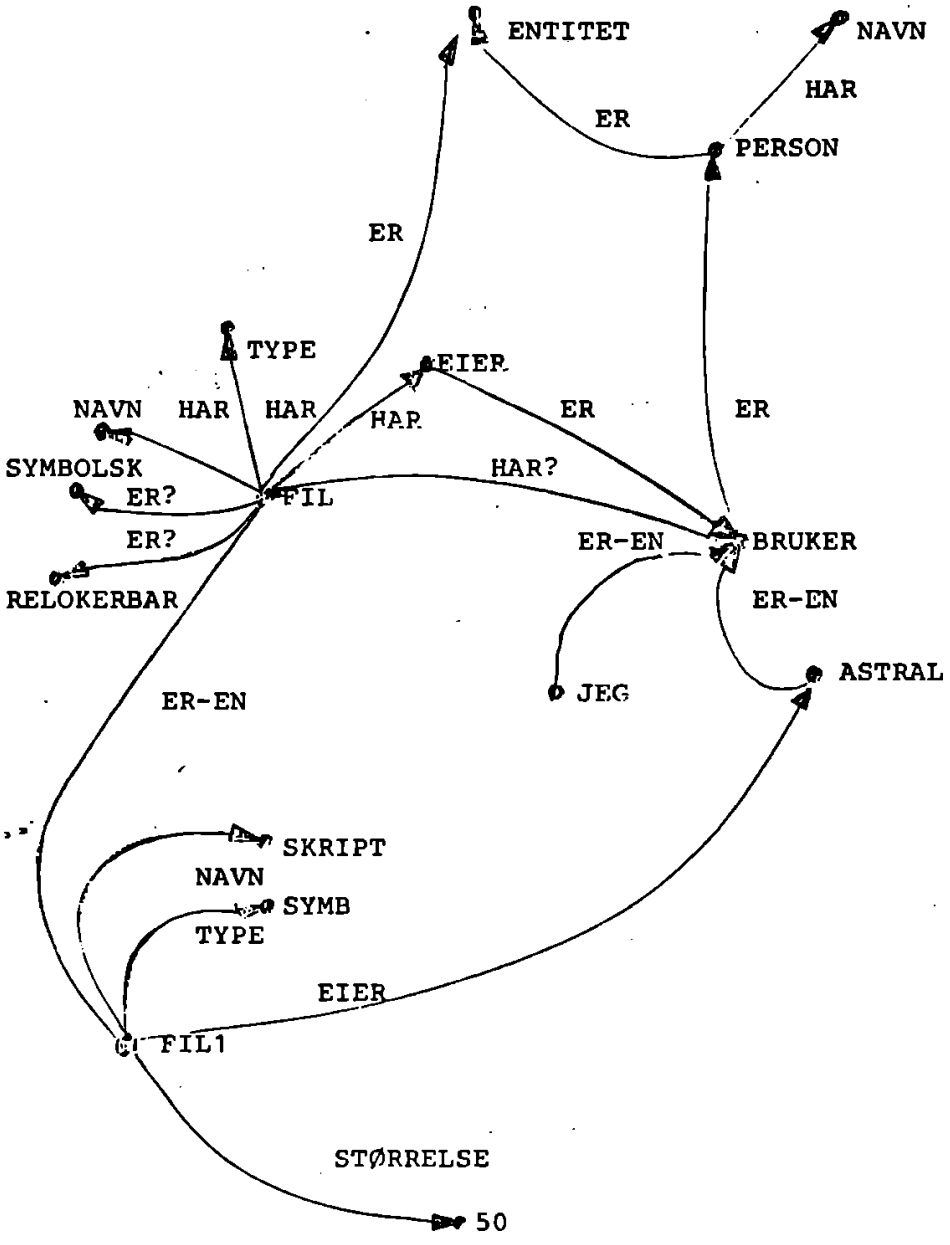
1) Dersom en entitetsklasse har et attributt eller en egenskap, så arves denne av alle entiteter som tilhører entitetsklassen.

2) En entitet (eller entitetsklasse) kan binde en verdi til et attributt dersom attributtet tilhører den omfattende klasse.

3) En egenskap kan bindes til sann eller usann dersom egenskapen er definert for klassen.

Hvordan defineres semantisk nett

Et semantisk nett kan dekomponeres i sine enkelte relasjoner, og hver av disse kan uttrykkes med en enkel setning i naturlig språk. Sammensetningen til et nett kan gjøres av et program-system.


ENTITET
NAVN
HAR
ER
PERSON
ER
TYPE
EIER
NAVN
HAR
HAR
HAR
ER
ER
SYMBOLSK
HAR?
ER?
FIL
ER?
ER-EN
BRUKER
RELOKERBAR
ER-EN
ASTRAL
JEG
ER-EN
SKRIPT
NAVN
SYMB
TYPE
EIER
FIL1
STØRRELSE
50

Figur 10

Skript

Informasjonen beskrevet i det semantiske nettet på figur 10 kan defineres ved en serie stiliserte setninger på norsk.

Alle personer er entiteter.

Alle personer har navn.

Alle brukere personer.

Alle eiere er brukere.

Jeg er en bruker.

ASTRAL er en bruker.

Alle filer er entiteter.

Noen brukere har filer.

Alle filer har navn.

Alle filer har type.

Alle filer har størrelse.

Alle filer har eier.

Noen filer er symbolske.

Noen filer er relokerbare.

FIL1 er en fil.

FIL1 har navn SKRIPT.

FIL1 har type SYMB.

FIL1 har eier ASTRAL.

Implementasjon

Implementasjon av semantiske nett blir gjort ved hjelp av PROLOG. Faktisk blir ikke nettet implementert internt som en nettstruktur, men blir lagret som tripler (tupler) i en relasjon. (Jfr. Realasjonsmodellen (8).)

For å demonstrere hvordan systemet virker, skal vi gå igjennom med et eksempel:

HVILKE SYMBOLSKE FILER HAR JEG?

som blir oversatt av leksikalanalysatoren til

(HVILKE SYMBOLSK FIL HA JEG ? NIL)

Syntaksanalysatoren vet at FIL er et klassenavn, mens SYMBOLSK er et adjektiv og JEG er en entitet. Den produserer derfor følgende semantiske tre

(HVILKE BÅDE (SYMBOLSK, FIL) HAS-AV JEG)

Den semantiske analysen går som følger:

1) BÅDE (SYMBOLSK, FIL)

   Det kontrolleres at adjektivet SYMBOLSK er definert for klassen FIL. Vi kan nå se bort fra adjektivet, og står igjen med

   HVILKE FIL HAS-AV JEG

2) Systemet undersøker hvilke klasser som har eller kan ha en FIL, og det finner én klasse BRUKER.

3) Det kontrolleres at entiteten JEG tilhører klassen BRUKER. (At JEG er synonym med en bruker ASTRAL finnes først etterpå.)

7. SVARFINNING

Svargenerering er basert på logikk og mengdelære. Eksemplet svarer til mengden

```
{x'|∃y'(x' er en fil &
        x' er symbolsk &
        y' er eier til x' &
        y' er brukernavn til JEG)}
```

Svar på slike uttrykk lar seg naturlig formulere i PROLOG.

Man kan ikke uttrykke all informasjon i et semantisk nett, til det er formalismen for grov.

Derfor må vi knytte en rekke pragmatiske betingelser til det semantiske nettet i form av logikk-programmer (5).

8. KONKLUSJON

En komplett analyse av verbfrie setninger er en nødvendig del av et hvert program for naturlig språkprosessering. Restriksjoner på bruk av verb er derfor i verste fall bare uttrykk for en midlertidig begrensning.

Vi tar et viktig steg når og hvis denne restriksjonen blir opphevet, for da åpner vi for spørsmål av høyere kompleksitet som datamaskinen neppe vil klare å forstå fullt ut. Restriksjonene som omfattes av datamaskinens kapasitet vil da bli meget vanskelige å formulere.

En dag kan vi ha den situasjon at vi har muligheter for å oversette spørsmål med verb til verbfrie spørsmål, men vi lar det være av hensyn til konsekvensene for forventningene til systemet.

Vår målgruppe er i første omgang brukere som er takknemlige for å kunne uttrykke seg på sitt morsmål, selv om språket er pålagt restriksjoner. Vi håper å finne et språklig kompromiss mellom det vi er i stand til å implementere på dagens maskiner, og det brukerne er villig til å innrette seg etter.

Epilog

I sommer kom jeg over en avisoverskrift som jeg overlater til leserne å analysere:

Tre dommere og vekk med Norge

9 REFERANSER

(1): Microcomputer-Based Natural-Language Understanding, Phase I

Machine Intelligence Corp.
Mountain View, Ca 1980.

(2): SINTRAN II User's Guide
NORSK DATA Publ.

(3): Stålhane, T: Leksikalsk analyse i Mjuke System

Nordiske Lingvistikkdager 1981
Universitetet i Trondheim

(4): Wirth, N: The Programming Language Pascal
Eidgenössische Technische Hochshule
Zürich, Juli 1973.

(5): Amble, T: Introduction to Logic Programming
RUNIT notat, Universitetet i Trondheim

(6): Pereira & Warren:
Definite Clause Grammars Compared with Augmented Transition Networks
Department of Artificial Intelligence
University of Edinburth

(7): Bates, M: The Theory and Practice of Segmented Transition Network Grammars

Natural Language Communication with Computers

Lecture Notes in CS, no 63, Springer-Verlag

(8): Date, C.J: An Introduction to Database Systems
Part 2 The Relational Approach
Addison Wesley

(9): Findler, N.V.(ED): Associative Networks

Representation and use of Knowledge by Computers

Academic Press

Kolbjørn Heggstad
Nordisk institutt,PDS
UiB

DATALINGVISTIKKEN OG DEI SPRÅKHEMMA

Lingvistikk og dei funksjonshemma i 1981.

Som lingvistar har vi som emneområde språkleg kommunikasjon. Det er grunn til å tru at vi kan lære meir om kommunikasjonssystemet om vi også studerer kva som eventuelt kan hemme eller hindre det.

Med dei tekniske hjelpemidla vi bruker får vi betre oversyn over dei ulike språkfunksjonane.

Med å studere ulike typar språkhemningar kan vi på den eine sida få ei lærerik innføring i ein språkleg mekanisme som vi enno ikkje forstår, og på den andre sida kan vi med datamaskinell hjelp løyse nokre av dei vanskane som mange har med å delta i den informasjonsstraumen som dagens kommunikasjonssamfunn har opna.

Ikkje minst forpliktar det i 1981, det internasjonale handikappåret. I den tida eg har halde på med dette har det slått meg at den kombinasjonen av innsikt vi har kan vere uhyre viktig. Vi har ein del teknisk innsikt, allsidig språkleg innsikt, har analysert og gjerne datamaskinell tilgang til dei største informasjonsbærarane, t.d. avisene.

Dei språkhemma.

Ein språkhemma er heilt eller delvis ute av stand til å delta i det hektiske kommunikasjons-samfunnet vi lever i. Ulik funksjonshemming krev tilsynelatande ulike tekniske løysingar, men svært ofte er det frå ein datalingvistisk synsvinkel berre variantar av løysingar. I dag har vi teknologi som gjer at vi kan dra mange fleire inn frå isolasjonen. Dei vi særleg tenkjer på i denne sammanhengen er dei syns-, høyrsels- eller talehemma.

Praktiske prosjekt ved PDS, delvis i samarbeid med CMI.

I 1980/81 har ein ved PDS, Universitetet i Bergen, tatt opp ei rekkje med prosjekt i samband med språkleg funksjonshemming. Saman med CMI har PDS fått løyvingar til å arbeide med språklege kommunikasjonshjelpemidlar for synshemma og døvblinde. Mykje av programmeringsarbeidet har vore utført av konsulent Michael Gillow.

Ei dagsavis for døvblinde.

I samarbeid med Foreningen Norges døvblinde har vi med finansiell støtte frå staten skaffa utstyr og utvikla eit tekstbehandlingssystem for punktskrift. Vi mottar dagleg ved instituttet tekster frå redaktøren av avisa. Tekstene blir overført direkte over telefon til ein mikrodatamaskin som så styrer resten av produksjonen. Avisa blir no produsert dagleg ut året 1981. På slutten av året skal vi også overføre avistekstene til eit trykkeri som produserer tilsvarande i storskrift for synssvake.

I samband med tekstbehandlingssystemet for dette prosjektet, har det også vore arbeidd med eit opplegg med tanke på å "forenkle" språket for denne heller språksvake gruppa.

Eit bibliotek i ein maskin.

God tilgang til aktuelle bøker har alltid vore eit problem for dei synshemma. Vi har derfor arbeidd ein del med planar om å utnytte det tilbodet som ein har ved at så mange bøker no blir trykte med hjelp av datamaskiner. Ved å utarbeide program som til ei kvar tid er tilpassa det aktuelle tekniske utstyret som skal til for å gjere boka tilgjengeleg for den synshemma, først og fremst i punktskrift eller kanskje i ulike grader av storskrift, ville mange nok eksemplar kunne gjerast tilgjengeleg til rett tid. Etter kvart som optisk lesing av bøker blir sikrare, vil tilbodet bli enno større. Med teksten i ein datamaskin vil ein kunne tilpasse seg dei ulike krav til representasjonsformer som kjem i åra framover.

Dette er eit tiltak som vi i Norge m.a. vil drøfte med Norsk tekstarkiv.

Elinfa og døvblindetelefon saman med teksttelefon.

Elinfa er eit firma som for nokre år sidan introduserte eit apparat som m.a. frå ein kassett kunne gjengi tekst i punktskrift ved hjelp av nåler i eit lesefelt. Eit spesielt mønster av nåler som stakk opp var ei tekstlinje. Eit tastatur for skriving høyrde også med. I dei siste åra har fleire andre tilsvarande apparat blitt utvikla. Ein mogeleg bruksmåte som vi tok opp er ved eit slikt apparat å kople dei døvblinde inn på det telefonsystemet som på prøvebasis er sett i gang for dei døve. Den såkalla teksttelefonen for dei døve er basert på at dei skriv på skjerm til kvarandre, no kan ein også med punktskrift vere med i denne telefonringen.

Oppslagsverk for blinde.

Oppslagsverk for blinde har vore vanskeleg å få i stand. Ein lydkassett er lite velegna, og punktbøker har vore for plasskrevande. Vi har derfor prøvd å utnytte utviklinga av billege mikrodatamaskiner og kombinert dette med eit Elinfa-apparat. Vi har no ein demonstrasjonsmodell av ei "automatisk" ordbok og ein telefonkatalog. Vi kan skrive spørsmålet inn som punktskrift og få svaret ut som punktskrift frå datamaskinen.

Avisspråk for døve.

For døve er eit skriftspråk som norsk eit framanspråk. Det viser seg vanskeleg å lese og noko bør gjerast for å tilpasse skriftspråket til denne lesargruppa. Ved PDS har ein derfor i samarbeid med Norges døveforbund engasjert seg i eit planleggings-arbeid for den språklege utforminga av ei veleigna avis for døve. Arbeidet er enno i startfasen. Det ser ut til å vere lite gjort med dette i Norge og fleire tiltak har vore drøfta.

Ordtavle for talehemma.

Eit av dei hjelpemidla som har vore brukt til talehemma har vore "peiketavler". Tavlene har gjerne hatt eit utval med ord som brukaren kunne peike på i den rekkjefølgje han ville "snakke". Til dette arbeidet har vi skaffa det engelske systemet SPLINK. Det har ei ordtavle på omlag 1000 ord, eit system som overfører dei orda det blir peikt på til ein vanleg fjernsynsskjerm. Overføringa er trådlaus. Ved PDS har ein utvikla ei norsk ordtavle og har no to norske SPLINK-system ute til utprøving. Eit av problema med denne typen utstyr er at vokabularet skal passe til alle pasientar i alle situasjonar. Ei vidareutvikling av dette har vore drøfta.

Talegjenkjenning og syntetisk tale.

Til fleire av dei prosjekta som er omtala ville eit system for talegjenkjenning og talegenerering vere interessant. Mange blinde er av ulike årsaker ikkje i stand til å bruke punktskrift. Likeeins ville døve kunne lese det som vart sagt, og den tale-hemma kunne t.d. frå tastatur gitt ei melding gjennom telefonen. I samband med prosjektarbeidet har ein del slikt apparatur vore prøvd og noko er innkjøpt. Ein del av arbeidet framover vil truleg bli knytt til noko av dette. Det vil vere mykje arbeid med å tilpasse dette utstyret til norsk språk og til vår spesielle bruk.

Konklusjon.

Som datalingvistar har vi mykje å tilføre dette arbeidsfeltet. Delvis har vi ein kombinasjon av datateknikk og lingvistikk som kunnskap, og delvis har vi ei kontaktflate med datafag, kommunikasjonsforskning og ulike greiner av lingvistikk som tilsaman gjer at eg meiner vi bør føle oss forplikta til å hjelpe dei språkhemma.

Anna Sågvall Hein
UCDL
Centrum för Datorlingvistik
Uppsala Universitet

UPPSALA CHART PARSER, Version 2 (UCP-2) - En översikt.

1. Inledning

Uppsala Chart Parser är en lingvistisk processor för analys av naturligt språk. Den representerar en vidareutveckling av de grundläggande ideerna i The General Syntactic Processor (se Kaplan 73) och utnyttjar härvid vidare ideer och erfarenheter presenterade av M. Kay (Kay 75, 77a och 77b), A. Sågvall Hein (Sågvall 77a, 77b, 78 och 80a) och M. Carlsson (Carlsson 80 och 81).

För närmare upplysningar om bakgrunden till UCP samt om hur den är relaterad till tidigare arbeten, se Sågvall 80b.

2. Allmänt om UCP.

Den centrala datastrukturen i UCP är en chart. Den fungerar som en 'anteckningsbok', där allt som äger rum under bearbetningen av ett språkligt uttryck (ordform, fras, sats) noteras. Charten representerar det obearbetade språkliga uttrycket, delanalyser samt den (eller de) resulterande analyserna, s.k. passiv information. Vidare innehåller den information om hur bearbetningen skall fortskrida, s.k. processinformation.

Charten är en riktad graf, bestående av noder (vertices) och bågar (edges). Noderna är numrerade och bågarna är etiketterade.

Man skiljer mellan inaktiva och aktiva bågar.

De inaktiva bågarna bär i sina etiketter den passiva informationen t.ex. lingvistiska beskrivningar över språkliga enheter. Initialt representeras det språkliga uttryck som skall analyseras (analysuttrycket) avm en enkel chart bestående av inaktiva bågar, där varje båge representerar en karaktär i analysuttrycket. Under bearbetningens gång introduceras kontinuerligt nya bågar i charten, vilka representerar partiella analysresultat. Den slutliga analysen lagras i etiketten till den (eller de) inaktiva båge (eller bågar) som omspänner hela charten från första till sista vertex. Om analysen inte lyckas, återfinns inte någon sådan båge i charten.

De aktiva bågarna innehåller i sina etiketter antingen namn på grammatiska regler som skall appliceras eller namn på lexikon (eller den punkt i ett visst lexikon) där lexikonsökning skall ske.

Utnyttjande av chartstrukturen gör det möjligt att

- känna igen samt beakta samtliga ambiguiteter i en sats
- analysera satsfragment
- undvika upprepad igenkänning av satsfragment

Hela bearbetningen i UCP tar formen av en följd av bearbetningssteg, tasks, som genereras och exekveras. Exekvering av en task innebär ett försök att tillämpa en grammatisk regel på en båge i charten eller en jämförelse mellan en båge i charten och en bokstav i något uppslagsord i något lexikon. (Uppslagsorden i lexikonen ligger lagrade som bokstavsträd.)

Nya tasks genereras automatiskt som följd av att nya bågar introduceras i charten. Detta sker enligt en allmän princip som säger att en task skall genereras för varje väg, path, i charten bestående av en aktiv edge följd av en inaktiv edge.

Introduktionen av nya bågar i charten styrs från de grammatiska reglerna och lexikonartiklarna. Kontrollen är sålunda decentraliserad.

När en task har genererats överförs den (scheduleras) till en agenda, varifrån den sedan hämtas (selecteras) för exekvering.

Agendan samt charten ger en komplett beskrivning av status på bearbetningen i varje moment.

UCP arbetar med godtyckligt antal lexikon och grammatikor.

I en och samma grammatiska formalism uttrycks såväl fonotaktiska som morfotaktiska och syntaktiska regler.

Formalismen är rent procedural.

I den procedurala formalismen uttrycker vi också informationen i de enskilda lexikonartiklarna.

Så snart ett uppslagsord återfunnits i något lexikon vidtar en interpretering av lexikoninformationen helt i analogi med interpreteringen av de grammatiska reglerna.

För varje enskilt lexikon kan vi också specificera vilka aktioner som skall utföras då ett lexikonsökningssteg lyckats. Även detta uttrycks i den grammatiska formalismen. Se vidare 4.1.1.1.

3. Den grammatiska formalismen

Den grammatiska formalismen inkluderar ett specificerat format, en struktur, för beskrivning av lingvistiska enheter samt en uppsättning grammatiska operatorer.

3.1 Strukturformatet

En struktur är en lista av egenskaper, där varje egenskap formuleras som ett attribut och ett värde.

3-1 visar en struktur som beskriver den svenska frasen 'i denna film'.

```
(SYN.CONST = PREP.PHRASE
 1 = (CAT = PREP
      LEX = I)
 2 = (SYN.CONST = NP
      1 = (CAT = DETER
           LEX = DENNA
           UTR.NEUTR = UTR
           ...)
      2 = (CAT = NOUN
           LEX = FILM
           GENDER = REAL
           UTR.NEUTR = UTR
           ...)))
```

Figure 3-1:

Ett attribut är ett (med några undantag, se nedan samt 3.3) fritt valt namn, en atom, t.ex. SYN.CONST (för syntaktisk konstituent) i figuren.

Ett värde är antingen en atom, t.ex. PREP.PHRASE i figuren eller en struktur, t.ex. (SYN.CONST = NP, 1 = (CAT = DETER, LEX = DENNA, ...), 2 = (CAT = NOUN, LEX = FILM, ...)).

Lingvistiska beskrivningar i form av strukturer lagras i etiketterna till bågarna i charten.

Med hjälp av de grammatiska operatorerna formuleras primitiva grammatiska operationer.

En grammatik består av en uppsättning grammatiska regler.

En grammatisk regel består av följd, en sekvens, av grammatiska operationer. I en grammatisk regel uttrycker man de operationer som skall utföras med avseende på en given inaktiv båge. _ Vilken den inaktiva bågen är specificeras i aktuellt bearbetningssteg (task).

Då vi formulerar en grammatisk regel refererar vi till den strukturella beskrivningen av den inaktiva bågen med hjälp av det reserverade attributnamnet *. (Den inaktiva bågens strukturbeskrivning utgör värdet på attributet *).

Även den aktiva bågens etikett har utrymme för en strukturbeskrivning. Det är där som interpretatorn lagrar den struktur som byggs upp under analysen av ett visst språkligt uttryck. Till den aktiva bågens strukturbeskrivning refererar vi i formuleringen av den grammatiska regeln med hjälp av attributnamnet &.

I 3-2 ger vi ett exempel på attributet & och dess värde för en aktiv edge.

```
(& = (SYN.CONST = PREP.PHRASE
      1 = (CAT = PREP
            LEX = I)))
```

Figure 3-2:

Den aktiva edge, vars strukturbeskrivning, egenskapslista, visas i 3-2, representerar en delanalys av uttrycket 'i denna film'.

3-3 visar denna edge _ den riktade bågen från 1 till 3 _ i den aktuella chartstrukturen.

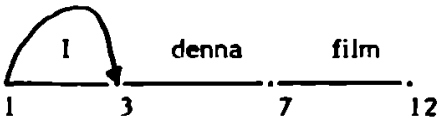


Figure 3-3:

Anm. 1 Figuren visar endast de bågar och noder som är väsentliga för att visa den aktiva edgen i sitt sammanhang.

Anm. 2 De inaktiva edgarna är inte riktade till skillnad från de aktiva. För detaljer rörande implementeringen av chartstrukturen, se Carlsson 80 och 81.

Förutom strukturbeskrivningen innehåller de aktiva bågarna också information om vilken regel (eller lexikonbåge vid lexikonsökningen) som skall tillämpas, samt uppgift om i vilket bearbetningssteg, task, som de genererats.

3-4 presenterar den fullständiga utskriften av den aktiva bågen i exemplet.

```
1--3  CREATOR: 39
      FEATURES: (& = (SYN.CONST = PREP.PHRASE
                      1 = (CAT = PREP
                           LEX = I)))
      LR-ACTION: (CONTINUE,
                  <* SYN.CONST> = 'NP,
                  <& :new> :=: <*>,
                  STORE),
                 RULEEXIT:(PREP:PHRASE)
```

Anm. De grammatiska operatorerna som förekommer under rubriken L(eft to) R(ight) ACTION förklaras nedan i 3.2.

Figure 3-4:

3.2 De grammatiska operatorerna

Med hjälp av de grammatiska operatorerna kan vi

- utföra tester,
- bygga upp lingvistiska beskrivningar, samt
- styra den vidare bearbetningen genom att införa nya bågar i charten

3.2.1 Testoperatorer

Testerna som utförs i de grammatiska operationerna avser egenskapslistorna hos den aktiva och den passiva båge som bearbetningssteget gäller.

Med hjälp av PATH-operatorn (notation: < >) kan vi extrahera värdet på en viss egenskap.

T.ex. <& SYN.CONST> applicerat på den aktiva edgen i 3-4 returnerar värdet PREP.PHRASE.

Med PATH-operatorn kan vi komma åt värdet av en egenskap på godtyckligt djup i en struktur.

T.ex. <& 1 CAT> applicerat på samma struktur ger värdet PREP.

En PATH-operation kan också returnera en hel delstruktur.

T.ex. <& 1> i exemplet ovan returnerar (CAT = PREP, LEX = I).

Samma resultat får vi genom att använda oss av det reserverade attributnamnet :LAST, som returnerar den sista delstrukturen i en strukturbeskrivning. Operationen skulle då formuleras <& :LAST>.

Alla operationer returnerar ett värde, antingen ett faktiskt värde som i exemplen ovan vilket då tolkas som TRUE eller ett negativt värde, NIL. Om den egenskap man frågar efter i PATH-operationen saknas i strukturen, så returneras värdet NIL. Ett PATH-uttryck kan sålunda i sig fungera som en test.

EQUALITY-operatorn (notation: =) används för att testa på likhet mellan två PATH-uttryck eller mellan ett PATH-uttryck och en atom (ett visst ord).

T.ex <& 1 LEX> = 'PREP i exemplet ovan, skulle returnera värdet TRUE. (Atomer måste markeras med ett enkelt anföringstecken enligt LISP-konvention.)

Det finns också en NEGATIONs-operator (notation: NOT) som t.ex. ger oss möjlighet att formulera ett test 'icke lika med'.

Med hjälp av OR-operatorn (notation: /) kan vi uttrycka alternativ, t.ex.

```
(<* FORM> = 'DEF /
 <* NUMB> = 'PLUR /
 <* PROPR> = T /
 <* CBLTY> = 'MASS)
```

i en NP-regel som krav på att ett ensamt substantiv skall få betraktas som en NP.

OR-operatorn returnerar TRUE om någon av dess operationer lyckas. Efterföljande operationer utförs ej (dependent disjunction).

Konditionala regler kan formuleras med hjälp av CONDITIONs- operatorn (notation: IF... THEN ... ELSE).

3.2.2 Presentation av SAME, ADVANCE, STORE och MAJORPROCESS i anslutning till ett exempel ur SVE.GRAM

För att illustrera den fortsatta presentationen av de grammatiska operatorerna anför vi i 3-5 en grammatisk regel för igenkänning av en NP-konstituent bestående av en determinerare och ett substantiv i svenska. Regeln är hämtad ur vårt fragment av en svensk grammatik för UCP.

```
DETER.NP
    <& SYN.CONST> :=: 'NP,
    <* CAT> = 'DETER,
    <& :NEW> :=: <*>,
    ADVANCE,
    <* CAT> = 'NOUN,
    <& :LAST NUMB> = <* NUMB>,
    <& :LAST UTR.NEUTR> = <* UTR.NEUTR>,
    (<& :LAST LEX> = <* DENNA>,
     <* FORM> = 'INDEF /
     <* FORM> = 'DEF),
    <& :NEW> :=:<*>,
    STORE,
    MAJORPROCESS(NP)
```

Figure 3-5:

I 3-6 presenterar vi en förenklad chartstruktur som visar en situaton där DETER.NP regeln aktiveras.

```
     I  O  denna      film
  ______________._________.
  1    3          9        14

  3--3 CREATOR: 13
       LR-ACTION: DETER.NP

  3--9 CREATOR: 12
       FEATURES: (* = (CAT = DETER
                       LEX = DENNA
                       NUMB = SING
                       UTR.NEUTR = UTR
                       DEF.INDEF = DEF
                       FORM = INDEF))

  . . .
```

Figure 3-6:

En grammatisk regel består av en SEQUENCE (notation: ,) av operationer. Under interpreteringen av en regel exekveras de olika operationerna till dess någon misslyckas (returnerar värdet NIL). Sekvensen som sådan returnerar värdet TRUE om alla de ingående operationerna lyckas (returnerar värdet TRUE). Om någon operation misslyckas, så returnerar sekvensen som sådan värdet NIL.

Sekvenser kan uppträda som argument till andra operatorer (t.ex. i konditionala uttryck och OR-uttryck).

DETER.NP regeln i 3-5 är en sekvens bestående av 11 operationer.

Den första operationen är en tilldelningssats, formulerad med hjälp av SAME-operatorna (notation: :=:). I operationen tilldelas attributet SYN.CONST i den aktiva bågens egenskapslista värdet NP. Med andra ord, så börjar vi här bygga upp en lingvistisk beskrivning av den konstituent vi förväntar oss att känna igen.

I nästa sats testar vi på den inaktiva bågens kategoritillhörighet. Vi kräver att den skall ha egenskapen CAT = DETER. (Till den syntaktiska kategorin DETER hänför vi t. ex. 'denna' och 'den här'.

I den tredje satsen gör vi en ny tilldelning, där vi använder oss av det reserverade attributnamnet :NEW samt av path-uttrycket <*>.

Satsen innebär att hela den inaktiva bågens egenskapslista <*> ges som värde till attributet :NEW i den aktiva bågens strukturbeskrivning. Användningen av det reserverade attributnamnet :NEW får till följd att det byggs upp en ny egenskap, vars attribut är en siffra, som är en enhet högre än tidigare numrerade attribut. Denna facilitet används för att kunna numrera delkonstituenter i en konstituent.

ADVANCE-operatorn är en av de 5 process-operatorerna. Med en process-operator menar vi en operator som lägger in nya bågar i charten och därigenom för bearbetningen framåt. (Alla processoperatorerna returnerar värdet TRUE.)

ADVANCE-operatorn har den effekten att en ny aktiv båge läggs in i charten. Denna båges egenskapslista innehåller attributet & med tillhörande värde. Den innehåller vidare resten av den grammatiska sekvens i vilken den ingår. Intuitivt svarar ADVANCE-operatorn mot att man tar ett steg framåt i analysuttrycket; en eller flera nya tasks kommer att genereras vilka gäller den nya aktiva bågen och följande inaktiva båge/bågar.

3-7 visar chartstrukturen (i förenklad form) efter exekveringen av ADVANCE-satsen i regeln DETER.NP.

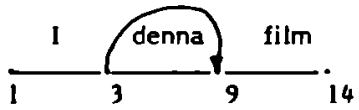
Den fortsatta interpreteringen av DETER.NP-regeln kommer sålunda att ske i en ny task som gäller den aktiva bågen från 3 till 9 samt den passiva bågen från 9 till 14.

Här testas först den inaktiva bågens kategoritillhörighet. Därpå testas överensstämmelse med avseende på egenskaperna UTR.NEUTR och NUMB. (Obs. Om någon test skulle misslyckas så avbryts exekveringen av tasken, eftersom operationerna ingår i en sekvens.)

För referens till den sist igenkända delkonstituenten i den aktiva edgen använder man sig av det reserverade attributnamnet :LAST.

Därpå följer en OR-operation. (Obs. Både 'denna' och 'den här' är kategoriserade som determinerare. Alternativen motiveras av att de kräver indefinit resp. definit form på substantivet.)

Efter OR-operationen följer en STORE-operation. STORE tillhör processoperatorerna. En STORE-operation innebär att en ny inaktiv edge införs i charten. Den får som strukturbeskrivning värdet av attributet &, dvs. den övertar den aktiva bågens




```
3--9 CREATOR: 29
     FEATURES: (& = (SYN.CONST = NP
                     1 = (LEX = DENNA
                          CAT = DETER
                          NUMB = SING
                          UTR.NEUTR = UTR
                          FORM = INDEF))
     LR-ACTION: (CONTINUE,
           (<CAT *> = 'NOUN,
           <& :LAST NUMB> = <* NUMB>,
           <& :LAST UTR.NEUTR> = <* UTR.NEUTR>,
           (<& :LAST LEX> = 'DENNA, <* FORM> = 'INDEF /
            <* FORM> = 'DEF),
           <& :NEW> :=: <*>,
           STORE,
           MAJORPROCESS(NP))
           RULEEXIT: DETER.NP)

9--14 CREATOR: 21
   FEATURES: (* = (CAT = NOUN
                   LEX = FILM
                   UTR.NEUTR = UTR
                   NUMB = SING
                   FORM = INDEF
                   ...))

...
```

Figure 3-7:

egenskapslista. Den omspänner den följd av bågar som regeln omfattar. I vårt exempel medför STORE att en ny inaktiv edge läggs in i charten från vertex 3 till vertex 14.

DETER.NP-regeln avslutas med operationen MAJORPROCESS(NP). Operatorn MAJORPROCESS är den tredje av processoperatorerna. Den tar som argument namnet på en grammatisk regel (eller namnet på en grammatik eller ett på ett lexikon). NP är namnet på en regel i vår svenska grammatik. Operationen MAJORPROCESS(NP) innebär att en ny aktiv edge, vars LR-ACTION är NP, införs i charten. Denna aktiva edge går från och till den nod i vilken aktuell regel aktiverades, dvs. i vårt exempel i vertex 4. I NP-regeln beskrivs NP-konstituenter, och regeln inkluderar igenkänning av efterställda bestämningar, dvs. relativa satser och prepositionsfraser. Tillgång till operatorn MAJORPROCESS innebär möjlighet att arbeta med ett look-ahead. Den bidrar till att göra analysen datadriven. Först när vi ser vad vi har initierar vi en viss process.

I 3-8 presenteras en analys av uttrycket 'I denna film från Kanada' enligt vårt nuvarande grammatikfragment för svenska.

```
I denna film från Kanada:

(SYN.CONST = PREP.PHRASE
 1 = (CAT = PREP
      LEX = I)
 2 = (SYN.CONST = NP
      1 = (SYN.CONST = NP
           1 = (CAT = DETER
                LEX = DENNA
                ...)
           2 = (CAT = NOUN
                LEX = FILM
                ...))
      2 = (SYN.CONST = PREP.PHRASE
           1 = (CAT = PREP
                LEX = FRÅN)
           2 = (SYN.CONST = NP
                1 = (CAT = NOUN
                     LEX = KANADA
                     PROPR = T
                     ...))))
```

Figure 3-8:

3.2.3 Presentation av PROCESS i anslutning till ett exempel ur SVE.DIC

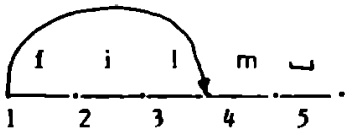
3-9 visar den lexikoninformation som är associerad med uppslagsordet 'film' i vår svenska applikation av UCP. Lexikonartikeln är hämtad från huvudlexikonet SVE.DIC.

```
FILM : <& LEX> :=: 'FILM,
       <& MORPH.CAT> :=: 'ROOT,
       <& ROOT.CAT> :=: 'NOUN,
       <& ROOT> :=: 'FILM,
       <& GENDER> :=: 'REAL,
       <& ANIM.INANIM> :=: 'INANIM,
       <& CBLTY > :=: 'COUNT,
       <& PROPR > :=: 'NO,
       <& UTR.NEUTR> :=: 'UTR,
       <& DECL> :=: '-ER,
       STORE,
       MAJORPROCESS(NOUN),
       ADVANCE,
       (<* CHAR :TYPE> = 'SEP, PROCESS(SEP)/
        (<* CHAR> = 'E, PROCESS(NUMB), PROCESS(FORM)/
         <* CHAR> = 'S, PROCESS(CASE), PROCESS(SVE.DIC)),
         MAJORPROCESS(COMPOUND))
```

Figure 3-9:

3-10 visar en (förenklad) chartstruktur, där denna 'regel' aktiveras. Exemplet gäller analysuttrycket 'film'.

Sekvensen i 3-9 inleds med 10 tilldelningssatser. Därpå följer en STORE-operation, som leder till att en ny inaktiv edge från 1 till 5 läggs in i charten. Denna edge har en egenskapslista som överensstämmer med tilldelningarna ovan.




```
1--2 CREATOR: 0
     FEATURES: (* = (CHAR = F))

2--3 CREATOR: 0
     FEATURES: (* = (CHAR = I))

3--4 CREATOR: 0
     FEATURES: (* = (CHAR = L))

4--5 CREATOR: 0
     FEATURES: (* = (CHAR = M))

5--6 CREATOR: 0
     FEATURES: (* = (CHAR =  ))

1--4 CREATOR: 6
     FEATURES: NIL
     LR-ACTION: <& LEX> :=: 'FILM,
                <& MORPH.CAT> :=: 'ROOT,
                 ...
 ...
```

Figure 3-10:

MAJORPROCESS(NOUN) innebär att en ny aktiv edge från och till vertex 1 läggs in i charten. Dess LR-ACTION är en hänvisning till den grammatiska regeln NOUN, som specificerar strukturen för ett svenskt substantiv. I kraft av den allmänna taskgenereringsprincipen kommer denna edge att leda till att substantivregeln kommer att börja tillämpas på stammen 'film'.

ADVANCE-operationen innebär att vi tar ett steg (edge) framåt i charten för att kunna se följande karaktär.

Här har vi specificerat 2 huvudalternativ

1. karaktären har typmarkering SEParator

2. karaktären är identisk med e eller s

För det första alternativet har vi utnyttjat en facilitet i UCP som gör det möjligt att lagra och återvinna egenskaper hos de enskilda karaktärerna. Sålunda har mellanslaget och skiljetecknen fått typbeteckningen 'separator'. Med hjälp av det reserverade attributnamnen :TYPE kan vi fråga på denna egenskap. Övrig information om separatorerna finns lagrad i ett separatorlexikon (SEP), där varje separator ligger som ett eget uppslagsord.

PROCESS är den 4.e processoperatorn. Den initierar processerande med avseende på en viss grammatik eller ett visst lexikon. I argumentet till PROCESS anger man namnet på grammatiken (eller visst läge i en aktiv grammatik) eller på lexikonet. Processerandet startar i den vertex i vilken den aktuella aktiva edgen slutar. (Jfr. MAJORPROCESS som initierar processerande i den vertex där den aktuella aktiva edgen börjar.)

PROCESS(SEP) leder i vårt exempel till att lexikonsökning i separatorlexikonet startar i vertex 5. Om påföljande karaktär är 'e' så initieras processerande i både numeruslexikonet (NUMB), jfr. 'filmer', och i formlexikonet (FORM), jfr. 'filmen'. Är påföljande karaktär 's' så initieras lexikonsökning i kasuslexikonet (CASE), jfr. 'films'.

Inte i något av de sistnämnda fallen kan man utesluta möjligheten att andra ledet i en sammansättning följer. Därav MAJORPROCESS(COMPOUND), PROCESS(SVE.DIC).

3.2.4 Presentation av MINORSTORE samt ALTERNATION och RULE CALL i anslutning till ett exempel ur SVE.GRAM

MINORSTORE fungerar som STORE med den skillnaden att den inaktiva edge som läggs in omspänner samma sekvens av edgar som den aktuella aktiva edgen. (STORE-operatorn lägger in en inaktiv edge som omspänner samma följd av edgar som den aktiva samt följande inaktiva.) MINORSTORE i kombination med ADVANCE kompletterar våra möjligheter till ett look-ahead av en edge (jfr. MAJORPROCESS).

ALTERNATION är en operator som möjliggör parallellprocesserande. Den används i situationer där man inte t. ex. med OR-operatorn kan välja alternativ utan måste gå fram med flera tolkningar.

I 3-11 illustreras användningen av ALTERNATION i den grammatiska regeln NO.NOUN.ENDING i vår experimentgrammatik SVE.GRAM.

```
NO.NOUN.ENDING:

    (<& DECL> = '-, (<& NUMB> ::: 'SING // <& NUMB> ::: 'PLUR) /
     <& NUMB> ::: 'SING),
    <& FORM> ::: 'INDEF,
    <& CASE> ::: 'COMMON,
    STORE;
```

Figure 3-11:

Regeln i 3-11 aktiveras i en situation där man har en följd av en substantivisk stam och en separator. Utgående från stammens deklinationstyp gör den aktuella tilldelningar. Med hjälp av OR-operatorn gör den åtskillnad mellan substantiv av deklinationstyp '-' (ingen ändelse i pluralis) och övriga. För substantiv tillhöriga den förstnämnda typen görs två parallella tilldelningar till attributet NUMB (numerus), nämligen SING och PLUR. Övriga substantiv tilldelas värdet SING för numeruskategorin. I charten avspeglas parallella tilldelningar i form av alternativa edgar.

3-11 exemplifierar ytterligare en facilitet i den grammatiska formalismen. För att göra grammatiken överskådlig och för att slippa upprepa sekvenser av operationer som förekommer på flera ställen i grammatiken eller i flera lexikonartiklar, kan vi extrahera sådan sekvenser och föra upp dem under egna namn i grammatiken. Tack vare RULE CALL faciliteten kommer dessa sekvenser att interpreteras då deras namn påträffas i en regel. NO.NOUN.ENDING-regeln i figuren är en sådan sekvens. De grammatiska operationerna i regeln kommer att exekveras då interpretatorn träffar på namnet NO.NOUN.ENDING i regeln NOUN.

3.3 Reserverade attributnamn

I 3.1 nämns att attributnamn med några undantag när kan väljas fritt. Det finns tre typer av reserverade attributnamn, nämligen

1. & och * ,

2. :NEW och :LAST, samt

3. övriga attributnamn, prefigerade med : , t.ex. :TYPE.

Dessa tre typer av attributnamn behandlas olika av interpretatorn.

& och * behandlas som de fritt valda attributnamnen, dvs. de tolkas bokstavligt.

2. utlöser vid interpreteringen subrutinanrop. :NEW har den effekten att ett attributnamn skapas. Det skapade attributnamnet är en siffra, som är en enhet högre än det högsta redan förefintliga sifferattributet i den aktuella aktiva edgens strukturbeskrivning (jfr. 3-5). :LAST i ett path-uttryck identifieras av interpretatorn som en referens till det sista attributet i en given strukturbeskrivning (jfr. 3-5).

3. förekommer enbart i path-uttryck som avser karaktärer. Förekomst av prefixet : på ett attributnamn signalerar till interpretatorn att egenskapen inte ligger lagrad i den inaktiva edgens egenskapslista utan som en global egenskap till karaktären i fråga. Denna facilitet gör det möjligt att utifrån grammatiken kunna komma åt globalt lagrade egenskaper avseende karaktärerna.

I vår svenska applikation har vi bl. a. utnyttjat :-konventionen för att kunna fråga på globalt lagrade fonetiska särdrag utifrån fonologiska regler. Exempel på detta ges i 4.1.1.

4. Processintegrering

Som framgått av presentationen av de grammatiska operatorerna har man fullständig frihet att initiera olika slags processer från en grammatisk regel eller en lexikonartikel. Härvid utnyttjar man operatorerna PROCESS och MAJORPROCESS. Man har möjlighet att experimentera med såväl regelstyrd (top-down) som datastyrd (bottom-up) bearbetning tillika med en kombination av båda.

Förutom från grammatiska regler och lexikonartiklar kan man initiera godtyckligt processerande från de aktioner som kan specificeras att gälla för sökningen i ett givet lexikon (se 4.1.1.1).

Exemplen ovan har visat samspelet mellan grammatiska regler och lexikonartiklar. Nedan illustrerar vi processintegrering avseende lexikonsökning och omskrivning.

4.1 Lexikonsökning och omskrivning

Ett problem som ständigt dyker upp i samband med språkanalys är frågan om hur man skall hantera fonologiska och morfofonematiska alternationer. Vanligtvis definierar man s. k. omskrivningsregler (rewriting rules) vilka återför alternanterna till en kanonisk lexikonform.

Nästa fråga gäller hur interpreteringen av omskrivningsreglerna skall integreras med övrig bearbetning.

Den grammatiska formalismen i UCP erbjuder en notation i vilken omskrivningsreglerna kan definieras.

Vad gäller integreringen av omskrivningsprocessen med övriga processer, så har man en gynnsam situation i UCP.

För ett givet lexikon kan man specificera vilka aktioner som skall utföras i samband med att en lexikonsökningstask lyckas. Detta innebär, att man utifrån signaler i analysuttrycket kan aktivera olika omskrivningsprocesser.

Nedan presenterar vi en strategi för integrering av omskrivning och lexikonsökning på ett svenskt exempel.

4.1.1 Ett exempel från svenskan

I många svenska ord förekommer s. k. flyktig vokal. Vi tänker på fall som 'cykel' och 'cyklar', 'äpple' och 'äppelträd', 'kommen' och 'komna' etc.

I dessa fall kan man tänka sig att betrakta formen utan flyktig vokal som den kanoniska lexikonformen, och via någon omskrivningsregel återföra den utvidgade formen till den kanoniska.

4-1 ger exempel på hur en sådan regel skulle kunna se ut i vår grammatiska notation. I 4-1 utnyttjar vi en möjlighet, som vi tidigare inte nämnt, nämligen den att kunna ge argument till STORE-operatorn.

STORE utan argument har den effekten att en ny edge läggs in i charten. Denna edge omspänner samma följd av bågar som den aktuella aktiva edgen inklusive närmast följande inaktiva edge. Den övertar den aktiva edgens egenskapslista.

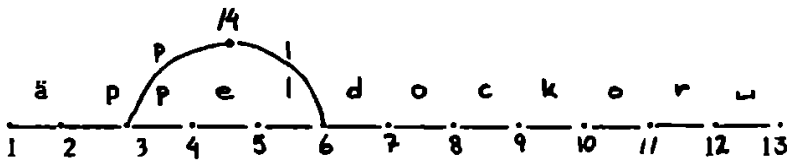
Om STORE ges med argument, så lägger den in lika många nya inaktiva edgar som

```
UNSTABLE.VOWEL
    (<* CHAR :MODE> = 'STOP/<* CHAR> = 'S/<* CHAR> = 'M),
    <& FIRST.CONS CHAR> :=: <* CHAR>,
    <& FIRST.CONS REWR> :=: 'UNSTABLE.VOWEL,
    ADVANCE,
    <* CHAR :TYPE> = 'VOWEL,
    <& FIRST.CONS UNSTABLE.VOWEL> :=: <* CHAR>,
    ADVANCE,
    (<* CHAR :MODE> = 'LIQUID/<* CHAR> = 'N),
    <& SEC.CONS CHAR> :=: <* CHAR>,
    NOT <& FIRST.CONS CHAR> = <& SEC.CONS CHAR>,
    <& SEC.CONS REWR> :=: 'UNSTABLE.VOWEL,
    <& SEC.CONS UNSTABLE.VOWEL> :=:
      <& FIRST.CONS UNSTABLE.VOWEL>,
    STORE(<& FIRST.CONS>,<& SEC.CONS>)
```

Figure 4-1:

antalet argument. I argumenten specificeras egenskapslistorna för de olika bågarna. Följden av inaktiva edgar som läggs in i charten omspänner samma följd av edgar som den som STORE utan argument inför.

Appliceras omskrivningsregeln UNSTABLE.VOWEL i vertex 3 i en chart som representerar ordet 'äppeldockor' så får vi följande resultat (se 4-2).




```
3--3 CREATOR: 9
       LR-ACTION: UNSTABLE.VOWEL

3--4  CREATOR: 0
      FEATURES: (* = (CHAR = P))

3--14 CREATOR: 12
      FEATURES: (* = (CHAR = P
                      REWR = UNSTABLE.VOWEL
                      UNSTABLE.VOWEL = E))
4--5 CREATOR: = 0
     FEATURES: (* = (CHAR = E))

5--6 CREATOR: = 0
     FEATURES: (* = (CHAR = L))

14--6 CREATOR: 12
      FEATURES: (* = (CHAR = L
                      REWR = UNSTABLE.VOWEL
                      UNSTABLE.VOWEL = E))
```

Figure 4-2:

Vi har sålunda en väg i charten (1 - 2 - 3 - 14 - 6) som svarar mot uppslagsordet i lexikonet.

```
ÄPPL  <& MORPH.CAT> :=: 'ROOT,
      <& ROOT> :=: 'ÄPPL,
      <& ROOT.CAT> :=: 'NOUN,
      (IF <* UNSTABLE.VOWEL>
         THEN <* UNSTABLE.VOWEL> = 'E,
              STORE,
              ADVANCE,
              PROCESS(SVE.DIC),
              MAJORPROCESS(COMPOUND)
         ELSE ...
```

Figure 4-3:

I 4-3 ges lexikonartikeln till roten 'äppl'. Som framgår av 4-3 så testar vi i lexikonartikeln på omskrivning samt flyktig vokal innan vi går vidare. (ELSE-alternativet gäller 'äpple, äpplet' etc.)

Ännu har vi inte berört hur vi får regeln UNSTABLE.VOWEL att aktiveras i vertex 3 i exemplet. Det sker utifrån en lexikonsökningstask på följande sätt.

4.1.1.1 Aktioner i samband med lexikontasks
För varje lexikon i UCP specificeras 4 obligatoriska 'entries', nämligen

1. SEARCH-KEY
2. FORWARD-ACTION
3. GATHERING-RULE
4. TOTAL-HIT-RULE-DEFAULT

I SEARCH-KEY anges på vilket attribut jämförelsen skall ske. I våra applikationer har vi arbetat med CHAR som söknyckel.

I FORWARD-ACTION anges hur bearbetningen skall gå vidare då man funnit överensstämmelse mellan en karaktär i analysuttrycket och en karaktär i bokstavsträdet (lexikonet). I våra applikationer har vi där ADVANCE som aktion, vilket innebär att man stegar sig fram till nästa karaktär.

I GATHERING-RULE kan man ange övriga aktioner som man vill utföra vid en träff.

I TOTAL-HIT-RULE-DEFAULT kan man specificera vad som skall ske i det fall uppslagsordet inte har någon information associerad med sig. Vi har där STORE.

I GATHERING-RULE för SVE.DIC (vårt svenska huvudlexikon) ger vi signalen för att UNSTABLE.VOWEL regeln skall aktiveras.

4-4 visar GATHERING-RULE för SVE.DIC.

```
GATHERING-RULE
    ADVANCE,
    (<* CHAR :MODE> = 'STOP / <* CHAR> = 'S / <* CHAR> = 'M),
    PROCESS(SVE.PHON);
```

Figure 4-4:

SVE.PHON är namnet på en svensk fonologisk grammatik, vars första regel är UNSTABLE.VOWEL. PROCESS(SVE.PHON) leder till ett försök att tillämpa UNSTABLE.VOWEL på den karaktär som uppfyller villkoren i satsen ovan.

5. Indexgrammatiken

Under analysen av ett visst uttryck införs kontinuerligt nya bågar i charten och inga bågar raderas ut.

Om vi t.ex. gör en satsanalys, kommer charten till slut att innehålla många olika slags bågar. Det är fråga om edgar som representerar (omskrivna och icke omskrivna) karaktärer, morfer, ordformer och syntaktiska konstituenter.

Låter vi den allmänna taskgenereringsprincipen verka utan restriktioner, så kommer det att genereras många inherent improduktiva tasks. Det är t. ex. meningslöst att generera en lexikonsökningstask där den inaktiva edgen representerar en syntaktisk konstituent.

Vi måste ha en möjlighet att kunna uttrycka vilka slags processer som skall verka på vilka slags lingvistiska enheter. Detta gör vi i en särskild grammatik, den s. k. indexgrammatiken. Indexgrammatiken är del av den språkspecifika informationen i en UCP parser.

I 5-1 ger vi några regler ur indexgrammatiken i vår svenska applikation.

```
SVE.DIC: CHAR;

NOUN: MORPH.CAT;

NP: CAT / SYN.CONST;

UNSTABLE.VOWEL: CHAR, NOT UNSTABLE.VOWEL;

...
```

Figure 5-1:

Den först regeln i 5-1 säger följande: Processering med avseende på lexikonet SVE.DIC skall endast gälla inaktiva edgar, vilka innehåller attributet CHAR, dvs. vid lexikonsökning i lexikonet SVE.DIC skall endast karaktär-bågar beaktas.

Den andra regeln säger, att processering med avseende på den grammatiska regeln NOUN1 endast skall gälla inaktiva edgar vilka innehåller attributet MORPH.CAT, dvs. vid interpretering av den grammatiska regeln NOUN skall endast morf-bågar beaktas och så vidare.

> Anm. UCP håller själv reda på vilka namn som refererar till lexikon och vilka som refererar till grammatikor och grammatiska regler.

6. Implementering

UCP består av ett kärnprogram, ett utskriftsprogram och en editor.

Kärnprogrammet anropas via funktionen PROCESS. Den tar tre argument, nämligen analysuttrycket (ett ord eller en hel sats omgiven av parenteser), namn på den överordnade grammatiken samt optionellt namnet på aktuell indexgrammatik.

T.ex.

```
(PROCESS (I denna film från Kanada får vi träffa en kvinna.)
         SVE.GRAM
         SVE.INDEX)
```

Bearbetningen startar med att ett initialt chart byggs upp. Därpå appliceras den första regeln i grammatiken på den första noden i charten. I vår svenska applikation lyder den första regeln

```
START.RULE
    PROCESS(SVE.DIC);
```

Vad som därpå sker bestäms av hur analysuttrycket ser ut samt av den språkspecifika informationen (lexika och grammatikor). Kontrollen över bearbetningen är sålunda decentraliserad. Den vilar på grammatikor och lexika.

Om analysen lyckas returnerar PROCESS värdet True i annat fall NIL. Analysresultatet skrivs ut med funktionen PRINTRESULT av CHART.

UCP körs vanligen interaktivt. Efter ett PROCESS-anrop hamnar man automatiskt i TRACE-mode. Därvid har man möjlighet att få de olika bearbetningsstegen (tasken) utskrivna på bildskärmen. Man kan också välja att gå ur TRACE-mode och enbart titta på slutresultatet. För de olika kommandona i 'trace-paketet', se Carlsson 81.

Med hjälp av utskriftsprogrammet TYPE kan man skriva ut såväl lexika (eller enskilda lexikonartiklar) som grammatikor (eller enskilda regler). Man kan också göra t. ex. (TYPE K) och få information om globalt lagrade egenskaper avseende karaktären K. UCP håller själv reda på vad som är lexikon, vad som är grammatik och vad som är en enskild karaktär.

Programmet EDIT är ett mycket viktigt hjälpprogram. Via EDIT editerar man i förefintliga lexika och grammatikor samt får hjälp med att bygga upp nya sådana. EDIT körs interaktivt.

UCP är implementerad i ett subset av BBN INTERLISP (se Teitel 74), nämligen Uppsala INTERLISP (se UDAC: 75). Den körs mot en IBM 370.

Angående implementeringen, se Carlsson 80 och 81.

7. Aktuella tillämpningar

Många ideer rörande utformningen av UCP har kommit fram under arbetet på en processmodell för ordigenkänning i finska (se Sågvall 78 och 80b).

Huvudapplikationen rör grammatisk analys av svenska. Den är bl. a. att betrakta som ett delprojekt inom projektet Datorsimulering av textförståelseprocessen. Projektet ingår i forskningsprogrammet för Uppsala Programming Methodology and Artificial Intelligence Laboratory (UPMAIL), delvis finansierat av Styrelsen för Teknisk Utveckling (STU). Projektet bedrivs i samarbete med Mats Carlsson, UCDL och UPMAIL.

Arbete pågår också med att definiera lexika och grammatik för automatisk morfologisk analys av serbo-kroatiska i UCP. Detta görs inom ramen för JUBA-projektet vid Slaviska institutionen i Lund (se Đurović 79).

UCP kan betraktas som ett redskap för studier av lingvistiska processer och deras samspel.

8. Planerad vidareutveckling

Den mest näraliggande vidareutvecklingen av UCP rör task-kommunikation och i anslutning därtill utarbetande av en metodik för sofistikerad schedulering och selectering av tasks.

9. Referenser

Carlsson 1980
M. Carlsson
Uppsala Chart Parser 1 - Program documentation
Report no. UCDL-R-80-2
Center for Computational Linguistics
Uppsala University, 1980

Carlsson 1981
M. Carlsson
Uppsala Chart Parser 2 - Program documentation
Report no. UCDL-R-81-1
Center for Computational Linguistics
Uppsala University (under utgivning)

Ďurovič 79
Durovic, L. & Stankovski, M.
Hemspråksutvecklingen hos serbokroatisk/kroatoserbisk-talande barn i Sverige,
Slaviska institutionen
Lund, 1979

Kaplan 1973
R. Kaplan
A General Syntactic Processor.
I: Rustin, R. (red.),
Natural Language Processing.
New York: Algorithmic Press, 1973
ss. 193-241

Kay 73 M. Kay
The MIND System
I: Rustin, R. (red.)
Natural Language Processing
New York: Algorithmic Press, 1973
ss. 155-188

Kay 75 M. Kay
Syntactic Processing and the Functional Sentence Perspective.
I: Schank, R. & Nash-Webber, B.L. (red.)
Theoretical Issues in Natural Language Processing (TINLAP-1)
Cambridge, Mass., 1975
ss.6-9

Kay 77a M. Kay
Morphological and Syntactic Analysis.
I: Zampolli, A. (red)
Linguistic Structures Processing
Amsterdam: North-Holland, 1977
ss. 131-234

Kay 77b M. Kay
Reversible Grammar.
I: Handbook for the 1977 Nordic Summer School in Computational Linguistics
Palo Alto, Xerox PARC, 1977

Sågvall 77a
A. Sågvall Hein

Chartanalys och morfologi.
I: Rapporter från Språkdata 3.
Föredrag från en konferens i Göteborg 10-11 okt. 1977
ss. 87-93

Sågvall 77b
A. Sågvall Hein
An Approach to the Construction of a Text Comprehension System for X-ray Reports.
I: Schneider, W. & Sågvall Hein, A. (red.)
Computational Linguistics in Medicine
Amsterdam: North-Holland, 1977
ss. 91-99

Sågvall 78
A. Sågvall Hein
Finnish Morphological Analysis in the Reversible Grammar System.
I: Proceedings from the 7th International Conference on Computational Linguistics, Bergen, 14-18 August, 1978
(under utgivning)

Sågvall 80a
A. Sågvall Hein
An Outline of a Computer Model of Finnish Word Recognition.
Fenno-ugrica suecana 3/1980
Finsk-ugriska institutionen
Uppsala Universitet, 1980

Sågvall 80b
A. Sågvall Hein
An Overview of the Uppsala Chart Parser Version 1 (UCP-1)
Report no. UCDL-R-80-1
Center for Computational Linguistics
Uppsala Universitet, 1980

Teitel 74
Teitelman, W., Hartley, A. K., Goodwin, J. W., Lewis, D. C., Bobrow, D. G., Masinter, L. M.
INTERLISP reference manual.
Palo Alto, Xerox PARC, 1974

UDAC: 75 INTERLISP/360 and /370 user reference manual.
Uppsala University Data Center
Uppsala, 1975

Knut Hofland
NAVFs edb-senter for humanistisk forskning
Postboks 53
N-5014 Bergen-Universitet

GRAMMATISK MERKING AV LOB-KORPUS.

Innledning

Denne artikkelen omtaler det pågående arbeid med grammatisk merking (på ordklassenivå) av LOB (Lancaster-Oslo/Bergen) korpus. Dette er et samarbeidsprosjekt mellom Britisk Institutt i Oslo, NAVFs edb-senter i Bergen og universitetet i Lancaster. På grunn av reduserte bevilgninger vil hovedtyngden av arbeidet bli gjort i Lancaster.

LOB korpus er en britisk-engelsk parallell til det amerikanske Brown korpus som ble ferdig i 1967. For opplysninger om Brown korpus se Francis (1979) og Kucera & Francis (1967). Arbeidet med LOB korpuset ble startet i Lancaster i 1970 av Geoffrey N. Leech og ble fullført i Norge ved et samarbeid mellom Stig Johansson, Oslo og NAVFs edb-senter, Johansson et al. (1978). Til LOB korpus er det laget en KWIC-konkordans på mikrokort, Hofland & Johansson (1979) og det er under utgivelse bearbeidete ordlister til materialet bl. a. med sammenligninger med Brown korpus, Hofland & Johansson (1981).

Brown og LOB korpora inneholder hver 500 tekstutsnitt på omlag 2000 ord, totalt 1 million ord, fra forskjellige typer trykket tekst (inndelt i 15 kategorier) utgitt i 1961. De to korpora har samme oppbygging og egner seg derfor godt til sammenligninger. I 1979 ble det gjort ferdig en grammatisk merket versjon av Brown korpus. En analyse av dette materialet av Kucera & Francis er under utgivelse.

Arbeidet med å finansiere et tilsvarende prosjekt for merking av LOB korpuset ble startet i 1979. Forarbeidet ble påbegynt i 1980 og fra 1981 er det bevilget midler, i England også for 1982. Merkingen av teksten gjøres av følgende grunner:

a) den gjør det mulig å søke i tekst både etter kombinasjon av ord og grammatiske koder
b) separerer homografer
c) gjør senere lemmatisering av materialet enklere
d) gjør mulig automatisert syntaktisk analyse av materialet

Metode

For også å kunne sammenligne de merkete korpora vil det samme

kodesystemet som ble brukt ved merkingen av Brown korpus bli brukt ved merkingen av LOB korpus. Merkesystemet gir i hovedsak koder basert på ordklassetilhørighet og kodene kan deles inn i 5 typer (se fullstendig kodeliste i appendix a):

a) åpne ordklasser
b) funksjonsord (lukkete ordklasser)
c) viktige enkle ord som not, be, have, do
d) tegnsetting som kan ha syntaktisk informasjon
e) bøyingsmorfemer til a) og c)

Metoden som skal brukes er en modifisert utgave av den som ble brukt ved merking av Brown korpus, Greene & Rubin (1971). Denne består av 5 trinn:

1) pre-editering av teksten
2) oppslag i ordliste (Brown: 2860 ord, 61% med en kode)
3) oppslag i endelsesliste basert på inntil 5 siste tegn, (Brown: 446 endelser, 51% med en kode)
4) kontekstregler innen setning, brukt på ord som har fått flere koder etter trinn 2 og 3 (Brown: 77% rette koder)
5) Manuell entydiggjøring og oppretting av feile koder

Arbeidet med å tilpasse endelseslisten til britisk-engelsk og utvidelse av denne og ordlisten er beskrevet av Mette-Cathrine Jahr i neste artikkel.

Ved modifikasjon av metode og tabeller har følgende materiale vært tilgjengelig:

1) Det merkete Brown korpus, og følgende lister
   a) alfabetisk ordliste med tilhørende koder
   b) finalalfabetisk ordliste med tilhørende koder
   c) alfabetisk liste over koder med tilhørende ord
   d) maskinell produsert endelsesliste basert på endelser med koder som forekommer i minst 5 ord
2) LOB korpus med ordlister og KWIC-konkordans
3) Finalalfabetisk ordliste til både Brown og LOB korpus (totalt ca. 75 000 grafiske ord)
4) Liste av ord som forekommer i begge korpora (ca. 25 000)
5) Liste av ord som bare forekommer i LOB korpus (ca. 25 000)

I forbindelse med forberedelsen til prosjektet er en del vanlige homografer som may, to, that, merket manuelt utifra den eksisterende KWIC-konkordans. Merkene overføres til teksten og vil særlig få betydning ved bruk av kontekstreglene.

Pre-editering av teksten.

Ved kodingen av Brown korpus ble teksten redigert ved at en del tegnsetting og koder ble fjernet. Noen ord ble slått sammen til enheter som f.eks. navn på personer og organisasjoner, datoer ol. Disse fikk spesielle koder tilordnet. Stor forbokstav ble bare beholdt for egennavn. I LOB korpuset kan en del av de eksisteren-

de koder brukes i dette arbeidet. Det gjelder markering av setningstart, koder for forkorting og utenlandske ord, overskrifter ol.

Endelseslisten

Denne listen inneholder tradisjonelle avledningss- og bøyingssuffikser, men også endelser som kan identifisere en ordklasse uten at den har noen grammatisk funksjon. I endelseslisten ønsker en "de (lengste) endelser som identifiserer færrest mulig ordklasser (helst bare en) og så mange ord som mulig". Eksempel fra listen:

```
   IVE      --> JJ - NN    (adjektiv eller substantiv)
 CEIVE      --> VB         (verb)
  RIVE      --> VB
  SIVE      --> JJ
  TIVE      --> NN - JJ
  VIVE      --> VB
```

Den lengste endelse som fins i listen brukes.

Ordlisten

Ordlisten inneholder funksjonsord og ord som ikke følger endelseslisten eller de spesialbehandlinger som foretas. Videre inneholder den alle ord med frekvens 50 eller mer i Brown korpus. Ved merking av LOB korpuset er hele ordlisten fra det merkete Brown korpus tilgjengelig. Men noen av ordene der som bare har fått en kode kan allikevel være homografer slik at ord fra denne ordlisten må spesialbehandles. En mulighet er i tillegg å slå opp i endelseslisten for å finne eventuelle andre koder som kan forekomme.

Spesialbehandling av en del ord

Programmet til Greene & Rubin foretar først oppslag i ordliste. Hvis ordet forekommer der velges koden(e) fra ordlisten, ellers foretas en sjekk etter spesielle ord.

a) ord som begynner med $ merkes NNS.
b) For ord som inneholder apostrof fjernes endingen (N'T, 'LL, 'RE, 'VE, 'D, 'S, ') og resten av ordet sjekkes. Koder for endingen henges på som en tilleggskode.
c) koder som er satt på under pre-editering overføres til de enkelte ord
d) ord med stor førstebokstav får kode NP.
e) ord med bindestrek splittes opp i to deler. Som regel får ordet koden til siste ledd. En del kombinasjoner av koder og

endelser for de to delene behandles spesielt.
f) Ord som starter med et siffer får kode CD. Tall skrevet med bokstaver må stå i ordliste unntatt -TEEN.
g) Ord som slutter på ST, RD, ND og som har siffer som første tegn får kode OD.
h) Ord på formen UN...ED merkes JJ. Der hvor formen UN... er et verb må dette stå i ordlisten.

Ord som ikke slutter på enkel S sjekkes mot endelseslisten. Hvis endelsen ikke fins der, får ordet kode NN - VB - JJ.

Ord som slutter på S får spesialbehandling. Ordet gies foreløpig kode NNS eller VBZ. S'en fjernes fra ordet og dette sjekkes i ordlisten. Hvis det fins der velges en av mulighetene, NNS hvis NN og ikke VB fra ordlisten, VBZ hvis VB og ikke NN fra ordlisten.

Dersom ikke ordet uten S fins i ordlisten sjekkes endelsen til ordet.

a) ved -ING gis kode NNS
b) ord som ender på konsonant sjekkes i endelseslisten. Ord som slutter på konsonant+S må stå i ordlisten
c) -IE forandres til Y og ordet sjekkes i ordliste
d) ved ord som slutter på -SES, ZES, HES, XES sjekkes ordet uten -ES i ordlisten
e) ord som slutter på I(S) gis kode NN, EAU(S) kode NNS, OU(S) kode JJ, U(S) kode NN
f) ellers sjekkes mot ordliste/endelsesliste

I tillegg til denne spesialbehandlingen av S kan det være aktuellt også å fjerne endelser som -ISH, -ED, -(E)R, -LY og sjekke resten av ordet mot ordliste og endelsesliste. Dersom f.eks. et ord slutter på -ISH og resten av ordet ikke er et verb, så gis ordet kode JJ.

Kontekstregler

Når alle ordene i en setning har fått koder, skal kontekstreglene velge ut riktig kode der et ord har fått flere koder, basert på kodene til de omsluttende ord (for Brown korpus inntil 2 ord på hver side). For at reglene skal kunne brukes, må et eller flere av de omsluttende ord ha en entydig kode. Kontekstreglene kan være av 2 typer:

1) negative, f.eks. at VB ikke kan etterfølge AT

   AT ? --> -VB

2) positive, f.eks etter en modal kan det komme verb i grunnform

   MD ? --> VB

Ved utarbeidingen av kontekstreglene til Brown korpuset ble 900 setninger merket manuelt. Det ble kjørt ut sorterte lister over kombinasjoner av koder og kontekstreglene ble satt opp etter dette grunnlaget. Kontekstreglene til LOB korpuset vil bli laget på grunnlag av hele det merkete Brown korpuset.

Totalt kan det være 8 mulige regler for et ord med flere koder

```
1)  A  B  ?  C  D
2)  A  B  ?  C
3)     B  ?  C  D
4)     B  ?  C
5)  A  B  ?
6)        ?  C  D
7)     B  ?
8)        ?  C
```

Dersom ordene i posisjon A, B, C eller D har flere koder kan regelen ikke brukes. Reglene prøves i rekkefølge 1-8. Hvis en regel løser opp en tvetydighet prøves de andre reglene om igjen.

Eksempel:

```
when            WRB
the             AT
boy's           NN$   NN+BEZ   NN+HVZ
old             JJ
horse           NN    VB
is              BEZ
here            RN
```

I første omgang fins det ingen regel for den første tvetydigheten. Til den neste kan regelen

```
?  BEZ      --> -VB
```

brukes. Nå kan imidlertid regelen

```
?  JJ  NN   --> NN$
```

brukes og begge tvetydigheter er oppløst.

Referanser

Francis, W. Nelson. 1979. Manual of Information to Accompany a Standard Sample of Present-Day Edited American English, for Use with Digital Computers. Rev. ed. Providence, RI: Department of Linguistics, Brown University.

Greene, BarbaraB. & Rubin, Gerald M. 1971 Automatic Grammatical Tagging of English, Providence, RI: Department of Linguistics, Brown University

Hofland, Knut & Stig Johansson. 1979. Microfiche concordance of the Lancaster-Oslo/Bergen Corpus. Bergen: NAVFs edb-senter for humanistisk forskning.

Hofland, Knut & Stig Johansson. 1981. Word Frequencies in British and American English. Bergen: NAVFs edb-senter for humanistisk forskning

Johansson, Stig, Leech, Geoffrey N. & Helen Goodluck. 1978. Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers. Engelsk Institutt, Universitetet i Oslo.

Kucera, Henry & W. Nelson F rancis. 1967. Computational Analysis of Present-Day American English. Providence, RI: Brown University Press.

Kucera, Henry & W. Nelson Francis. (under utgivelse) Frequency Analysis of English Usage: Vocabulary and Grammar.

Appendix A

LIST OF TAGS FOR THE BROWN CORPUS

.	., ;, ?, !
(	(
*	NOT, N'T
)	)
-	dash
,	,
:	:
ABL	pre-qualifier (QUITE, RATHER)
ABN	pre-quantifier (HALF, ALL)
ABX	pre-quantifier/double conjunction (BOTH ... AND)
AP	post-determiner (MANY, SEVERAL, NEXT)
AT	article (A, THE, NO)
BE	BE
BED	WERE
BEDZ	WAS
BEG	BEING
BEM	AM
BEN	BEEN
BER	ARE, ART
BEZ	IS
CC	coordinating conjunction (AND, OR)
CD	cardinal numeral (ONE, 1)
CI	conjuction/preposition (AFTER)
CS	subordinating conjunction (IF, ALTHOUGH)
DO	DO
DOD	DID
DOZ	DOES
DT	singular determiner (THIS, THAT)
DTI	singular or plural determiner (SOME, ANY)
DTS	plural determiner (THESE, THOSE)
DTX	determiner/double conjunction (EITHER ... OR)
EX	existential THERE
FW	foreign word (hyphenated to regular tag)
HD	word occurs in headline (hypenated to regular tag)
HV	HAVE
HVD	HAD (past tense)
HVG	HAVING
HVN	HAD (past participle)
IN	preposition (AMONG, BETWEEN)
JJ	adjective
JJR	comparative adjective
JJS	semantically superlative adjective (CHIEF, MAIN)
JJT	morphologically superlative adjective (BIGGEST)
MD	modal auxiliary
NC	cited word (hyphenated to regular tag)
NN	singular or mass noun
NN$	possesive singular noun
NNS	plural noun
NNS$	possesive plural noun
NP	proper noun (may be hyphenated to other tag)

```
NP$  possesive singular proper noun
NPS  plural proper noun
NPS$ possesive plural proper noun
NR   adverbial noun (HOME, TODAY, WEST)
OD   ordinal numeral (FIRST, SECOND)
PN   nominal pronoun (EVERYBODY, NOTHING)
PN$  possesive nominal pronoun
PP$  possesive personal pronoun (MY, OUR)
PP$$ second possesive personal pronoun (MINE, OURS)
PPL  singular reflexive (intensive) pronoun (MYSELF)
PPLS plural reflexive (intensive) pronoun (OURSELVES)
PPO  objective personal pronoun (ME, HIM, IT, THEM)
PPS  3rd. sing. nominative personal pronoun (HE, SHE, IT, ONE)
PPSS other nominative personal pronoun (I, WE, THEY, YOU)
QL   qualifier (VERY, LOTS, FAIRLY)
QLP  post-qualifier (EASY, ENOUGH)
RB   adverb
RBR  comparative adverb
RBT  superlative adverb
RI   adverb/preposition (portmanteau) (ABOVE, ALONG)
RIP  adverb/preposition/particle (portmanteau) (DOWN, IN)
RN   nominal adverb (HERE, THEN, INDOORS)
RP   adverb/particle (BACK, AWAY)
TO   infinitive marker TO
TT   word occurs in title (hyphenated to regular tag)
UH   interjection
VB   verb, base form
VBD  verb, past tense
VBG  verb, present participle, gerund
VBN  verb, past participle
VBZ  verb, 3rd. sing. present
WDT  wh-determiner (WHAT, WHICH)
WP$  possesive wh-pronoun (WHOSE)
WPO  objective wh-pronoun (WHOM, WHICH, THAT)
WPS  nominative wh-pronoun (WHO, WHICH, THAT)
WQL  wh-qualifier (HOW)
WRB  wh-adverb (HOW, WHEN, WHERE)

------------------------

N'T  *
'LL  +MD
'RE  +BER
'VE  +HV
'D   +MD  +HVD  +DOD
'S   $    +BEZ  +HVZ
S'   $
```

Mette-Cathrine Jahr
Stig Johansson
Britisk Institutt
Universitetet i Oslo

GRAMMATISK MERKING AV THE LANCASTER-OSLO/BERGEN CORPUS: ORDKLASSEBESTEMMELSE VED HJELP AV ORDSLUTT

1 Målsetting

I forbindelse med prosjektet "Grammatical Tagging of the Lancaster-Oslo/Bergen Corpus" har vi i Oslo spesielt konsentrert oss om å revidere Greene og Rubins suffiksliste og ordliste for Brown Corpus.[1] Våre reviderte lister tar hensyn til både Brown Corpus og Lancaster-Oslo/Bergen LOB) Corpus, dvs. tilsammen ca. 2 millioner ord løpende tekst. Riktignok er dette et relativt beskjedent materiale i forhold til alt som finnes av engelske tekster, men vi tror likevel at de resultatene vi er kommet frem til, er ganske allmenngyldige.

2 Problemstilling

For den som ønsker å utføre en automatisk grammatisk analyse, byr engelsk på spesielle problemer, både fordi språket inneholder et usedvanlig stort antall homografer, og fordi det så godt som fullstendig mangler distinktive bøyningsendelser. Ikke desto mindre viser Greene og Rubins arbeid (1971) klart at det i stor utstrekning lar seg gjøre å bestemme ordklasser ut fra avledningsendelser og andre typer sluttsekvenser. (Betegnelsene "suffiks" og "endelser" vil her bli brukt i betydningen ordslutt eller sluttsekvens.)

3 Materiale og metode

Da Greene og Rubin (1971) laget sin suffiksliste, brukte de foruten en final-alfabetisk ordliste for Brown Corpus (ca. 50.000 ordtyper), final-alfabetiske ordlister fra Dolby og Resnikoff (1967). Vi har benyttet en lignende fremgangsmåte i arbeidet med å revidere suffikslisten. Følgende materiale ble brukt:

A: En final-alfabetisk ordliste over de tilsammen ca. 75.000 ordtypene (grafiske ord) som finnes i Brown Corpus og LOB Corpus.[2]

B: En frekvensliste over grammatiske koder som forekommer ved hyppige endelser (fra én til fem bokstaver), basert på den grammatisk merkede versjonen av Brown Corpus (heretter kalt suffiks/kode-listen). Noen eksempler:[3]

IVE	JJ	(= adjektiv)	254
	NN	(= substantiv, sg.)	50
	NP	(= egennavn)	25
	VB	(= verb, infinitiv)	20
	CD	(= grunntall)	10
RIVE	VB	(= verb, infinitiv)	7
SIVE	JJ	(= adjektiv)	65
	NN	(= substantiv, sg.)	6
TIVE	JJ	(= adjektiv)	182
	NN	(= substantiv, sg.)	36
	NP	(= egennavn)	16

Nedre frekvensgrense ble satt til 5, dvs. en grammatisk kode måtte opptre sammen med et bestemt suffiks minst fem ganger for å bli tatt inn i listen.

C: Final-alfabetiske ordlister fra Dolby og Resnikoff (1967), som bygger på Shorter Oxford English Dictionary og Merriam Webster New International Dictionary.

D: Forskjellige verker som behandler morfologi og orddannelse i engelsk, spesielt Ljung (1974) og Marchand (1969).

Vi tok vårt utgangspunkt i den final-alfabetiske ordlisten basert på de to korpusene (LOB og Brown) og begynte med å merke ordene etter Greene og Rubins suffiksregler og skille ut alle unntak. Ved å sammenligne antall ord som dekkes av en regel med antall unntak, kunne vi bedømme hvor effektiv hver enkelt regel var. Underveis ble vi oppmerksomme på feil og uoverensstemmelser. Lite effektive regler ble avslørt og nye regelmessigheter ble oppdaget. Som hjelpemidler i arbeidet med å finne frem til nye regelmessige endelser, benyttet vi også listene nevnt under punkt B, C og D ovenfor. Dette skulle gi arbeidet vårt større generell gyldighet.

4 Den reviderte suffikslisten

Resultatet av vårt arbeid er en revidert og utvidet suffiksliste over vel 600 suffikser med tilhørende grammatiske koder. I utgangspunktet mener vi at Greene og Rubins metode er god. Vi har derfor utarbeidet den reviderte suffikslisten etter deres prinsipper. De forandringene vi har gjort, berører ikke selve strukturen i merkeprogrammet deres. Under punkt 6 nedenfor vil vi imidlertid diskutere noen mer radikale endringer som kanskje burde overveies nøyere.

4.1 Stryking av suffiksregler

Ca. 80 av Greene og Rubins suffiksregler er blitt tatt ut av listen. I eksemplene nedenfor er endelsene skrevet forfra på

vanlig måte slik at de er lettere å identifisere og lese:

LB -→ NN: Regelen dekker bare tre ord (Dekalb, Kolb, bulb), hvorav to er egennavn som vi ikke behøver å ta hensyn til i suffikslisten. Det tredje ordet dekkes av en annen eksisterende regel: B -→ NN-VB.

IELD -→ NN-VB: Regelen dekker bare fem ord (shield, windshield, sunshield, wield, yield). Vi lar en annen eksisterende regel behandle dem: LD -→ NN-VB.

ZARD -→ NN: Det er bare fem vanlige ord som slutter på ZARD,[4] hvorav to er unntak fra regelen. De to unntakene settes inn i ordlisten, de øvrige blir ivaretatt av en annen eksisterende regel: RD -→ NN.

RDE -→ NN: Foruten egennavn er det bare ett vanlig ord som dekkes av denne regelen (horde). Vi lager en ny, effektiv suffiksregel: DE -→ NN, som dekker 42 tilfeller og bare gir ett unntak.

UGE -→ NN: Regelen dekker følgende vanlige ord: gauge, refuge, centrifuge, huge, deluge, gouge, rouge. Ett av disse er et adjektiv, de andre kan være både substantiv og verb. Vi setter huge inn i ordlisten og lar de øvrige bli tatt hånd om av en annen eksisterende regel: GE -→ NN-VB.

EPE -→ NN-VB
OUPE -→ NN : Disse reglene behandler bare tre vanlige ord (hvorav ett blir galt kodet): crepe, cantaloupe, troupe. Vi innfører en ny regel: PE -→ NN som dekker 24 ord og gir ett unntak.

GNE -→ NN: Det eneste vanlige ordet som ender på GNE er champagne. Vi innfører en ny regel: NE -→ NN. Denne regelen dekker 46 ord og gir ingen unntak.

EFY -→ VB: To vanlige ord har denne sluttsekvensen, ett av dem et adjektiv: defy og beefy. Vi setter begge i ordlisten.

Noen av Greene og Rubins suffiksregler er høyst merkelige, f.eks. LYE -→ JJ (adjektiv) og LOREN -→ NNS (substantiv, plural). Sekvensen LYE finner vi bare i sitater fra eldre engelsk, og LOREN forekommer bare som et egennavn.

De suffiksene vi har fjernet, dekker svært få ord og ville ofte ha ført til at ord var blitt galt kodet. Som eksemplene ovenfor viser, har vi latt de berørte ordene bli behandlet (1) av andre eksisterende regler, (2) ved å innføre nye og mer effektive regler, eller (3) ved å sette de relevante formene inn i ordlisten.

4.2 Endring av grammatiske koder

I ca. 25 tilfeller der de opprinnelige suffiksreglene ble beholdt,

endret vi de grammatiske kodene. Koder ble fjernet i følgende tilfeller:

		Fjernet kode	Effektivitet [5]
AD	-→ NN-JJ	JJ	67/ 8
SIDE	-→ JJ-NN	JJ	61/10
LE	-→ NN-VB	VB	164/ 2
OLE	-→ NN-VB	VB	61/ 5
TIME	-→ NN-JJ	JJ	61/ 5
RNE	-→ NN-VB	VB	29/ 0
ITE	-→ NN-VB-JJ	VB	140/14
H	-→ NN-VB	VB	122/12
PH	-→ NN-VB	VB	15/ 1
TEN	-→ NN-VB	NN	36/ 6
IN	-→ NN-VB	VB	400/21
ON	-→ NN-VB	VB	393/30
O	-→ NN-VB	VB	879/44
ER	-→ NN-VB-JJR-RBR	RBR	453/17
IR	-→ NN-VB	VB	34/ 4
UENT	-→ NN-JJ	NN	15/ 3
EST	-→ JJT-RBT	RBT	267/61

Som vi ser av forholdet mellom antall behandlede ord og antall unntak, virker de reviderte reglene ganske godt.[6] Etter mye nøling bestemte vi oss for å stryke RBR (= adverb i komparativ form) og RBT (= adverb i superlativ form) fra kodesettet for ER og EST. Tallene i suffiks/kode-listen taler imidlertid for seg selv:

ER	NN	(= substantiv, sg.)	913
	VB	(= verb, inf.)	123
	JJR	(= adjektiv i komparativ form)	140
	RBR	(= adverb i komparativ form)	24
EST	JJT	(= adjektiv i superlativ form)	143
	RBT	(= adverb i superlativ form)	9

I grammatikker (bl.a. Quirk et al. 1972:294) påpekes det dessuten at det bare er et lite antall adverb som kompareres med ER og EST. Komparativ- og superlativformer på ER og EST som kan være adverb, ble satt inn i ordlisten.

I seks tilfeller føyde vi til koder:

		Tilføyet kode	Effektivitet
AID	-→ VBN-VBD	JJ	33/ 4
WARD	-→ RB	JJ	44/13
NINE	-→ JJ	NN	16/ 1
BORNE	-→ VBN	JJ	10/ 0
ESQUE	-→ JJ	NN	11/ 1
SIZE	-→ NN-JJ	VB	20/ 1

En av våre hovedregler har vært ikke å sette til nye koder fordi dette innebærer at reglene blir mindre presise. Vi har i stedet foretrukket å spesifisere unntak i ordlisten. For suffiksreglene nevnt ovenfor, ville antall unntak imidlertid ha blitt uakseptabelt høyt hvis vi ikke hadde satt inn de nye kodene. I to tilfel-

ler var vi i sterk tvil om hvorvidt vi skulle føye til nye koder eller ikke:

ED	-→ VBN-VBD	+JJ?
ING	-→ VBG	+NN-JJ?

Siden ord på ED ofte er adjektiver og de på ING ofte adjektiver eller substantiver, var det fristende å sette til de kodene vi har ført opp ovenfor med spørsmålstegn. Når vi allikevel bestemte oss for ikke å gjøre det her, var det dels på grunn av at verbformene av slike ord forekommer langt hyppigere, dels fordi vi ikke ønsket å ødelegge mulighetene for en sammenligning mellom LOB og Brown på dette punktet.

I følgende tre tilfeller endret vi de grammatiske kodene helt:

	Ny kode	Effektivitet
UND -→ JJ	NN	16/3
EDE -→ NN	VB	13/2
WHERE -→ NN	RB	7/1

Vi følte oss også fristet til å endre koden til NP i en del tilfeller:

	Effektivitet med NP-kode
WE -→ NN-VB	22/ 3
I -→ NNS	559/58
Z -→ NN-VB	139/11
HR -→ NN	8/ 0

Langt de fleste ord med disse sluttsekvensene er egennavn. Men siden egennavn behandles gjennom en spesialrutine i merkeprogrammet (se Hoflands artikkel), bestemte vi oss for ikke å endre kodene. To av de opprinnelige suffiksreglene ble sløyfet (for WE og HR), og ord som ikke er egennavn, ble satt inn i ordlisten. Regelen I -→ NNS ble beholdt, siden pluralformer av substantiver utgjør en ganske stor gruppe av ord på I. Vi beholdt også regelen Z -→ NN-VB. Denne regelen gir ingen unntak selv om den behandler svært få vanlige ord.

Meningen er at rekkefølgen av kodene skal gi uttrykk for den relative frekvensen, dvs. at den hyppigste og mest sannsynlige koden kommer først. Vi endret derfor rekkefølgen på kodene i følgende tilfeller:

	Rettet til
AC -→ JJ-NN	NN-JJ
ID -→ NN-JJ	JJ-NN
END -→ NN-VB	VB-NN
ISE -→ NN-VB	VB-NN
WISE -→ JJ-RB	RB-JJ
ERSE -→ NN-VB-JJ	VB-JJ-NN
ETE -→ NN-VB-JJ	JJ-NN-VB
TIVE -→ NN-JJ	JJ-NN

		Rettet til
SH	-→ VB-NN	NN-VB
ASH	-→ VB-NN	NN-VB
IAN	-→ NN-JJ	JJ-NN
HUMAN	-→ NN-JJ	JJ-NN
LIER	-→ NN-JJR	JJR-NN
NCT	-→ NN-JJ	JJ-NN
ANT	-→ NN-JJ	JJ-NN
NENT	-→ NN-JJ	JJ-NN
RENT	-→ NN-JJ	JJ-NN
TENT	-→ NN-JJ	JJ-NN
LY	-→ JJ-RB	RB-JJ
ERY	-→ JJ-NN	NN-JJ
ARRY	-→ NN-VB	VB-NN

For å bestemme rekkefølgen på kodene, brukte vi suffiks/kode-listen for Brown Corpus og den merkede final-alfabetiske ord-listen for både LOB Corpus og Brown Corpus (jfr. punkt 3 A og B). Den førstnevnte listen er ikke tilstrekkelig i seg selv fordi den bare tar hensyn til Brown Corpus, og fordi den gir frekvensene for hver sluttsekvens uten hensyn til suffiksreglene, slik at alle ord på f.eks. LERY og NERY er inkludert i "statistikken" for ord som slutter på ERY. I suffikslisten viser "statistikken" bare de forekomstene på ERY som ikke allerede er behandlet av reglene for LERY og NERY.

4.3 Innføring av nye suffiksregler

Den største endringen i forhold til Greene og Rubin er at vi har innført ca. 240 nye suffiksregler. I eksemplene nedenfor er nye suffikser understreket:

		Effektivitet
IC	-→ JJ-NN	256/ 1
RIC	-→ JJ	81/ 7
ISTIC	-→ JJ	81/ 1
D	-→ NN-VB	165/23
HOOD	-→ NN	23/ 1
NE	-→ NN	46/ 0
INE	-→ NN-VB	222/20
NINE	-→ NN-JJ	16/ 1
RINE	-→ NN-JJ	27/ 1
TINE	-→ NN-JJ	38/ 1
ZINE	-→ NN	8/ 0
TE	-→ NN	70/ 1
ATE	-→ VB-NN-JJ	266/ 2
CATE	-→ VB-NN	34/ 4
DATE	-→ VB-NN	14/ 1
GATE	-→ VB-NN	46/ 2
PHATE	-→ NN	9/ 2
IATE	-→ VB-NN	34/ 6
LATE	-→ VB-NN	68/ 4
TATE	-→ VB-NN	40/ 0
VATE	-→ VB	12/ 1

		Effektivitet
AL	-→ JJ-NN	811/24
ICAL	-→ JJ	243/ 6
IONAL	-→ JJ	106/ 6
UR	-→ NN-VB	38/ 0
EUR	-→ NN	19/ 0

De fleste av de nye suffiksreglene gir en mer presis ordklasse-angivelse. For RIC og ISTIC, for eksempel, trenger vi bare én kode, mens de ville få to koder hvis de skulle behandles av IC, som i Greene og Rubin. Innføringen av NE og TE illustrerer en annen endring i forhold til Greene og Rubin. Alle ord som slutter på NE og TE må de enten føre opp i ordlisten eller merke NN-VB-JJ, dvs. det kodesettet som blir gitt til alle ord som ikke dekkes av listene eller andre rutiner i merkeprogrammet.

En del nye suffikser er blitt tatt inn i listen som følge av at vi har forsøkt å være mer konsekvente enn Greene og Rubin (1971: 30) når det gjelder å gi så mange ord som mulig en entydig grammatisk kode så tidlig som mulig, selv om dette ikke nødvendigvis betyr en mer nøyaktig ordklasse-angivelse. Vi kan gi noen eksempler som illustrerer dette (nye sekvenser er understreket):

			Effektivitet
ELY	-→ RB		134/16
ATELY	-→ RB		42/ 2
IVELY	-→ RB		90/ 1
ALLY	-→ RB		98/ 3
CALLY	-→ RB		191/ 0
NALLY	-→ RB		54/ 0
RALLY	-→ RB		22/ 1
UALLY	-→ RB		27/ 0
IST	-→ NN		58/ 7
CIST	-→ NN		11/ 0
OGIST	-→ NN	(G&R: GIST)	23/ 0
LIST	-→ NN		55/ 2
MIST	-→ NN		13/ 1
NIST	-→ NN		53/ 0
RIST	-→ NN		33/ 2
OGY	-→ NN		6/ 0
OLOGY	-→ NN		54/ 0

Nye suffiksregler ble inkludert hvis de er svært effektive og/eller hvis de representerer produktive endelser. Vi kan ikke påstå at vi har vært hundre prosent konsekvente. Noen av våre suffiksregler kunne godt utelates og andre settes inn, men vi har ikke sett på dette som så forferdelig viktig, siden slike endringer ikke har noen innvirkning på valget av grammatisk kode.

5 Den reviderte ordlisten

Vår reviderte suffiksliste er supplert med en ny ordliste på noe under 5000 ord med tilhørende grammatiske koder. Ordlisten inneholder unntak fra suffiksreglene såvel som alle ord i LOB Corpus med en frekvens på 50 eller høyere. Vi har sett bort fra følgende unntak: utenlandske ord, arkaiske former, forkortelser og ord med unormale stavemåter. Videre har vi ikke tatt hensyn til egennavn og ord med bindestrek, fordi disse behandles ved hjelp av spesielle rutiner i merkeprogrammet.

I arbeidet med å skille ut unntak og bestemme grammatiske koder har vi hovedsakelig holdt oss til Longman Dictionary of Contemporary English (1978-utgaven), selv om vi også har brukt andre oppslagsverker, spesielt A Grammar of Contemporary English (1972).

For å øke nytten av den reviderte ordlisten, har vi i tillegg tatt med alle former fra Greene og Rubins ordliste som enten er unntak fra våre suffikslister, eller har en frekvens på 50 eller høyere (i Brown Corpus). Dette skulle gjøre arbeidet vårt mer anvendelig, idet det går ut over grensene for vårt primære siktemål, nemlig en grammatisk merking av LOB Corpus.

6 Diskusjon

Våre reviderte ord- og suffikslister tar hånd om en betydelig del av de tilsammen ca. 75.000 ordtypene i LOB Corpus og Brown Corpus. Hvis antall ord som behandles av suffiksreglene legges sammen med antall ord i ordlisten, kommer vi opp i over 50.000. De aller fleste av disse får bare én grammatisk kode. Imidlertid er det vanskelig å si helt nøyaktig hvor effektive de nye listene er. Tallet 50.000 er for så vidt for høyt, siden det inkluderer egennavn og ord med bindestrek, som behandles av spesialrutiner i merkeprogrammet (rutinen for ord med bindestrek gjør riktignok også bruk av listene).[7] På den annen side blir et stort antall av de resterende ca. 25.000 ordtypene tatt hånd om av andre spesialrutiner i programmet. Dette gjelder særlig grunntall og ordenstall, ord med apostrof og de fleste ord på S (hvor spesialrutinen også anvender ordlisten).

Ikke desto mindre, hva de nøyaktige tallene for "effektivitet" enn måtte være, er det helt klart at de er svært høye. Resultatene blir enda mer imponerende hvis vi, i stedet for å se på antall ordtyper som behandles, ser på det totale antall løpende ord i de to korpusene som blir tilfredsstillende merket ved hjelp av listene. Dette har vi ikke regnet ut, men tallet må være enormt høyt, fordi ordlisten omfatter alle høyfrekvente ord i begge korpusene, og vi vet at disse utgjør en meget stor del av en løpende tekst.

Det er imidlertid ikke nok at de reviderte listene kan behandle alle ord i de to korpusene. For at de skal kunne ha større allmenn interesse, må de også kunne anvendes på andre engelske tekster. Greene og Rubin (1971:41) hevder at det er "almost certain that any 'new' word contained in a sample of present-day

American English will be given its correct tag(s) by matching with the Suffix List". Vi bestemte oss for å teste vår reviderte suffiksliste på et utvalg av nye ord. Femten sider av A Dictionary of New English (1963-72) ble vilkårlig utvalgt, og alle nye former på disse sidene ble registrert. Egennavn, ord med bindestrek og forkortelser ble holdt utenfor testen. Videre unnlot vi å ta med tilfeller der gamle, etablerte ord hadde fått nye betydninger, med mindre dette falt sammen med nye grammatiske funksjoner. Av de 94 ordene vi registrerte, fikk 52 én enkelt, korrekt kode, 33 fikk to koder hvorav én var den riktige. Ett ord fikk tre koder inklusive den riktige, et annet ord ble ikke behandlet i det hele tatt, og syv ord ble galt kodet.[8]

Dette fører oss over til en vurdering av visse svakheter ved suffikslisten vår, og mulige forbedringer av merkeprogrammet.

Fire av de syv ordene som ble galt kodet, var adjektiver med prefikset multi-. Selv om prefikser vanligvis ikke er særlig pålitelige som ordklasse-indikatorer, er det åpenbart noen som ganske regelmessig opptrer ved spesielle ordklasser. Følgende prefikser innleder for eksempel bare substantiver og adjektiver:

micro-	pseudo-
multi-	semi-
non-	ultra-
proto-	vice-

En av Greene og Rubins spesialrutiner ser på en kombinasjon av et prefiks og et suffiks, nemlig UN...ED. Det måtte være mulig å inkorporere en prefiksrutine ved sekvenser som er typiske for spesielle ordklasser.

To av de nye ordene som ble galt kodet, er eksempler på substantiver brukt som verb (collage og network). Det er ikke lett å sikre seg riktig koding av slike tilfeller bare ved hjelp av endringer i suffikslisten eller andre deler av merkeprogrammet. Vi må innrømme at det alltid vil være en liten gruppe ord som ikke kan behandles riktig. Dette er prisen vi må betale for å oppnå en mer eksakt spesifisering i de aller fleste tilfellene.

En alvorlig innvending mot Greene og Rubins suffiksliste, såvel som vår egen reviderte liste, er at den ikke gjør bruk av visse opplagte regelmessigheter i engelsk orddannelse. Vi vet for eksempel at ord som ender på ER er komparativformer av adjektiver hvis resten av ordet er et adjektiv (old-er), og at de er substantiver hvis den øvrige delen av ordet er et verb (speak-er), etc. Greene og Rubin hadde en regel som gav disse ordene fire koder: ER -→ NN-VB-JJR-RBR. Dette betyr at store grupper av vanlige ord får flere koder, noe som vil føre til problemer på et senere stadium når kontekstreglene skal tre i funksjon. Det er også andre vanlige suffikser som ikke blir tilfredsstillende behandlet:

ED -→ VBN-VBD: Man utnytter ikke det faktum at ED-former er adjektiver når det ikke er et verb som går forut for ED (wick-ed).

EN → JJ-NN: Vi klarer ikke å fange opp at EN er en regelmessig adjektiv-endelse hvis det står et substantiv foran (wood-en, earth-en) og like regelmessig en verb-endelse hvis det kommer et adjektiv foran (black-en, stiff-en).

ISH → JJ-VB: Man kan faktisk si at hvis det som står foran ISH er et ord, er kombinasjonen et adjektiv (seven-ish, grey-ish, boy-ish).

LY → RB-JJ: Vi går glipp av muligheten til å trekke den opplagte slutning at et adjektiv fulgt av LY er et adverb.

Y → JJ-NN: Vi kan ikke få frem at et substantiv fulgt av Y normalt er et adjektiv (beef-y, hill-y).

Ved alle disse vanlige, produktive endelsene fører våre regler til unødig tvetydighet, selv om tvetydigheten i mange tilfeller reduseres ved at vi innfører lengre sekvenser med entydige koder (spesielt når det gjelder ord på LY). Det ville sannsynligvis være mye bedre å innføre flere avkortingsrutiner som svarer til den for ord som slutter på S (se Hoflands artikkel).

Vi har kommet med noen forslag til hvordan Greene og Rubins merkeprogram skulle kunne forbedres ved å innføre spesialrutiner som innebærer at programmet ser på prefikser og kutter av suffikser. Selv om denne delen av merkeprogrammet muligens blir noe mer komplisert enn nå,[9] vil dette delvis kunne kompenseres ved at man kan eliminere noen av de eksisterende spesialrutinene (jfr. Hoflands artikkel). Grunntall og ordenstall kunne for eksempel behandles av et lite antall suffiksregler, og mange vanlige former med apostrof kunne bare settes inn i ordlisten. Legg også merke til at spesialrutinen for ord som UN...ED ville bli overflødig hvis ED ble behandlet på en mer generell måte. Videre ville lengden på suffikslisten kunne reduseres betraktelig hvis suffiksavkorting ble brukt i større utstrekning.

Til slutt vil vi nevne at det er en mengde suffikser som nesten utelukkende forekommer i egennavn (BURGH, SHIRE, FORD, SKI, etc.). Det er klart at et stort antall egennavn kunne behandles ved at man innfører NP-suffiksregler. Dette ville redusere behovet for pre-editing (som den nåværende rutinen forutsetter).

7 Konklusjon

Selv om det finnes problemer i forbindelse med suffikslisten (og mulige forbedringer er blitt foreslått), må vår konklusjon bli at den i det store og hele fungerer meget bra.[10] Vi har sett at listen gir tilfredsstillende behandling av ordene i de to korpusene såvel som av de "nye" ordene i utdragene fra A Dictionary of New English. Kan dette bety at det er større overensstemmelse mellom ordform og ordklasse i engelsk enn det vi hittil har antatt? Det er helt opplagt at vi finner en ganske stor grad av regelmessighet så snart vi beveger oss utenfor den sentrale kjernen av høyfrekvente, først og fremst germanske, ord. Det er interessant å observere at denne regelmessigheten ikke begrenser

seg til suffikser i lingvistisk forstand, men også omfatter ordslutt i sin alminnelighet. Blant de "nye" ordene som fikk én eneste og riktig kode, finner vi former som duende, duka og dumdum. Regelmessigheter av denne art kunne bare oppdages ved hjelp av datamaskin og bør uten tvil utnyttes maksimalt i automatisk grammatisk analyse.

FOTNOTER

1 Knut Hoflands artikkel (i denne publikasjonen) om grammatisk merking av LOB Corpus tar også for seg Greene og Rubins "Automatic Grammatical Tagging of English" (1971) basert på Brown Corpus. Vi forutsetter derfor at dette er kjent stoff og henviser heretter til Hoflands artikkel når det gjelder spesielle punkter i Greene og Rubins merkeprogram.

2 Listene nevnt under punkt A og B er laget ved NAVF's EDB-senter for humanistisk forskning i Bergen, Norge. Programmerer: Knut Hofland.

3 Suffikser og grammatiske koder vil bli skrevet med store bokstaver, slik at det er lettere å skille dem ut fra den øvrige teksten.

4 Med "vanlige ord" mener vi enkeltord (uten bindestrek) med normal stavemåte, unntatt egennavn og forkortelser. Egennavn og ord med bindestrek behandles ved egne rutiner i programmet (se Hoflands artikkel).

5 "Statistikken" som er oppført ved hver regel i den reviderte suffikslisten angir hvor "effektiv" hver regel er. Tallene i venstre kolonne angir det totale antall ord med en bestemt sluttsekvens (som ikke allerede er behandlet av regler for lengre sekvenser), mens tallet i høyre kolonne angir antall unntak fra regelen.

6 De opprinnelige suffiksreglene ville også ha gitt en god del unntak.

7 Inkludert i dette tallet er også ca. 10.000 forekomster av ord på ED og ING som vi mener ikke blir tilfredsstillende kodet.

8 Resultatene av denne testen fikk oss til å foreta noen mindre endringer i suffikslisten, slik at antall galt kodede ord dermed er blitt redusert fra syv til fem (4 adjektiver med prefikset multi- samt network brukt som verb). En ny suffiksregel (É -→ NN-JJ) tar seg nå av det ordet som ikke ble behandlet i det hele tatt.

9 En del av disse rutinene vil muligens forutsette en større ordliste.

10 Den fullstendige suffikslisten vil bli publisert i Johansson og Jahr (under utgivelse).

REFERANSER

Barnhart, Clarence L., Steinmetz, Sol og Robert K. Barnhart. 1973. A Dictionary of New English 1963-1972. London: Longman.

Dolby, J.L. og H.L. Resnikoff. 1967. The English Word Speculum. Bind III og V. Den Haag: Mouton.

Greene, Barbara B. og Gerald M. Rubin. 1971. Automatic Grammatical Tagging of English. Providence, R.I.: Department of English, Brown University.

Johansson, Stig og Mette-Cathrine Jahr. "Grammatical Tagging of the Lancaster-Oslo/Bergen Corpus: Predicting Word Class from Word Endings". I Stig Johansson, utg., Computer Corpora in English Language Research. NAVF's EDB-senter for humanistisk forskning, Bergen. (under utgivelse)

Ljung, Magnus. 1974. A Frequency Dictionary of English Morphemes. Data linguistica 9, Universitetet i Göteborg. Stockholm: AWE/Gebers.

Longman Dictionary of Contemporary English. 1978. London: Longman.

Marchand, Hans. 1969. The Categories and Types of Present-Day English Word-Formation. Annen utgave. München: C.H. Beck'sche Verlagsbuchhandlung.

Quirk, Randolph, Greenbaum, Sidney, Leech, Geoffrey N. og Jan Svartvik. 1972. A Grammar of Contemporary English. London: Longman.

Anne Golden
Norskundervisningen for utenlandske studenter
Universitetet i Oslo

PRESENTASJON AV PROSJEKTET LÆREBOKSPRÅK

Når de fremmedspråklige elevene og de utenlandske studentene begynner henholdsvis på skolen og ved universitetet, starter de med generelle norskkurs. Disse kursene er generelle i den forstand at de bygger opp elevenes/studentenes norske syntaks ved hjelp av tekster som omhandler forskjellige dagligdagse situasjoner og hendelser. Bøkene som brukes er gjerne generelle innføringsbøker i norsk, dvs de er ikke beregnet på de enkelte elevers videre behov for spesiell norskopplæring. Når så elevene/studentene er blitt gode nok i norsk etter de forskjellige klassetrinns kriterier, blir de integrert i vanlige norske klasser, dvs de får undervisning på norsk sammen med norske elever/studenter. Men erfaringen viser at de får store vanskeligheter når de skal begynne å lese fagbøker. I grunnskolen er det bl a O-fagsbøkene [1] som volder problemer, også for de elevene som gjorde det bra i det generelle norskkurset. I ekstratimene i norsk blir det ofte til at lærerene leser O-fagsleksene sammen med elevene og forklarer det som står der. Flere lærere har klaget over at det ikke finnes noe materiell som elevene kan arbeide med slik at overgangen til de forskjellige fagene blir lettere for elevene.

Med prosjektet LÆREBOKSPRÅK vil vi prøve å lage et slikt støttemateriell for fagene fysikk, geografi og historie, slik at det blir bygget en bro mellom de generelle norskkursene og norsken i disse fagene i grunnskolen. Vi må derfor finne fram til hva det er som karakteriserer disse fagtekstene og hvorfor de er så vanskelige for de fremmedspråklige elevene. Spørsmålet blir da på hvilken måte disse tekstene skiller seg fra de tekstene som elevene har arbeidet med i de generelle språkkursene. For å gi et svar på dette må vi undersøke følgende:

1. Ordforrådet
2. Syntaksen i setningene
3. Oppbygningen av tekstene utover setningsplanet

Sannsynligvis er det flere faktorer som er med på å vanskeliggjøre overgangen til "faglekser". Selve læringssituasjonen blir forandret i og med at fagtekstene formidler kunnskap som skal læres. I lærebøkene i norsk som fremmedspråk er det språket selv som skal læres, men i fagbøkene er språket et middel til kunnskapstilegnelse. Med andre ord, når elevene begynner å "lese fag", får de en ny oppgave lagt til den de hadde da de bare arbeidet med det nye språket, nemlig at de skal kunne forstå og gjengi med egne ord den nye kunnskapen som teksten gir. For at elevene skal kunne mestre den nye oppgaven, slik at de kan

---

1) Orienteringsfag er historie naturfag og geografi integrert

lære fagene, er det nødvendig at de allerede så langt det er mulig behersker den type språk de finner i lærebøkene.

I første omgang har vi tatt for oss ordforrådet i O-fagsbøkene. I de generelle norskkursene er tekstene ofte bygd opp omkring sentrale sosiale situasjoner som 'på postkontoret', 'på trygdekontoret', 'på restaurant' osv og ordforrådet som blir presentert er følgelig det som brukes i slike situasjoner. Når vi sammenlikner dette ordstoffet med det i O-fagsbøker er det to typer av ordforråd som lett kan skape problemer for elevene:

Ordtype 1: Fagtermer
Ordtype 2: Ord som tilhører det allmenne ordforrådet,men som forekommer sjelden i de generelle språkkursene

Med ordtype 1 mener vi ord og uttrykk som inngår i selve faget og som enten er nye eller får et nytt og skarpere definert innhold også for de norske elevene i klassen. Eksempler fra en fysikkbok for 6. klasse: oppdrift, friksjon. Disse fagtermene er vanskelige, men fordi de er vanskelige også for de norske elevene, vil de oftest bli gjennomgått og forklart av faglæreren.

Med ordtype 2 mener vi ord og uttrykk som tilhører det allmenne ordforrådet, men som har høyere frekvens i spesielle fagtekster enn i generelle språkkurs. Eksempler fra samme fysikkbok: legeme, avta, gnidning, rivning. Dette er ord som vil være vanligere i fysikkbøker enn f eks historiebøker. Disse ordene vil være kjent for de fleste norske elevene, men fordi de ikke blir behandlet i undervisningen, vil de ofte være spesielt vanskelige for de fremmedspråklige elevene. Ofte vil fagtermene bli forklart ved hjelp av nettopp slike ord - i den fysikkboka vi hentet eksemplene fra ble friksjon forklart ved gnidning og rivning - og dette fører til at de fremmedspråklige elevene ikke kan nytte den hjelpen faglæreren gir når han/hun gjennomgår det nye stoffet.

Vi vil derfor kartlegge disse to ordtypene. For å skille ut fagtermene har vi søkt assistanse hos faglærerene. På alfabetiske ordlister har vi bedt dem krysse av de ordene de regner for fagord i en klasserømssituasjon.

Vi har tatt for oss et utvalg av fysikk-, geografi- og historiebøker fra 4. til 9. klasse. Årsaken til at vi har valgt akkurat disse fagene er at de presenterer stoffet på forskjellige måter. På den ene siden står fysikkfaget som i stor utstrekning presenterer fakta med klare årsaksrelasjoner. På den andre siden står historiefaget som er langt mer beskrivende i formen. Geografifaget blir en mellomting av disse idet naturgeografien ligger nærmere fysikken i form, mens kulturgeografien kan sammenliknes med historiefaget. Vi vil derfor undersøke i hvor stor utstrekning dette gir seg utslag i ordforråd og syntaks.

Vi har lagt vekt på å finne fram til bøker som er mye brukt, men som samtidig er forholdsvis nye, slik at vi kan regne med at de vil være aktuelle i en del år framover. På barnetrinnet finnes disse fagene i O-fagsbøker, men kapittelinndelingen følger svært ofte de tradisjonelle faggrensene, så det har vært forholdsvis enkelt å skille ut de enkelte fagene. På hvert klassetrinn har

vi valgt ut mellom tre og seks bøker fra forskjellige forlag innen hvert fag. På ungdomstrinnet er derimot den tradisjonelle faggrensen opprettholdt og vi har valgt ut to til tre hele bøker fra forskjellige forlag innen hvert fag. Vi har fått kjørt ut alfabetisk sorterte ordlister med frekvens. Programmet gir oss også antall løpende ord, antall ulike ord og T/T-ratio. Når alle tekstene er skrevet inn, vil vi få i alt 18 frekvensordlister.

Ordlisteprogrammet vil regne alle identiske tegnsekvenser avgrenset av skilletegn eller mellomrom for samme ord, og i frekvensordlistene er homografene følgelig ikke skilt ut.

Eks fra fysikk 9. klasse:
hell (1)
helle (1)
heller (19)
helt (44)

Vi har derfor gått gjennom alle listene og plukket ut alle ord som kunne være homografer (se nedenfor ang kriteriene for homografer). Deretter har vi kjørt ut konkordanser på disse ordene. Fra konkordanslistene kan vi skille ut de forskjellige homografkomponentene som finnes i teksten og hvilken frekvens de har.

Eks fra fysikk 9.klasse:
heller (adv) (8)
heller (verb, 'tømmer')(10)
heller (verb, 'skrår') (1)
heller (subst) (0)

Et ord som har valgfri ortografi (eks likne og ligne) vil bli ført opp som to forskjellige ord i frekvensordlisten siden ordlisteprogrammet regner to ord for ulike idet ett av tegnene varierer. Feilstavinger vil også bli ført opp som egne ord, og ikke minst vil bøyde former av ett og samme grunnord bli regnet for forskjellige ord.

Vi har derfor funnet det nødvendig å lemmatisere ordene, og dette arbeidet er blitt ganske omfattende. I følge Allén (1970) er et lemma "en grupp ordformer inom en ordklass vilka kan hänföras till antigen en och samma flexionsserie eller flera i tal och/eller skrift konvergerande serier vars divergenser visar rent fakultativ variation" (bind 2, side XVIII). Et lemmatiseringsarbeid der de forskjellige ordenes lemmatilhørighet bestemmes, vil følgelig holdes innenfor ordklassene.Vi har derimot funnet det hensiktsmessig å gå ut over ordklassene i vårt lemmatiseringsarbeid. Vårt utgangspunkt er jo å kartlegge frekvente, men sannsynligvis ukjente ord for elevene. Vi vil da måtte sette en nedre grense for hva vi regner for frekvente på de forskjellige klassetrinnene. Et ord med mange forskjellige avledningsformer vil da ikke komme med blant disse hvis hver av formene har lav frekvens. Det er med andre ord ikke de enkelte ordenes frekvens vi vil fange opp, men snare finne ut hvor frekvent en semantisk gruppe med lik rot er. Hvis elevene har arbeidet med ordlaging og segmentering, må man kunne anta at så snart ett av

ordene innen en slik semantisk gruppe er kjent, vil de andre ordene som dannes ved at man legger til eller trekker fra suffikser også være forståelige for elevene. Et eksempel: hvis en elev kjenner til ordet oppfinne vil han eller hun også forstå en oppfinner og en oppfinnelse ved hjelp av et visst kjenskap til ordlagingsregler. I vårt lemmatiseringsarbeid har vi derfor samlet et ord og alle avledningene som er dannet av dette ordet ved hjelp av suffikser og infikser. Vi har valgt å si at disse ordene tilhører samme rotlemma for å skille det fra lemmatisering av ord innen samme ordklasse, og prosessen kaller vi rotlemmatisering.

Det har vært nødvendig med visse restriksjoner og modifiseringer for å bestemme rotlemmatilhørigheten til ordene.

1. Når rotlemmatiseringen går ut over ordklassegrensen er det en forutsetning at uttalen eller skriftbildet til en av de bøyde formene av ordet i den ene ordklassen naturlig kjeder seg sammen med minst en av formene fra den andre ordklassen.
   Eks: verbet bryte og substantivet brudd vil tilhøre samme rotlemma fordi substantivformen kjeder seg naturlig sammen med partisippformen av verbet, brutt.
   Hvis derimot skriftbildet eller uttalen av de to ordene spriker for mye, vil vi la de tilhøre to forskjellige rotlemma.
   Eks: verbet gå og substantivet gange vil tilhøre to forskjellige rotlemma.
2. Avledningen må ikke forandre innholdet i noen større grad.
   Eks: dag og daglig vil tilhøre samme rotlemma, men ikke tid og tidlig.
3. En bøyningsrekke med suppletiv bøyning vil tilhøre forskjellige rotlemma.
   Eks: god vil tilhøre et rotlemma, mens bedre og best vil tilhøre et annet.
4. I forbindelse med ord som betegner nasjon eller nasjonalitet vil vi også la sammensatte ord tilhøre samme rotlemma som usammensatte.
   Eks: fransk, franskmann og Frankrike vil tilhøre samme rotlemma.

Som oppslagsord har vi valgt den formen som har færrest bokstaver av ordklassenes grunnform selv om denne formen ikke er belagt i tekstene. Hvis flere former har samme antall bokstaver, har vi valgt substantivformen. Det er imidlertid et unntak: hvis den korteste formen er relativt sjelden og ikke belagt i tekstene, blir den nest korteste formen oppslagsord.
Eks: tikke er oppslagsord i stedet for tikk og strekke i stedet for strekk.
Ved nasjon og nasjonalitetsbetegnelser er nasjonens navn oppslagsord, uansett lengde. Det samme gjelder andre proprier.
For å spare arbeid har vi bare ført opp oppslagsord i de tilfellene der minst to forskjellige former med samme rotlemmatilhørighet er belagt i tekstene. Hvis et ord står alene innenfor sitt rotlemma, vil det ikke få eget oppslagsord.

Kriteriene for bestemmelsen av homografer må sees i sammenheng med den rotlemmatiseringen vi utfører. Fordi vi skal fram til

et rotlemmanivå, er det ikke nødvendig å skille mellom homografkomponenter som tilhører samme rotlemma selv om de tilhører ulike ordklasser.
Eks: arbeider (verb) og arbeider (substantiv) tilhører samme rotlemma og har oppslagsordet arbeid. Vi regner derfor ikke arbeider som homograf. Derimot regner vi et ord som homograf hvis det har flere betydningavarianter, selv om komponentene vil tilhøre samme ordklasse og ha identiske bøyningsformer.
Eks: bestå har forskjellig betydning i uttrykkene 'bestå eksamen' og 'bestå av noe'.
Her er det nødvendig med spesifiseringer:

1. Hvis et ord har en abstrakt og en konkret betydning og den abstrakte er en ren overføring av den konkrete, regnes ikke ordet for homograf.
   Eks: framkalle vil ikke bli delt etter betydningene 'fotografisk prosess' og 'tankeprosess'. Derimot vil vi ta med eksempler både fra den konkrete og den abstrakte varianten hvis det blir aktuelt å lage øvingsoppgaver til tilsvarende ord.
2. Hvis ordet har flere betydningsnyanser som det er vanskelig å skille, regnes ikke ordet for homograf. I noen tilfeller skiller vi ut en betydning og lar resten stå sammen.
   Eks: side vil ha en komponent 'side i en bok', mens andre betydninger står sammen.

Fordi homografsepareringen utføres manuelt og fordi siktemålet er å lage øvingsmateriell, har vi ikke tatt ut konkordansen til homografer hvor det er lite sannsynlig at den ene komponenten vil forekomme i fagtekstene.
Eks: skål som verb i imperativ.
Andre homografer har vi regnet som uaktuelle i et fag, men aktuelle i de andre.
Eks: pil hvor betydningen 'treslag' regnes for uaktuell i fysikktekstene.

Etter at homografsepareringen er utført og rotlemmatilhørigheten bestemt, kjører vi et program som slår sammen alle ordene innenfor samme rotlemma. Vi får ut lister som oppgir samlet frekvens for rotlemmaet og som samtidig spesifiserer de formene som er belagt i teksten og frekvensen til disse.

Eks: bor (12): bore (3), borer (1), bores (1), boring (5)
boringene (2)

Når vi har fått ut alle de 18 rotlemmatiserte listene, vil vi ta stilling til hvor høy den relative frekvensen bør være for at rotlemmaene kan regnes for frekvente. Vi vil så lage øvingsoppgaver med de aktuelle rotlemmaene, et hefte for hvert fag og hver klasse.

I O-fagsbøkene er det en del uekte sammensatte verb, eks gi opp gi ut. Fordi vi ikke får ut frekvensen av disse formene på frekvensordlistene, vil vi ta ut konkordansen til en del adverb og preposisjoner som ofte blir brukt som verbalpartilker, slik at

vi også kan fange opp de løse sammensetningene.

Når vi er ferdige med behandlingen av ordforrådet, vil vi undersøke deler av syntaksen og oppbygningen av teksten utover setningsplanet. Vi ønsker spesielt å sammenlikne tekster på forskjellige klassetrinn som behandler like eller beslektede emner. Vi har foreløpig ikke tatt stilling til i hvilken grad vi skal bruke EDB til dette formålet.

Prosjektet LÆREBOKSPRÅK er finansiert av KUD (skoleforskningsmidler). Prosjektledere er Anne Hvenekilde og Anne Golden.
Addressen er : Nordisk institutt, pb 1013, Blindern,
Universitetet i Oslo

Programmene vi har kjørt er beskrevet i

Ivar Fonnes: Tekstanalyseprogrammer på DEC-10
stensil, UiO, 1980

S.Allén: Nusvensk frekvensordbok 1, Göteborg 1970

Anne Karin Ro
Engelsk institutt
Eirik Lien
Edb-tjenesten for humanistiske fag
Universitetet i Trondheim

WYCLIFFES BIBELTEKSTER PÅ RA2

Bakgrunn for prosjektet

Prosjektet som skal beskrives her, datamaskinell registrering av ortografiske variantforekomster i 12 håndskrifter inneholdende The General Prologue to the Wycliffite Bible, med henblikk på datering og lokalisering av håndskriftene, er et ledd i et større forskningsprosjekt som pågår ved Engelsk institutt i Trondheim under ledelse av professor Conrad Lindberg. Det utvidede prosjektet har to hovedsiktemål: (1) en nyutgivelse av The Middle English Bible, også kalt The Wycliffite Bible eller The Lollard Bible, en bibeloversettelse fra latin til senmiddelengelsk; (2) inngående studier av ca. 230 bibelhåndskrifter for om mulig å tilveiebringe mer presis informasjon om oversettelsesarbeidet som tilkjennes den engelske reformator John Wycliffe og hans læresvenner.

En fullstendig utgave av The Wycliffite Bible, basert på ca. 170 håndskrifter, ble utgitt i 1850 av J. Forshall og F. Madden, som hadde brukt 20 år på å undersøke håndskrifter lokalisert i Storbritannia og Irland. Forshall og Madden fant at håndskriftene kunne inndeles i to grupper; den ene gruppen oppviste en nesten ordrett oversettelse fra den latinske originalversjonen, den andre gruppen presenterte en oversettelse der idiom og syntaks samsvarte med engelsk språkbruk rundt 1400. (Bibeloversettelsen antas å være utført i perioden 1370-1395.) Forshall og Madden opererte derfor med to versjoner av bibeloversettelsen; en bokstavelig, tidlig versjon (Early Version) og en friere, senere versjon (Later Version), og deres 1850-utgave presenterer versjonene i parallelle kolonner på samme side.

Forshall og Maddens teorier om tilblivelsen av bibeloversettelsen ble ikke alvorlig utfordret før den svenske forskeren Sven L. Fristedt publiserte sin doktoravhandling om The Wycliffite Bible i 1953. Fristedt fant at håndskriftene på ingen måte lot seg gruppere så entydig i en tidlig og en senere versjon som Forshall og Madden hadde antydet. Det fantes håndskrifter som delvis samsvarte med den ene versjonen, delvis med den andre, og Fristedt mente at disse håndskriftene kunne representere en mellomliggende gruppe. Fristedts teori

gikk da ut på at det forelå ikke to separate oversettelser, snarere én original oversettelse (Early Version) med stadige revideringer, som i siste instans munnet ut i den senere versjon. Spørsmålet om antall versjoner ville imidlertid bare kunne svares på etter inngående analyser av hvert enkelt håndskrift og detaljerte sammenlikninger av alle håndskriftene. Fristedt påpekte også en del svakheter ved 1850-utgaven, som gjorde den uegnet for slike analyser; bl.a. hadde Forshall og Madden modernisert ortografi og tegnsetting slik at den redigerte teksten til en viss grad avvek fra basehåndskriftet.

Flere av Fristedts synspunkter ble delt av Conrad Lindberg, som siden 1952 har arbeidet med tekststudier av bibelhåndskriftene, primært med håndskrifter som Forshall og Madden klassifiserte som Early Version. I 1970 publiserte Lindberg i Studia Neo-Philologica en oversikt over kjente eksisterende Wycliffe Bible håndskrifter, og kunne vise til et antall på ca. 230, altså ca. 60 flere enn Forshall og Madden hadde undersøkt. Dermed forelå et større, ukjent materiale, som ved nærmere undersøkelse kanskje kunne gi svar på en del av spørsmålene i forbindelse med tilblivelsen av The Middle English Bible, i første rekke om hvor mange versjoner oversettelsen kan inndeles i; datering av de ulike håndskriftene, som muligens kan føre til en angivelse av den innbyrdes rekkefølgen av dem; og dialektal bestemmelse av håndskriftene, som muligens kan føre til geografisk lokalisering av dem og gi en pekepinn om opphavsmennene bak de forskjellige revideringene. I sitt arbeid med bibeltekstene har Lindberg kommet til den konklusjon at bare gjennom en detaljert sammenlikning av alle bibelhåndskriftene når det gjelder ortografi, tegnsetting, morfologi, vokabular og syntaks, i tillegg til en sammenlikning av håndskriftene med den latinske originalversjonen, kan man håpe å komme fram til antakelige løsninger på disse problemene.

The General Prologue: Innhold og betydning

Den delen av The Middle English Bible som er i fokus her, The General Prologue, skiller seg ut ved at dette ikke er en oversatt tekst, men en genuin middelengelsk tekst, på ca. 40 000 ord. Teksten er således spesielt interessant, fordi den både kan kaste lys over bibeloversettelsen og gi verdifull informasjon om senmiddelengelsk fra en språkhistorisk synsvinkel. The General Prologue er oppdelt i 15 kapitler og er i hovedsak en oppsummering av innholdet i de gammeltestamentlige bøker, ispedd argumenter for nødvendigheten av en bibel på morsmålet - et omstridt spørsmål på den tida -, og ender med en relativt detaljert beskrivelse av de prinsipper og metoder som ligger til grunn for det oversettelsesarbeidet som prologforfatteren har vært hovedansvarlig for.

The General Prologue finnes i 12 av de ca. 230 håndskriftene man kjenner til; de fleste av disse håndskriftene tilhører gruppen klassifisert av Forshall og Madden som Later Version,

men noen er knyttet til håndskrifter som er en blanding av Early Version og Later Version. Ett håndskrift inneholder utelukkende The General Prologue. Ikke alle håndskriftene presenterer den fullstendige prologteksten; her følger en oversikt over de håndskrifter som inneholder i det minste ett fullstendig kapittel:

A. MS W.H.Scheide 12, Princeton, N.J. (fullstendig)
B. MS CUL Mm.2.15, Cambridge (fullstendig)
C. MS CUL Kk.I.8, Cambridge (kap.1-9, 11-15)
D. MS Corpus Christi College 147, Cambridge (fullstendig)
E. MS University College 96, Oxford (fullstendig)
F. MS Bodley 277, Oxford (kap.1)
G. MS Lincoln College Latin 119, Oxford (kap.1-9,kap.11)
H. MS Royal I.C.VIII, British Library (kap.1)
I. MS Harley 1666, British Library (kap.1-15, men 15 fragmentarisk)
J. MS Trinity College Dublin 75, Dublin (fullstendig)

I tillegg finnes fragmenter av The General Prologue i MS Worcester Cathedral F.172 (K) og i MS British Library 10.046 (L).

Skissering av prosjektfaser

I studiet av håndskriftene har jeg funnet det formålstjenlig å dele prosjektet i to faser:

1.fase: Tekstredigering (avsluttet)

Siktemålet i denne fasen var å utarbeide en nyutgave av The General Prologue, basert på MS W.H.Scheide 12 (A), et håndskrift som ikke tidligere er kollatert av Forshall og Madden. Alle håndskriftene er blitt sammenliknet og varierende lesearter innen morfologi (hvor grammatisk signifikant), vokabular og syntaks notert i fotnoter. I tillegg forekommer språklige analyser av varierende lesearter. Det er også foretatt en detaljert sammenlikning med Forshall og Maddens utgave, slik at feilforekomster i 1850-utgaven er blitt rettet opp. En foreløpig glossar fra middelengelsk til moderne engelsk er blitt utarbeidet.

2.fase: Datamaskinell registrering av alle ortografiske variantforekomster i håndskriftene (under arbeid)

Det fantes ingen enhetlig nasjonal ortografisk standard i middelengelsk. Middelengelske håndskrifter reflekterer derfor regionale lingvistiske særdrag og kan således gi verdifull informasjon om dialektale forhold. Av denne grunn kunne bibeltekstenes ortografiske og grammatisk insignifikante morfologiske varianter vise seg å være av betydning når det gjaldt å bestemme den dialektale distribusjon av håndskriftene. Ifølge artikler publisert av A. McIntosh (Edinburgh University) og M.L.Samuels (Glasgow University) forefinnes utarbeidet

detaljerte senmiddelengelske dialektkart som muliggjør datering og geografisk lokalisering av senmiddelengelske håndskrifter, ved å bruke den såkalte 'fit-technique'. Dialektkartene er hittil upubliserte, men tilgjengelige for forskere ved henvendelse til The Middle English Dialect Project ved Edinburgh University. Den såkalte 'fit-technique' forutsatte imidlertid at man utarbeidet en 'Linguistic Profile', d.v.s. en, fortrinnsvis fullstendig, analyse av ca. 300 forskjellige grafsekvenser for hvert håndskrift, og det ble derfor besluttet å foreta en registrering og analysering av alle variantforekomstene i The General Prologue. Ettersom tekstmaterialet var såvidt omfattende, over 300 000 ord, ble en datamaskinell registrering foretrukket framfor en manuell bearbeiding av variantene.

Edb-opplegget

To problem måtte løses når vi først hadde valgt å bruke edb:

- hvordan skal dataene overføres til maskinleselig form

- hvilken metode skal brukes for å sammenlikne skrivevarianter av samme ord fordelt på de enkelte manuskriptene

Det er klart at registreringsarbeidet ville bli enormt om vi skulle skrive inn de 12 aktuelle manuskriptene fortløpende og hver for seg. Fordi ikke alle de 12 tekstene er fullstendige og fordi de fullstendige tekstene ikke nødvendigvis behøver å ha samme antall ord, ville vi etterhvert få problemer med å vite om vi er kommet like langt i alle tekstene. Dessuten ville mange ord som - tross alt - er like i de 12 tekstene bli skrevet 12 ganger i stedet for en gang. Her blir det et stort skrivearbeid (40 000 ord x 12 - manglende deler) - la oss si ca. 300 000 ord - og enhver effektivisering vil gi gunstige utslag.

Vi valgte i stedet å bruke an annen metode, å snu det hele "på hodet". Ved å gjøre det oppnår vi at vi kan bruke vår egen innsikt til å si hvilke ord som hører sammen og ikke overlate til maskinen å kontrollere om vi er kommet like langt i alle 12 tekstene til enhver tid.

Men vi har ennå ikke oppnådd noen rasjonaliseringsgevinst. Den får vi hvis vi nå ser nærmere på dette. Her er det nemlig lett å se hvilke ord som skrives likt og hvilke manuskripter de er knyttet til. La oss "nummerere" manuskriptene A,B,...,L. Vi kan da knytte denne koden til de aktuelle skrivevariantene og slippe å skrive flere ord enn nødvendig. Vi har ikke regnet noe på hvor mye vi har spart på dette, men et raskt overslag viser i hvert fall at vi har redusert skrivearbeidet med bortimot 2/3. Det gir jo en betydelig arbeidsbesparelse når innmatingen blir på omkring 100 000 ord i stedet for ca. 300 000!

I tillegg er det knyttet til tagger som bærer av grammatisk informasjon. De er alltid markert i dataene med $. Disse taggene gjør det også mulig å skille mellom likt skrevne ord som har forskjellig betydning og funksjon (homografer).

For å få oversikt over de forskjellige skrivevariantene av samme ord har vi laget et programsystem som består av disse tre programmene:

delogsort
lager to ordlister
1. Hver eneste skriveform får knyttet til seg et løpenr som forteller i hvilken rekkefølge i teksten det kommer. Alle skrivevariantene av samme ord på samme sted i de 12 tekstene får samme nr. I tillegg blir versjonskoden knyttet til.
2. En alfabetisk liste over alle ord med tilknyttet løpenr.

I tillegg lager programmet en utlisting av alle ordene den fant i den aktuelle teksten, en inventarisering. Den er utgangspunkt for å lete etter ord.

input
mater inn de to listene i databasen,
- den ene slik at det er mulig å starte med et bestemt løpenr. og finne alle skrivevarianter knyttet til det, og med manuskriptkoder koplet til de enkelte skrivevariantene
- den andre slik at det er mulig å starte med en bestemt skrivevariant og se hvilke løpenr. som er knyttet til den.

For å legge opp databasen bruker vi databasesystemet RA2, som er utviklet ved Regnesentret ved Universitetet i Trondheim.

letord
får som inndata de "kanoniske" formene som vi vil ha som oppslag. En kanonisk form er den skrivevarianten av et ord som vi ønsker å oppfatte som felles form. Maskinelt har det ingen betydning hvilken form vi velger, men rent praktisk vil vi vanligvis velge den formen som ligger nærmest opptil skrivenormen idag.

Programmet tar det første ordet i denne lista og slår opp i den alfabetiske delen av databasen. Der finner den alle løpenr. knyttet til den skriveformen, og legger dem på en egen arbeidsliste. Så går programmet til det første løpenummeret på den lista og finner der de aktuelle skrivevariantene som er knyttet til. Disse variantene, samt manuskriptkode legger den på en annen arbeidsliste. Den starter så på toppen av denne lista og går inn i den alfabetiske delen igjen for å se om det er knyttet andre løpenr. til den varianten. Hvis det er det, kompletteres arbeidslista med løpenr. Slik gjør den med alle variantene på denne ordlista.

Hvis det er kommet til nye løpenr., går programmet til løpenummerdelen av databasen, og ser etter om det er nye skrive-

varianter der som ikke er registrert.

Slik veksler den mellom oppslag i de to listene, kontrollerer og eventuelt supplerer arbeidslistene til alle mulighetene er sjekket. Da har den grunnlag for å skrive ut en oversikt som kan se slik ut:

x x x x ALL DET

	A	B	C	D	E	F	G	H	I	J	SUM
ALLE	6	6	6	7			5		7	14	51
ALL					16					2	18
AL	11	11	11	10	1		12		10	1	67
											136

Her er alle skrivevariantene knyttet til versjonskode og summert opp. Det sier seg selv at med store datamengder blir dette tidkrevende, selv med raske og effektive dataanlegg. Mange av oppslagene er dessuten resultatløse, men det vet vi ikke før oppslaget er gjort.

Resultatet må sies å være godt, programmet letord er i stand til å finne de enkelte skrivevariantene av samme ord. Tidsforbruket er akseptabelt, på et UNIVAC 1100/62-anlegg bruker programmet i underkant av 10 minutter cpu-tid til å gå gjennom et "gjennomsnittskapittel" og sette opp tabellene. Et slikt kapittel har ca. 3 500 ord i løpende tekst, ca. 2 500 graford (med tagger inkludert) og ca. 750 kanoniske former.

Hanne Ruus
Institut for nordisk filologi
Københavns Universitet

ORDBØGER FOR FREMTIDEN

I midten af det tyvende århundrede havde et ordbogsarbejde typisk følgende faser:

1. Opstilling af retningslinjer for udvælgelse af materiale.
2. Indsamling af materiale.
3. Fastlæggelse af redaktionsprincipper. Udarbejdelse af redaktionsregler.
4. Redaktion af ordbogsartiklerne.
5. Trykning og distribution af ordbogen.

Alle dele af processen blev udført manuelt. En stor del af fase 2, 4 og 5 bestod af af- eller renskrivningsarbejde og korrekturlæsning.

I begyndelsen af det enogtyvende århundrede vil automatiseringen have ført til, at et ordbogsarbejde typisk har følgende faser:

1. Detaljspecifikation af ordbogens indhold og opstilling.
2. Materiale til ordbogen fremtages automatisk fra andre maskinlæsbare ordbøger og fra ord- og tekstbanker.
3. (Semi-)automatisk redaktion af ordbogsartikler.
4. Trykning eller distribution ad andre databærende medier som informationsnetværker og teletekst.

I denne version af arbejdsgangen vil fastlæggelse af redaktionsprincipper og redaktionsregler indgå i arbejdet med detaljspecifikationen af ordbogen.

Fase 2, fremskaffelse af materiale, vil kunne foregå automatisk, idet udvælgelseskriterierne vil fremgå af detaljspecifikationen.

Fase 3 vil i det ideelle tilfælde bestå i en kontrollerende gennemlæsning af ordbogsartiklerne, som er fremstillet automatisk ved hjælp af generelle programmer, hvis specielle anvendelse er udledt af detaljspecifikationen.

Fase 4 foregår helt automatisk. Af trykningsmetoder kan man tænke på laserprintning, mikrofiche eller fotosætning. Ved distribution ad andre databærende medier vil det primært være et spørgsmål om at vejlede ordbogsbrugerne i, hvilke ordrer de skal skrive for at finde de nye oplysninger i eksisterende databaser. Sådanne vejledninger vil også kunne fremstilles automatisk og lægges ind som en del af systemerne.

Her i firserne er vi nået et godt stykke vej mod automatiseret fremstilling af ordbøger:

Maskinellet til datamatiseret ordbogsarbejde findes stort set i dag. Lagring af store datamængder er f.eks. ikke længere noget problem. Det mindst menneskevenlige på maskinelsiden er inddateringsmedierne, hvor f.eks. diakritiske tegn som oftest kræver specialbehandling.

På programmelsiden bør det ikke vare længe, før man kan købe en generel programpakke, der er egnet til udarbejdelse af ordbøger. Ingredienserne til en sådan pakke findes allerede i tekstbehandlingssystemer, i dokumentations- og informationssystemer og i forskelligt databaseprogrammel.

Man kan pege på flere andre faktorer, der befordrer automatiseringen af den almensproglige leksikografi:

1. Automatiseringen af trykkeprocessen.
Så godt som alle bøger sættes i dag automatisk og deres tekst gøres derfor maskinlæsbar i fremstillingsprocessen. Denne trykketekniske udvikling kan man få gavn af også i næsten afsluttet ordbogsarbejde f.eks. ved at få gjort teksten maskinlæsbar, når kladden til det endelige manuskript foreligger, som det sker ved Nye Ord i Dansk (Riber Petersen 1981).

2. Opbygningen af større samlinger af maskinlæsbare tekster. I Skandinavien kan man pege på Logoteket ved Språkdata i Göteborg, Norsk Tekstarkiv, DANwORDs samling af tekstprøver i København. Sådanne tekstsamlinger kan danne basis for ordbogsprojekter, som det f. eks. er planlagt i Göteborg med projektet Leksikalisk databas (Allén, Gavare, Ralph 1981).

3. Maskinlæsbare ordbøger.
Alle ordbøger, der udgives nu, eksisterer i maskinlæsbar form jf.1. Hvis man kan komme til rette med eventuelle copyrightproblemer, er der her et rigt materiale for andre ordbøger. Desuden findes der ordbøger, der er født maskinlæsbare. Det gælder alle ordbøger, der anvendes inden for projekter om automatisk analyse af naturlige sprog f.eks. kunstig

intelligens og maskinoversættelse. Sådanne ordbøger kan indeholde anselige datamængder f.eks. indeholder den tyske analyseordbog til SUSY ved SFB 100 i Saarbrücken omkring 300 000 leksikalske indgange (Maas 1980).

I den almensproglige leksikografi er frekvensordbøgerne nærmest automatiseringen. I udarbejdelsen af Nusvensk Frekvensordbok er der anvendt edb i alle faser af arbejdet, i DANw ORDsprojektet er tekstprøveindsamling og indkodning manuel, mens fremstilling af frekvenslister foregår automatisk.

Nærmest den fuldstændig automatiserede ordbogsfremstilling er man for tiden i fagsprogsleksikografien, hvor f.eks. TEAM, Siemens' fagsproglige database, kun behøver seks uger til at fremstille en ordbog. Til en flersproget ordbog over edb-termer brugte de således fem timers maskintid til at udtage materiale, en halv times maskintid til at redigere det og halvanden times maskintid til at fotosætte ordbogen (Sager and McNaught 1980).

Automatiseringen af ordbogsfremstillingen vil ændre det leksikografiske arbejde. F.eks. vil effektiv søgning kræve, at man begrænser antallet af synonyme betegnelser og at forskellige typer af oplysninger holdes klart adskilt.

Den mest indgribende ændring i ordbogsarbejdet vil nok være, at den detaljerede planlægning flyttes til den indledende fase. Denne planlægning kan blive meget omfattende, især når det drejer sig om leksikalske data, som tænkes lagret maskinelt til direkte benyttelse for oversættere, terminologer og undervisere. Det ser man af forarbejderne til DANTERM (Frandsen og Nistrup Madsen 1980, Nistrup Madsen 1981) og til British Linguistic Data Bank (McNaught 1981).

Hvis man ændrer sine planlægningsvaner og datamatiserer sit ordbogsarbejde, får man til gengæld en række muligheder, som savnes ved manuel ordbogsfremstilling jf. en række bidrag i denne publikation.

Datamater er uovertrufne til at sortere, søge og ændre - hundrede procent konsekvent. Man har altså mulighed for at undersøge klassificeringer og typer af oplysningers distribution i ordbogen på alle tænkelige måder med deraf følgende gevinst i form af konsekvens i behandlingen af samme type ord forskellige steder i alfabetet. Hvis man har planlagt sin ordbogsartikel passende struktureret, giver ændringsfaciliteterne mulighed for at ændre samtlige forekomster af en bestemt værdi på en bestemt plads i artiklerne lige til dagen før, man sender ordbogen til trykning.

Der bliver også mulighed for at foretage eksperimenter, som på længere sigt vil kunne ændre alle forestillinger om, hvordan ordbøger er indrettet og ser ud.

Man kan afprøve ordning af ordene efter deres betydningsmæssige sammenhæng f.eks. i et flerdimensionalt netværk som nævnt af Ralph (1979). Behov for forskning i denne retning kan udledes af Sager and McNaught (1980). Deres forespørgsler til potentielle brugere af en British Linguistic Data Bank viser, at oversættere gerne vil have overbegreb, synonymer og antonymer på kildesproget foruden målsprogsækvivalenten, når de slår et ord op.

I ordbøger, der som de fagsproglige databaser er tilgængelige via elektroniske medier, kan man anvende den ordning af ordene, der er mest meningsfuld for brugerne f.eks. en sammenkædning efter betydningsfællesskab, idet det overlades til programmerne at finde rundt i datamængden efter de kriterier, som kan udledes af brugerens spørgsmål.

Hvis teletekst ad åre bliver lige så udbredt, som telefonen er det i dag, er det nærliggende at gøre også almensproglige ordbøger tilgængelige via elektroniske medier. I så fald kan man også i disse ordbøger udnytte mulighederne for at koble ordene sammen efter betydningsfællesskaber i ordstoffet frem for efter formelle egenskaber som samme begyndelsesbogstav. Her vil man få hårdt brug for resultater af forsøg med at kæde almensprogets ord sammen i thesauruslignende strukturer.

Henvisninger.

Allén, Sture: Nusvensk Frekvensordbok, baserad på tidningstekst, 1-4, 1970-1980.

Allén, Sture, Gavare, Rolf and Ralph Bo 1981: Språkdata Research Report 1980.
Compiling nr. 10, feb. 1981.

Frandsen, Lene and Nistrup Madsen, Bodil 1980: The Setting up and Operation Of a Danish Terminological Data Bank (The DANTERM Project), i Human Translation - Machine Translation ed. by Suzanne Hanon and Viggo Hjørnager Pedersen = NOK 39, Romansk Institut, Odense Universitet, s.121-131.

Maas, Heinz-Dieter 1980: Zur Entwicklung des Übersetzungssystems SUSY und seiner einzelnen Komponenten, i Maschinelle Übersetzung, Lexikographie und Analyse, Akten des 2. Internationalen Kolloquiums Saarbrücken, November 1979, herausgegeben von Hans Eggers = Linguistische Arbeiten, Neue Folge, Heft 3,1 Universität des Saarlandes, Sonderforschungsbereich 100, s.7-16.

Maegaard, Bente og Ruus, Hanne 1978: DANwORD, Hyppighedsundersøgelser i moderne dansk: Baggrund og materiale, i Danske Studier 1978, s.42-70.

Maegaard, Bente og Ruus, Hanne: Hyppige Ord i Danske Børnebøger, Gyldendal, nov. 1981

Maegaard, Bente og Ruus, Hanne: Hyppige Ord i Danske Romaner, Gyldendal, nov. 1981.

McNaught, John 1981: Terminological Data Banks: a model for a British Linguistic Data Bank (LDB), i Aslib Proceedings 33 (7/8), July/August, s.297-308.

Nistrup Madsen, Bodil 1981: Nye veje inden for fagsproglig leksikografi, i SPRINT 2, Sproginstitutternes tidsskrift Handelshøjskolen i København, s.18-23.

Ralph, Bo 1979: Leksikologi som datalingvistik, i Nordiske Datalingvistikdage i København 9.-10. oktober 1979, Foredrag udgivet af Bente Maegaard, s.161-170.

Riber Petersen, Pia 1981: Ordbøger og edb, i SAML 8, Udgivet af Københavns Universitets Institut for anvendt og matematisk lingvistik, s.179-191.

Sager, J.C. and McNaught, J. 1980: Feasibility Study of the Establishment of a Terminological Databank in the U.K., British Library R. & D. Report Nr. 5642.
Selective Survey of Terminological Databanks in Western Europe, British Library R. & D. Report Nr. 5643.
Model Specification of a Linguistic Databank for the U.K., British Library R. & D. Report Nr. 5644.

Jonas Löfström
Språkdata
Göteborgs universitet

DOLDA ORDBILDNINGSMÖNSTER. NÅGRA PROBLEM INOM DATAMASKINELL LEXIKOLOGI

Med dolda ordbildningsmönster avses de regelbundenheter i vårt ordförråd som inte framträder i ordböcker så som dessa brukar vara uppbyggda. Det lingvistiska problem vi här har att lösa med datamaskinella metoder är hur ordbildningsstrukturerna i språket skall kunna registreras och presenteras i bearbetningar av lexikaliskt material. En utgångspunkt är att det vid lexikologiskt arbete ofta kommer fram information som inte är direkt avsedd för publicering respektive som publiceras men i ostrukturerad form. Med hjälp av datamaskinen är det möjligt att strukturera och lagra informationen så att den senare blir lätt åtkomlig. Man bygger upp ett välordnat lager av data. Detta kan ske på olika sätt rent tekniskt, men det går vi inte in närmare på här.

När vi talar om dolda ordbildningsmönster menar vi alltså att den alfabetiska förteckningen endast i begränsad utsträckning direkt synliggör ordbildningsmekanismen. En relativt vanlig och samtidigt enkel åtgärd för att råda bot på detta är att göra en finalalfabetisk sortering av det lemmatiserade ordmaterialet. Många sammansättningstyper och avledningstyper framträder då. Men redan det faktum att prefigerade ord kan fungera som andra led i en sammansättning och att suffixavledningar kan följas av ett led i en sammansättning gör att den information vi får i en initial- respektive finalalfabetisk listning inte är fullständig. På motsvarande sätt uppträder vanliga grundmorfer på olika håll i orden. Det kan vara med praktiskt taget oförändrad betydelse eller med större eller mindre betydelseförskjutningar. För en komplett lexikonbeskrivning av ett språk behöver vi även denna information om grundmorfernas distribution. Detta har inte bara ett stort teoretiskt intresse. Den praktiska betydelsen av sådan kunskap torde vara uppenbar.

Den information som ligger dold eller osystematiserad i de flesta ordböcker kan man alltså lyfta fram. Det gäller då att veta vad som är relevanta upplysningar och att sedan bearbeta och sortera dem på ett konsekvent sätt. Om man lyckas lagra sin information väl kan man sedan få svar på sina frågor och till och med på oställda frågor. Det är det kanske mest lockande perspektivet – att genom ett konsekvent analysförfarande avslöja oväntade och okända egenskaper i ordbildningssys-

temet. Man står då inför uppgiften att modifiera sin modell.

Om man utarbetar en ordbok med annat huvudsyfte än att beskriva ordbildningsstrukturer ställs man inför valet att göra den morfologiska beskrivningen samtidigt med det övriga eller att göra en separat morfologisk omgång. Mycket talar för att en preliminär genomgång av materialet med avseende på segmentering (av orden), separering (av homonyma enheter) och klassificering av enheterna i morfem (d v s fastläggande av tillåten allomorfi) bör föregå det övriga arbetet.[1] Vid arbete med datamaskinella metoder finns ju då all möjlighet att med utgångspunkt i regler och/eller en förberedande analys av ett begränsat material låta datorn presentera förslag som lexikologen har att acceptera, förkasta eller justera. Resultatet torde lättare bli enhetligt och arbetet gå snabbare med en sådan uppläggning. Det innebär bl a att man har en ram att arbeta inom vilket gör att problem och egenheter lättare uppmärksammas. Annars blir det av förklarliga skäl lätt så att man bortser från det som man inte just för tillfället skall beskriva. Detta vill säga att frågor om kommutation, d v s vilken segmentering av orden man kan tillåta sig, morfotax (vilka morfemtyperna och deras funktion är) och allomorfi (vilken växling som ska accepteras) aktualiseras. Av det sagda framgår klart att ett fast format för den information man vill kunna ge är en stor fördel i arbetet. Till det som redan sagts kommer frågan om polysemi. Både vad det gäller grundorden och affixen är betydelsen viktig för att förstå ordbildningsmönstren.

Morfotax

Det är av grundläggande betydelse vilken morfotaktisk information man bestämmer sig för att ta hänsyn till och redovisa. I viss mån beror valet på vilket språk det gäller men även syftet kan spela in. För de nordiska språken är väl en rimlig indelning: prefix, grundmorf(er), fog, avledningssuffix samt böjningsändelse. Denna senare har en marginell betydelse för ordbildningen sett från en lexikalisk synpunkt. Till det nämnda kommer också infix som en enhet som kan visa sig vara en alternativ lösning när man stöter på problem i analysen av ordförrådet.

Kommutation

Fastställandet av ordledstyper ger oss en fastare ram för kommutationen. Det får i det enskilda fallet avgöras på vilka grunder kommutationen kan ske. I ett väl beskrivet språk har man ju ett försprång i det att en lång rad suffix och prefix är kända och väldefinierade. Med utgångspunkt i en förteckning över dessa förenklas kommutationsfasen. Det är dock

uppenbart att varje någorlunda stort material kommer att bjuda på svårigheter och att lösningen där till stor del får bero på syftet med undersökningen. Förutom hänsyn till språklig referensram (språkkänsla, kompetens), morfotaktiska mönster och semantisk affinitet har man att beakta variationen. Att *-are* och *-ing* är avledningssuffix i svenskan råder ingen tvekan om, men hur är det med *-ats*? Om man vid sidan av *segla* och *seglare* har *seglats* så torde man få notera *-ats* som ett substantivavledande suffix, även om det knappast är produktivt. Svårare är de fall där det som föregår ett eventuellt suffix inte finns i något annat ord, t ex *giljotin* och *magasin*. Det är först om man på andra grunder kunnat etablera *-in* som ett acceptabelt suffix som kommutationen här kan ske och ge restmorfemet *giljot-* respektive *magas-*.

Allomorfi

Det är inte ett helt trivialt problem att avgöra vilken allomorfi man skall räkna med. Skall man acceptera endast etymologiskt riktig växling eller även synkront sett rimlig variation? Frågan om semantikens roll vid bedömningen är avgörande. Syftet med den aktuella undersökningen får avgöra.

Den inomlemmatiska variationen som ju oftast finns förtecknad är en god utgångspunkt eftersom den ibland även fungerar mellan lemman. En del ordbildning har ju historiskt sett skett med utgångspunkt just i någon av lemmats böjningsformer. I andra fall finns det liknande regulariteter mellan olika typer av ord i ordförrådet. Det ger oss då en sammanhållen typ av växling som det kan råda stor enighet om och som oftast speglar en tidigare produktivitet. I många fall har dessa regulariteter sannolikt också en psykologisk relevans. Även andra typer av regelbunden växling kan noteras. Det gäller t ex fonologisk (och grafonomisk) variation inom en morfotaktisk ram.

Polysemi

När det gäller betydelsen är den morfologiska nivån svårbemästrad. En viss spännvidd i betydelsen inom ett morfem är nödvändig för att instanser från olika ord överhuvudtaget skall kunna föras samman i ett morfem. Men en gräns för hur abstrakt betydelsen får vara måste ändå sättas. Är t ex *sätt* i *sätta*, *bosättning*, *färgsättning*, *ifrågasätta*, *fortsätta*, *sättmaskin*, *besättning*, *undsättning* och *översättning* samma morfem? Det skall i detta sammanhang framhållas att olika delar av ordförrådet naturligen inte ligger på samma abstraktionsnivå. Speciellt det klassiska och romanska språkgodset bjuder på tydliga ordbildningsmönster men med grundmorfer vars betydelse för en nordisk språkkänsla är mycket vag.

Prefix och suffix företer en liknande skala av betydelser. Men redan en indelning av dem i typer (person- eller verktygsbeteckning, abstrakt substantiv o s v) ger en nyansering av betydelsestrukturen.

Ordförbindelser

Till ordbildning i strikt mening hör kanske inte förbindelser av ord, men viss fraseologi bör nog noteras för att fullständiga bilden av ordförrådets struktur. Att *hand* återfinns i *handtag, handgemäng, hantverk* (med allomorfen *hant*), *egenhändig* (*händ*) och *valhänt* (*hänt*) är klart, men det ligger nära till hands att redovisa även förekomsten i fraser som *i första hand, i sista hand, ta hand om* o s v. Jämför *omhändertagande* och *förstahandsinformation*.

Materialtyper

För lexikologiska syften torde man nästan helt kunna nöja sig med lemmatiserade ord som utgångsmaterial. En viss kontroll mot löpande text kan dock vara påkallad. En del ord förekommer endast i pluralis (t ex *vara i faggorna*) eller i bestämd form (*vara i görningen*) och de skall naturligtvis redovisas på något sätt. En typ av sammansättningar, *1900-talet*, förekommer knappast i obestämd form singularis, men man skulle ändå vilja registrera ordbildningstypen som ju har den egenheten att förledet är utbytbart men inte är av den typ som förtecknas i ordböcker. Detsamma gäller abbreviationer.

För att få fram språkets böjningsmorfologi behövs för de syften lexikologiskt arbete avser i ett väl beskrivet språk inte någon kommutation av det fullständiga olemmatiserade materialet. För vissa ord med mera fraseologisk användning samt för kontroll av polysemi är däremot den löpande texten värdefull som komplement. Dessutom innehåller böjningsformerna en hel del av den allomorfi som det är praktiskt att redovisa i ordbildningssammanhang.

Lagring

Lagringen av information torde kunna ske på olika sätt och får rättas efter syftet. En uppdelning av orden i morfem med möjlighet till märkning av dem är en första förutsättning. Vidare bör märkningen vara åtkomlig för sökning och sortering samt kunna avse olika nivåer eller typer av analys. Eventuellt kan information om alternativa morfemgränser vara aktuell. Det är då att notera att morfemens status och inbördes koppling noggrant bör kontrolleras så att resultatet av

maskinella bearbetningar av ett stort material inte blir felvisande. Ett kommutationstest kan ju inte utföras på båda analyserna av ett ord samtidigt. En annan möjlighet vid kommutationen är att laborera med olika starka morfemgränser och redovisa dem.

Några exempel

I det följande tar jag upp några exempel på de typer av problem som man ställs inför vid en morfologisk analys. De illustrerar också i viss mån de ovan nämnda aspekterna av morfologin.

Sammansättningar

Sammansättningarna kan vara två- eller flerledade. De tvåledade är hierarkiskt uppbyggda: *riksbanks-fond, arbetsmarknads-styrelse, krigs-skadestånd, turisttrafik-förening*. Bindestrecken här markerar en primär gräns. Men i leden *riksbanks, arbetsmarknads, skadestånd* och *turisttrafik* finns ytterligare, sekundära gränser. Dessa olika gränser kan markeras på önskat sätt för att möjliggöra en kartläggning av bruket av fogar (i det här fallet *-s-*) i olika sammansättningar. Kopulativa (flerledade) sammansättningar av typ *dansk-norsk-svensk* har inte den ovan beskrivna hierarkiska uppbyggnaden. De aktualiserar också ett ordfogningssätt som använder icke-alfabetiska tecken. Det kan vara värt att uppmärksamma eftersom en vacklan i bruket av böjningsändelser (jfr nedan) tycks göra sig gällande.

En märkning av ordleden med avseende på ordklass kan ge en kompletterande bild av ordbildningen. I många fall kan det vara svårt att från en synkron synpunkt avgöra ett sammansättningsleds ordklasstillhörighet. Är *spelhåla* sammansatt av *spel* (nomen) eller *spela* (verb) och *låda*? Möjligen kan etymologisk information vara intressant här.

Avledda sammansättningar utgör ett annat problem. Det gäller att återge den annorlunda strukturen och eventuellt antyda den bakomliggande syntagmen. *Blåögd* ('med blå ögon') står ensamt utan något **ögd* att falla tillbaka på. Morfemet *öga* saknar verbavledning men har här i sammansättningen en participliknande form och adjektivisk funktion. En märkning med ordklasstillhörighet på *ög* och angivande av *-d*:s funktion ger en sammanställning som jämförd med andra liknande kan ge perspektiv på ordbildningsmönstren.

Det normala fallet är att förleden i en sammansättning är oböjd. Som vi nyss antydde kan den också utgöras av stammen så att ordklassen inte framgår. Dessutom finns fogeelement: *s*

eller en vokal. Böjd form i egentlig mening förekommer också i några fall och är därför av intresse för förståelsen av (produktiv) ordbildning. Substantiv kan som förled stå i pluralis, *ländergrupp*, vilket kanske bäst noteras genom en markering på pluralmorfemet. Att nöja sig med att urskilja hela ordet *länder* eller bara *land* kan ställa till problem vid en mer generell behandling av allomorfi. Bestämd form i sammansättningar, *toppenbra*, *bottennapp*, tycks också kunna härledas från en syntagm. Vissa förled kan dock bli så produktiva att de snarare får betraktas som en ny enhet skild från simplex, t ex *toppen-* vid sidan om *topp*. Att detta får konsekvenser för analysen är uppenbart. Därför är en komplett kartläggning av sådana fall av bestämd form i förleden en hjälp när man kommuterar fram grundmorferna.

I vissa fall förekommer inre böjning av adjektiv, där båda leden är samordnade, *brittiskt-franskt*. Analysen bör lagras så att bruket av böjning inne i sammanställningen kan relateras till dels slutledets böjning, dels bruket av ickealfabetisk fog. Även komparationsformer kan ingå i ordbildningen: *mindretal* (jfr *mindervärdeskomplex* med en allomorf *minder*), och *förstfödd*. Att registrera dessa morfem och deras formella och semantiska egenskaper kan skapa förutsättningar för en detaljerad ordbildningslära.

De verb som har både s k lös och fast sammansättning utgör ett område som kan sprida ljus över intressanta ordbildningsprinciper. Med en genomtänkt analys och klassificering kunde man sammanställa olika fall av den här typen: *avbryta*, *bryta av*, *avbrott*, *avbruten* och eventuellt fler. Som synes kommer här även substantiv(avledning) in i bilden. Detta är ett bra exempel på behovet av ett effektivt grepp på den allomorfiska växlingen. Morfemet *bryt* har här allomorferna *bryt*, *brut* och *brot(t)*. Kan dessa presenteras samlat är mycket vunnet i åskådlighet. De semantiska egenskaper som finns här gör det fördelaktigt att även beakta lexem med mer än ett ingående ord. De sammansättningar som innehåller ickealfabetiska tecken, numeraler, abbreviationer, egennamn o s v saknas ofta i lexikon. Om man har tillgång till löpande text kan man få en bild av en mycket utbredd ordbildningstyp. Ex.: *1900-tal*, *50-kort*, *ABF-cirkel*, *Uddevallabo*. Det är inte helt orimligt att tänka sig en specialbehandling av dessa så att de åtminstone redovisas på efterleden.

Det finns en del sammansättningsled som fått en så allmän användning att de närmar sig prefix- eller suffixstatus. *Ur* i *urdum*, *uråldrig* och *vänlig* i *barnvänlig*, *riktig* i *sittriktig*. Skall dessa fall föras upp som egna enheter eller inte?

Avledningar

I sammansättningar har man grovt taget att göra med hela ord som kombineras med varandra. I vissa fall faller ett ordbildningselement i slutet av förleden: *yxa* och *skaft* ger *yxskaft*. Låt oss kalla den del av ordet som återstår när ordbildningselementet avlägsnats för grundmorf.

Avledning blir då ordbildning med tillägg av prefix eller suffix till grundmorfen. När det gäller suffix är det vanligt med grupper av ord som hör ihop betydelsemässigt och skiljs åt genom olika suffix eftersom de har olika syntaktisk funktion: *vals, valsa, valsare, valsning*. För vidare analys och bearbetning är alltså uppgift om vilken ordklass suffixet bildar oundgänglig. Vissa suffix kan bilda olika ordklasser, så t ex *-a*: *kratta* (verb), *kratta* (nomen). Detta är ytterligare ett skäl att märka dem för att kunna göra relevanta bearbetningar.

Prefixen ger inte av egen kraft ordet (ny) ordklass. I viss mån skulle de därför kunna sägas ha en starkare semantisk innebörd. I fall som det negerande *o-* i *oordning* framträder detta drag. Det är emellertid vanligt att prefixet endast har en lätt nyanserande funktion, *fästa - befästa*, ofta på gränsen till rent formell funktion. Att så är fallet har bl a att göra med att många prefix kommit in via lån av hela ordet. Ett speciellt problem med avledningarna är att vi har ett germanskt och ett romanskt mönster. Som vi skall se ställer detta till problem bl a vid kommutationen.

Prefixen har ofta en mycket vag betydelse men kan ändå kommuteras fram då de förekommer i många olika ord: *su-* i *suspekt* och *suterrängvåning*, *pro-* i *procent* och *producent*, *be-* i *beredskap* och *betyg*, *ge-* i *angelägen* och *gestalt*.

En grundmorf som *duk* i *introduktion, obduktion, produktion, konduktör* och *reduktion* framstår tydligt som en byggsten i vårt ordförråd. Simplex saknas (i den här aktuella betydelsen) och det är först vid analys som vi identifierar morfen i det enskilda ordet. Däremot accepterar språkkänslan lättare *produk* i *produktion* och *produkt* som meningsfulla ordled. Om man tar med även *producera, producent* så framstår samhörigheten klart. Vi har då också en allomorfisk växling *produc/produk*. Det finns anledning att acceptera olika nivåer i ordbildningen. De på olika kommutationsgrunder upprättade gränserna i orden kan med fördel hållas isär.[2] Om man utgår ifrån *duc/duk* kan man få följande lista:

in duc era	*konduk tör*
in duk tion	*obduc era*
intro duc era	*obduk tion*
intro duk tion	*produc era*
via duk t	*produk tion*
akve duk t	*produc ent*
	produk t

I den vänstra spalten identifieras en betydelse 'leda fram' hos *duc/duk* medan man i exemplen i högra spalten måst tillgripa en konkatenering av prefix och grundmorf för att få en psykologisktgodtagbar analys. På sitt sätt blir detta en kombination av synkrona och diakrona (etymologiska) aspekter.

Suffixen bildar flera räckor som *lära, lärare, (in)lärning, lärarinna, lärling*, där suffixets modifiering av grundmorfen är ganska klar och förutsebar. Många analogibildningar förekommer. Lexikaliseringar förekommer givetvis. Även det klassiska eller romanska materialet innehåller den här typen av mönster. Ta t ex *informera, information, informator, informatör, informativ*. Ett påtagligt problem om man behandlar olika delar av ordbildningen separat utgör de inkonsekvenser som blir följden. Det är inte ovanligt att man i en svensk ordbildningslära hittar exempel som *arrang-era, arrange-mang; medic-in, medika-ment; arrendat-or* (trots *arrendera*); *reforma-tor* (trots *reformera*). Beskrivningen av ordförrådet och ordbildningsprinciperna kan inte bli annat än fragmentarisk om man inte beaktar denna typ av regelbundenhet.

En del avledningar innehåller svårplacerade infix: *beskaffenhet* kopplas naturligen ihop med *beskaffad* och avledningssuffixet *-het* står för abstrakt substantiv. Men *-en-* låter sig inte så lätt föras till någondera av de ovan nämnda ordleden. Vi får helt enkelt räkna med ett extra infix som inte tycks bära någon egen betydelse eller ens funktion. Jämför *rosenknopp*. Även det inlånade klassiska eller romanska ordförrådet har – åtminstone sett ur svenskans perspektiv – sådana infix: jämför *human, humanitär* med *revolution, revolutionär*. Mellan grundordet *human* och den etablerade adjektivändelsen *-är* finns det ett "överflödigt" infix *-it*. En rimlig lösning vore kanske att behandla *-itär* som en variant av *-är*. Alternativen är att behandla det som en fristående enhet i ordbildningen eller att slå ihop *-it-* med det föregående grundledet *human* till *humanit-*. Detta senare framstår inte som någon lockande lösning; den kan bli aktuell först när grundmorfen inte kan kan fungera som simplex eller har någon klar betydelse. Vi står således med ett infix *-it-*. Andra sådana är det ovan nämnda *-en-*, *-nd-* som i *uppståndelse* där *-else* och *(upp)stå* kommuteras ut. Oftast ger väl infixen en bild av de rester som finns kvar i ordförrådet från äldre språkskeden eller från inlånat material.

Ett specialfall av ordbildning är att ordet byter syntaktisk funktion, d v s byter ordklass. Hur vanligt är det och vilka regler följer denna utveckling? Kortord som *bil* ur *automobil* (jämför *mobil*) ställer frågan om orden ska relateras till varandra på ett sådant sätt att *mo-* och *-bil* kommuteras ut eller inte. Det saknas alltså inte problem att ta ställning till.

Användningsområden

En detaljerad och genomtänkt märkning av ordleden ger möjlighet till sammanställningar med fyllig information. För internt bruk vid ordboksframställning och för forskningsändamål är det både möjligt och lämpligt att sortera sitt material på skiftande sätt alltefter de analyser som gjorts. Men även för en ordbok av ordinär typ kan man tänka sig en komprimerad morfotaktisk information. Sådan förekommer redan nu. Vissa avledningar och sammansättningar anges gärna under ett uppslagsord. Med det här skissade sättet att utmärka ordledstyper o s v, öppnar sig möjligheten att ge informationen i en mer systematisk form, t ex med angivande av vilken typ av ordbildning ett visst ord ingår i. Det skulle mer explicit än den alltid ofullständiga exemplifieringen kunna ange möjligheterna och göra oss mer medvetna om vilka faktorer som påverkar ordbildningen.

En frestande möjlighet, när man har en någorlunda komplett beskrivning av ordbildningsprinciperna, är att på grundval av explicita regler generera ord för att så testa reglernas hållbarhet.

Noter

1) Framställningen präglas säkerligen en hel del av erfarenheter från arbetet med Nusvensk frekvensordbok 4 (observera särskilt inledningen) och de många och långa diskussionerna med mina arbetskamrater och medförfattare till ordboken: Sture Allén, Sture Berg, Jerker Järborg, Bo Ralph och Christian Sjögreen. Jag tackar dem samt Åsa Abelin och Erland Gadelii som givit mig synpunkter på manus till detta föredrag.

2) Se t ex inledningen till Nusvensk frekvensordbok 4 och Alhaugs uppsats från Nordiska datalingvistikdagarna 1977.

Några referenser

Alhaug, G., Noen problemer ved opbygging av et datamaskinelt morfem-lexikon. Göteborgs universitet. Rapporter från Språkdata. 3. Nordiska datalingvistikdagar 1977, s. 5-12.

Nusvensk frekvensordbok baserad på tidningstext. 4. Ordled, betydelser. Data linguistica. 14. Almqvist & Wiksell International. Stockholm. 1980.

Liljestrand, B., Så bildas orden. Studentlitteratur. Lund 1975.

Söderbergh, R., Svensk ordbildning. Skrifter utgivna av Nämnden för svensk språkvård. 34. 2. uppl. Läromedelsförlagen. Stockholm 1971.

Helmer Gustavson
Riksantikvarieämbetet,
Stockholm

FÖRARBETEN TILL EN DATORISERAD RUNORDBOK

I det följande beskrivs arbetet med att upprätta ett ADB-baserat register över det språkliga materialet i Sveriges runinskrifter.Det skall också tjäna som en utgångspunkt för en planerad runordbok. Ur datorteknisk synpunkt kan det vara av intresse, eftersom det bygger på ett mikrodatorsystem och tillämpning av interaktiva program, som medger direktkommuniktion i klartext.Registrets innehåll kommer huvudsakligen att bygga på innehållet i inskrifterna i seriverket Sveriges runinskrifter (1900-). Registret är tänkt att bestå av två större delar: ett ordregister och ett register över de enskilda inskrifterna. Till ordregistret knyts, om så är lämpligt och möjligt, ett namnregister över personnamnen och ortnamnen i runinskrifterna.

Förutom att registren kommer att ligga lagrade för löpande ADB-behandling blir de också utgångspunkt för tryckta förteckningar, bl a i form av den nämnda runordboken.

Avsikten är att ADB-systemet skall göra runmaterialet i runinskrifterna snabbt och enkelt åtkomligt för olika sammanställningar och analyser och därigenom underlätta undersökningen och publiceringen av runinskrifterna i Sverige. Vi kan med systemets hjälp göra olika sorteringar och selektiva sökningar, som annar vore mycket tidsödande eller omöjliga att utföra. Vår avsikt är också att göra det tillgängligt för andra forskare i form av en "rundatabas".ADB-tillämpningen innebär också, att publiceringen av materialet kan ske på annat sätt än man tidigre haft möjlighet till. Den manuella bearbetningen medgav inte att man kunde göra omfattande och snabba ändringar i och redigeringar av ord- och namnförrådet i runinskrifterna. Sådana ändringar var nödvändiga, eftersom vissa resultat i de äldre utgåvorna var inaktuella eller felaktiga. Tanken på publicering av en ordbok sköts därför i realiteten på en avlägsen framtid, till den tidpunkt då man räknade med att ha ett adekvat material. Genom Runverkets datorsystem och ordbehandlingssystem räknar vi nu med att kunna göra preliminära utgåvor, med givna begränsningar, i form av ordlistor/namnlistor och att snabbt kunna införa rättelser och omredigeringar alltefter som vi har utvärderat det äldre materialet.

Registrets storlek

Materialet omfattar omkring 3500 runinskrifter i Sverige från förhistorisk tid och från nordisk medeltid. Huvuddelen består av inskrifter från 1000-talet och början av början av 1100-talet ("vikingatiden"). Det medeltida materialet utgör en fjärdedel. Denna runkorpus är publicerad i Sveriges runinskrifter, Danmarks Runeindskrifter (1941-42,Skåne,Blekinge och Halland) och Norges Innskrifter med de yngre Runer (1941- ,Bohuslän). I viss utsträckning har också

materialet i W.Krauses Die Runeninschriften im älteren Futhark (1966) använts. Det material som fr o m 1967 årligen publiceras i tidskriften Fornvännen under rubriken Runfynd 1966 osv ingår också i materialet.Detta tryckta runmaterial har tidigare blivit excerperat i två kortregister, ett ord- och namnregister på papperslappar enligt principen "en lapp för varje enskilt belägg" och ett inskriftsregister på nålkort med 100 sökmöjligheter. Det förra registret infattar omkring 27000 lappar, det senare omkring 3500 nålkort.

Kravspecifikation

När vi skulle välja datorsystem fanns följande krav, som skulle uppfyllas:

1.Det var nödvändigt, att den som inte hade någon erfarenhet av datorer skulle kunna köra systemet och ta ut information.

2.Det var önskvärt, att man hade direktkommunikation mellan användaren och datorn och att man uteslöt mellanliggande led i form av hålkort eller andra inmatningsmedia.Användaren skulle direkt kunna utbyta information med datorn och det skulle ske via en skärm eller på papper med hjälp av en printer.

3.Kommunikationen skulle ske enklast möjligt och i form av en direkt dialog i klarspråk.

4.Det skulle vara möjligt att distribuera den datorbehandlade materialet till intresserade forskare,universitetsinstitutioner och bibliotek på ett enkelt sätt: Antingen form av tryckt text eller med hjälp av disketter eller hårddisk som informationsbärare, vilka då skulle kunna användas i andra datoranläggningar. Dvs systemet måste vara översättningsbart till andra system.

5.Kostnadsfrågan var väsentlig.De begränsade tillgångarna skulle räcka till både maskinvara och programvara.

Dessa krav resulterade i valet av ett mikrodatorsystem med ett datorspråk som var lämpligt att hantera stora mängder alfanumeriska data.Vi fann det i datorsystemet ZILOG med programmeringsspråket Zilog Basic och PLZ. Det har senare kunnat kompletteras med ett ordbehandlingssystem.

Beskrivning av datorsystemet

Datorsystemet består av en centralenhet, två bildskärmar, en snabbskrivande matrisprinter och en långsammare skrivare. Systemets maximala kapacitet är 20 megabytes på hårddiskminne. Systemet gör det möjligt att arbeta "on line" både med bildskärmarna och skrivarna. Data levereras i behandlad form eller i rå form på papper eller bildskärm. Ordbehandlingssystemet utgörs av en METRIC 85 som gör det möjligt att internt redigera materialet före tryckning.

Runordsregistret

När man skall ställa i ordning ett material som runordsmaterialet för datorbearbetning är det nödvändigt att skapa en sammanhängande och logisk klassifikation. Den måste var tilämpbar på hela materialet och göra det möjligt att förmedla all den information som behövs.Eftersom runinskriftsmaterialet har publicerats under en mycket lång period och av olika forskare, så är det något oenhetligt och därigenom mindre lätthanterligt.Samtidigt har ett viktigt krav varit att minimera förberedelsearbetet för själva inskrivningen. En stor del av det hittillsvarande arbetet har därför varit att bygga upp en sådan klassifikation.

Problem som måste lösas

De principer som tillämpats vid publiceringen av Sveriges runinsskrifter har utvecklats efterhand. I de äldre utgåvorna används t ex en normalisering som avviker från den som används i de senare volymerna.

Även den grammatiska och semantiska terminologin varierar något i de olika utgåvorna. Ett annat problem av lexikalisk art är översättningen till nusvenska. Skall en etymologisk princip eller en semantisk princip tillämpas, skall t ex gæra bru översättas med "göra bron" eller "göra vägbanken" ? Vi har huvudsakligen följt den etymologiska principen, men är medvetna om dess begränsningar.
Problem av ett annat slag är de som hör samman med datorns teckenuppsättning. Ett sådant är datorns standarduppsättning av skrivtecken, som inte kan återge alla de specialtecken som används vid translitterationen av runorna med latinska bokstäver. Antingen kompletterar man den egna anläggningen med dessa specialtecken, eller också använder man standarduppsättningen. Vi har valt det senare.
Problem som återstår att lösa i fråga om ordregisterdelen är bl a frågan om vilken klassifikation som skall användas om ett separat namnregister läggs upp och hur det i så fall skall konstrueras, så att det passar in den allmänna strukturen i det existerande registret.

Beskrivning av registret
1.Registrets idé framgår av följande:
Fetstilsbeläggen utgör grunden. De organiseras efter normaliserad runsvensk form (utom de otolkade, som organiseras direkt utifrån fetstilsbelägget).Runregistret delas upp i tre delar, var och en med sin egen teckenuppsättning: urnordiska, vikingatida och medeltida runor.
2.Registrets struktur:

Post nr	Innehåll
1	Normaliserad runsvensk form.
2	Ordgrupp.
3	Nusvensk översättning som täcker hela betydelsefältet. Synonymer undviks.
4	Eventuella generella kommentarer avseende posterna 1-3.
5A	1:a fetstilsbelägget.
5B	Kontextbetydelse (anges då så anses nödvändigt).
5C	Uppgift om fetstilsbelägget.
5D	Eventuella kommentarer till fetstilsbelägget.
5E	Inskriftsförteckning
6A	2:a fetstilsbelägget etc.

Ex:

Post nr	Innehåll
1	stäinn
2	NSM
3	sten
4	-
5A	stain
5B	-
5C	ac,s
5D	-
5E	U117,U171,U244,Sö234,+Vr 4
6A	stin
6B	-
6C	ac,s
6D	-
6E	U240,+U176,U110B etc

Observara att stain U168 ac,s och stain U164 ac,p skall föras till olika fetstilsbelägg eftersom de getts olika tolkningar.

2.Som nämnts kommer runordsregistret att delas upp i en urnordisk, en vikingatida och en medeltida del. Transkriptionsreglerna blir av naturliga skäl något olikartad för de olika runalfabetena. Förutom de olika tecknen för translittereringen av runorna används också en mängd specialtecken för att t ex ange att runinskriften är förlorad, ange supplering,osäkerhet i läsningen av en runa,ange brott eller skada i inskriften på runstenen, beteckna specialruna (t ex binderuna) eller markera runa som inte kan bestämmas till sitt teckenvärde.
3.Varje ord i registret hänförs till exakt en av följande 21 ordgrupper:

Ordgrupp	Mnemoteknisk kod
Personnamn,mansnnamn	NPM
" ,kvinnonamn	NPK
Ortnamn	NO
Substantiv,mask	NSM
" ,fem	NSF
" ,neutr	NSN
" ,obestämt genus	NS
Adjektiv	AD
Verb	VB
Adverb	AV
Preposition	PP
Pronomen	PN
Konjunktion	KJ
Räkneord,grundtal	ROG
" ,ordningstal	ROO
Interjektion	IJ
Infinitivmärke	IM
Artikel	AT
Partikel	PT
Negerande suffix	SX
Otolkad	OT

4.Fragmentariska ord anges under ordgruppen OT (förutsatt att man inte kan ge ordet en tolkning), varvid ... anger var runor fattas,om detta kan avgöras.

5.Till varje tolkat ord hör en normaliserad runsvensk form, som är: nom.sing.obest.form för substantiv,nom.sing.mask.positiv stark form, om den finns, för adjektiv och räkneord, positiv för adverb,infinitiv för verb, nom.sing.mask.för pronomen (obs undantaget hon, som listas separat),nom.sing.mask. för artikel, nom.för personnamn,nom. för ortnamn, övriga ordgrupper anses oböjliga.

6.Uppgifter om fetstilsbelägget:

Huvudord	Kod	Uppgift
Substantiv	b	bestämd form
"	nm,ge,da,ac	kasus:nom.gen.dat.ack.
"	s,p	numerus:sing.resp. plur.
Adjektiv	ma,fe,ne	som för substantiv + genus
"	po,ko,su	" " " +komparationsform
"	st,sv	" " " + stark el. svag böjning.
Verb	hu, hj	huvudverb resp.hjälpverb
"	ind,kon,imp	modus:indikativ,konjunktiv,imperativ
"	akt,pas,dep	aktivum,passivum,deponens

Huvudord	Kod	Uppgift
Verb	inf,pre	infinitiv,presens
"	pret,fut,kns	preteritum,futurum,konditionalis
"	per,plu	perfekt,pluskvamperfekt
"	prp,pfp	presens particip,perfekt particip
"	1s,2s,3s	1:a,2:a,3:e person sing.
"	1p,2p,3p	" " " " plur.
"	ma,fe,ne,s,p	till participer anges genus+numerus
Adverb	po,ko,su	komparationsform
Prepositioner	ge,da,ac	kasusstyrning
Pronomen	ma,fe,ne,s,p	genus,numerus,kasus
Räkneord	" " " " "	" " " samt stark resp svag
Artiklar	" " " " "	" " "
Personnamn	nm,ge,da,ac	kasus
Ortnamn	" " " "	"

Övriga ord anses oböjliga.

7.För de ordgrupper som böjs med avseende på kasus gäller följande: Kasus anges där detta klart framgår, antingen genom böjningsform eller funktion i satsen. I de fall där dessa ej överensstämmer anses satsfunktionen ha högre prioritet.

8.För verben gäller följande:
I samtliga konstruktioner med hjälpverb+huvudverb anges hjälpverbet separat under "naturligt" tempus.Ex.häfiR pre.Vid huvudverb anges även hjälpverbet inom parentes i sin normaliserade runsvenska infinitivform.Ex.mun raþa: Under raða anges: hu,akt,inf,fut (munu) i nu nämnd ordning.För huvudverbet anges alltså såväl formell form som kontextuell betydelse. Om ett verb står utan hjälpverb utelämnas hu-kodningen. I de få fall där flera hjälpverb är kopplade till ett huvudverb, såsom i exemplet mun kietit lata, anges det överordnade (mun) på vanligt sätt. Det underordnade hjälpverbet anges som hjälpverb, med med både formell form och kontextuell betydelse (här inf,fut), Vid huvudverbet anges endast det överordnade hjälpverbet men med uppgift om konstruktionen i separat kommentar. Löst och fast sammansatta verb stå som de uppträder i runinskriften, dvs (när partikel och stam är delade åtskiljs de med mellanrum. När ord förekommer mellan partikel och stam avlägsnas dessa och detta markeras med en punkt.

9.Olika stavningsformer av ett ord (t ex aft och at) sammanförs under en normaliserad form då stor säkerhet om samhörighet råder. I den generella kommentarposten anges de olika varianterna.

10.För homografer gäller följande:
De delas upp efter betydelse och markeras med ett ordningsnummer omedelbart efter den normaliserade formen.Ex:

æinn1	räkneord
æinn2	adjektiv ("ensam")
æinn3	pronomen

De behandlas på följande sätt:æinn1,æinn2,æinn3 anges under respektive ordgrupp, givetvis anges osäkerhet med ? Om det inte på något sätt är möjligt att ge någon tolkning företräde framför de andra bör ordet föras under ordgruppen Otolkad (OT).

För närvarande pågår inskrivning av ordmaterialet på hårddisk. Samtidigt används det inskrivna materialet vid Runverkets språkliga undersökningar. Sökningsmöjligheterna i materialet är under utveckling. Följande sökningar kan göras för tillfället:

Man kan söka i 7 av registren samtidigt. I register 3 (Nusvensk översättning),6 (Kontextbetydelse) och 7 (Fetstilsuppgift) kan man dessutom söka med flera sökord på en gång (4,2 resp 8) med "," emellan. Om man exempelvis vill undersöka vilka maskulina substantiv som står i genitiv plural anger man NSM som sökord i post 2 och ge,p som sökord i post 7. Man kan också manipulera strängar enligt följande regler: 1.Tecken i början av ett sökord kan ersättas med $-tecken, t ex hittar$uR alla ord som slutar på -uR. 2.Tecken i slutet av sökord kan ersättas enligt samma princip, t ex hittar st$ alla ord som börjar på st-. 3.Om man vill finna en teckenföljd kan denna sökas med omgärdande $-tecken. Ex: uan hittar alla ord med teckenföljden uan, t ex i fetstilsbelägget buanta. 4. Man kan även utesluta delpostkombinationer genom att börja den första delposten med ø. Ex: Sökordet øne,ac i posten för fetstil ger alla fetstilsbelägg som inte är ne,ac. 5. Retrograda sökningar kan göras.

Vi strävar också efter att åstadkomma sökningar i kommentarposterna i form av en "fri sökning". Möjligheterna till det måste avvägas mot de strukturproblem som det för med sig och påfrestningen för datoranläggningens kapacitet och tidsförluster i sökningen.

Karen Margrethe Pedersen
Institut for dansk Dialektforskning

OM ANVENDELSE AF ET TEKSTKORPUS TIL SUPPLERING AF ORDBOGSMATERIALE.

Ved Institut for dansk dialektforskning arbejdes der på Ømålsordbogen, en ordbog over dialekterne på Sjælland, Lolland-Falster, Fyn og de omliggende mindre øer.

Materialet til ordbogen er indsamlet ved at dialektforskere er rejst ud og har optegnet ordstoffet eller ved at lægfolk har sendt oplysninger ind til instituttet, ofte som svar på spørgelister. En lille del af materialet udgøres af ældre, til dels utrykte ordbogsarbejder og af ekscerpter fra dialektlitteratur. 1)

Ordbogssamlingen omfatter ca. 2 mill. ordsedler. I langt de fleste tilfælde er seddel-materialet så omfattende, at der kan skrives rimeligt gode og fyldige ordbogsartikler på grundlag af det. Men til de mest almindelige ord (præp., konj. mm) er materialet for spinkelt. Disse ord er ikke blevet ekscerperet fra materialet i tilstrækkeligt omfang, netop fordi de er så almindelige og lidet påfaldende.

De forekommer til gengæld hyppigt i instituttets store samling af båndoptagelser med dialektalt talesprog. Dele af båndoptagelserne er aflæst i rigsmål og indlæst på magnetbånd. 2) Vi har således et tekstkorpus (nu på ca 12.000 linjer fra ømålsområdet og 13.000 linjer fra Jylland), hvor vi kan lade maskinen søge efter ord og udskrive dem med kontekst og tekstreferencer. Der søges hver gang en redaktør finder det nødvendigt, og det har bl a været tilfældet ved ordene altså og god.

Til redaktion af altså var der kun 17 sedler i samlingerne (mod 117 sedler på et ord som ambra). Kun på 2 af sedlerne var ordet brugt "med svækket betydning for at udtrykke sammenhæng med det foregående". Da det måtte formodes, at denne betydning var almindelig overalt, var det nødvendigt at supplere materialet. Suppleringen blev foretaget fra ovennævnte tekstkorpus (der dengang var på ca. 4.500 linjer) og gav tilstrækkeligt mange belæg til at vise, at den afsvækkede betydning var den almindelige, at den var udbredt over hele ømålsområdet, og at svækkelsen kunne gå så vidt, at ordet blev et rent "fyldeord". Suppleringen gav således en væsentlig forbedring af ordbogsartiklen.

Det var påfaldende, at der var stor forskel på frekvensen af altså i de enkelte tekster. Der var fx én tekst med 0 forekom-

ster i 494 linjer og én med 25 forekomster i 328 linjer. Frekvensforskellene var ikke geografisk betinget, men må skyldes individuelle variationer.

God, godt hører til de forholdsvis store ordbogsord med mange betydninger og mange faste forbindelser (i god gænge, af gode grunde, sige god for osv). Seddelmaterialet til ordet var spinkelt, men en supplering fra tekstkorpus'et gav knapt 150 nye belæg (i knap 6000 linjer tekst). Belæggene samlede sig i nogle få betydninger og inden for disse endda om visse underbetydninger. Således var 86 af de første 100 belæg fordelt på 3 af ialt 10 betydninger, mens resten af betydningerne enten var meget svagt belagt eller slet ikke belagt. Også de faste forbindelser var meget svagt belagt.

Det var ikke overraskende, at nogle betydninger var mere frekvente end andre, men for ordbogsredaktøren, der sidder med hele spektret af betydninger og underbetydninger, var det fascinerende og også lidt chokerende at se, hvor få og hvilke betydninger der var almindelige i brug.

For at få belæg på de sjældne betydninger og de sjældne ord skal man have et meget stort tekstkorpus. Men da det er de almindelige ord og ofte de almindelige betydninger af disse, der er svagt belagt i seddelmaterialet, kan selv et lille tekstkorpus som vores være et nyttigt supplement.

Tekstudskrifterne har også vist sig at være et nyttigt supplement til citatmaterialet. Citaterne bringes efter hver betydningsangivelse for at underbygge betydningsangivelsen og for at give eksempler på ordets brug i talesproget. Tekstudskrifterne giver autentiske eksempler, mens seddelmaterialet i mange tilfælde giver eksempler, der er konstrueret til lejligheden. Og udskrifterne indgår i en større kontekst, mens seddelmaterialet ofte mangler kontekst. På sedlerne kan der stå sætninger som han var afskåret fra at gøre det. Sådanne sætninger oplyser noget om ordenes syntaktiske muligheder (det hedder afskåret fra), men er iøvrigt næsten tomme for indhold (hvem var afskåret fra hvad og hvorfor). Hele konteksten (situationen) mangler. 3) Ved tilsvarende sætninger i tekstudskrifterne har man adgang til konteksten, og hvis sætningerne anvendes som citater, kan konteksten antydes i en parentes. Erfaringen har vist, at udskrifterne giver gode citater.

Noter.

1) se Inger Bévort: Indsamlingen af materiale til Ømålsordbogen i Dialektstudier. 1 (1964): 239-25o.

2) se Karen Margrethe Pedersen: Dialekttekster i rigsmålsnotation med becifring i Danske Folkemaal. 2o (1974): 29-46.

3) se Finn Køster: Ordbog over de danske ømål. Nogle problemer i betydningsafsnittene i Sven Bensson mfl (ed.): Dialektologkonferens 1978 (Göteborg 1978).

Håvard Hjulstad
Norsk termbank /
Norsk leksikografisk institutt

DATABEHANDLING AV NORSK HANDORDBOK

Det har vore arbeidd redaksjonelt med handordbøkene, ei for bokmål og ei for nynorsk, sidan 1974 som eit samarbeidsprosjekt mellom Norsk språkråd og Norsk leksikografisk institutt. Ordbøkene skal dekkje allmennspråket, og innanfor ei ramme på om lag 900—1000 sider skal dei gi ortografi, bøying, uttale, etymologi, synonym/definisjonar og døme på bruk.

Heilt frå starten var det meininga å kople datamaskina inn i arbeidet. Frå førsten var det til å velje ut ordtilfang og særleg til å samordne ordtilfanget i dei to utgåvene ein tenkte seg at datamaskina kunne yte ein innsats. Frå før var det lagra ved Prosjekt for datamaskinell språkbehandling i Bergen eit stort tilfang på bokmål og nynorsk. Ein tenkte seg at samsorterte lister av dette tilfanget kunne vere startgrunnlaget for redigeringa. Dette vart ikkje følgt opp, og redigeringa har heile tida vore reint manuell.

I 1979 kom ein i gang med dataføringa av nynorskutgåva. Det var lagt opp etter eit system som utvikla seg frå det som vart brukt til behandlinga av Norsk landbruksordbok. Innskrivingsformatet ligg nær opp til trykk; ein "simulerte" setjeri i innkodinga. Rett nok var kodesystemet meir fininndelt enn det som kjem fram i trykk. Det var etter måten enkelt å produsere ei trykt bok ut frå dette formatet på data, men vitskapleg bruk av data var heller tungvint.

I byrjinga av 1981 fann vi det føremålstenleg å leggje om dette opplegget. I datamaskinbaserte ordboksprosjekt snakkar ein no stadig meir om felt, og også Norsk handordbok har fått sine felt merkte i data.

Litt feltfilosofi

Noko er felles frå ordbok til ordbok.

gutunge|n
gut ... -unge som unge
gutunge gutungen gutungar gutungane

Det står det same, men på ulike vis. I maskinversjonen må ein få fram at det er det same, men på trykk må ein få lov til å uttrykkje det ulikt. Også andre opplysningar kan uttrykkjast ulikt om dei tyder det same.

I den grad det er råd å gi ei opplysning i ein kode eller ei anna "reinsa" form, bør denne forma finnast i maskinversjonen. Også i det andre av dei tre døma ovanfor må det vere råd å leite maskinelt på "gutunge".

Somtid er det mest føremålstenleg å analysere den redigerte "trykkdelen" og dra ut slike opplysningar. Andre gonger kan ein gå andre vegen og generere det som skal trykkjast ut frå dei koda opplysningane.

I Norsk handordbok gjer ein det på første måten. Redaktørane får "drive med sitt" utan å leggje om arbeidsforma. Men det er jo å vone at framtidige prosjekt kan endre på arbeidsforma på dette punktet. Da blir det truleg enda meir å tene på det også.

Ein post er no samansett av ein prosjektuavhengig del og ein eller fleire prosjektavhengige delar. Kvar av desse delane inneheld i regelen fleire felt. Når ein er ute etter ei opplysning, leiter ein berre i det feltet der ein kan vente å finne denne opplysninga.

Ned på jorda

Nokre postar i Norsk handordbok (nynorskutgåva) ser no slik ut:

```
NN001    kabrette
NN001a   F2
TR006
..OPP    £>$Ckabret>te@ f2
..ETY    (truleg sm o s norr tilnamn $Bkakbretta@ kanskje
         smh med $BI kabbe@ 'klump')
..DEF    ostevelling av innkokt myse
=
NN001    kabel
NN001a   M1t
TR006
..OPP    £>$Ckabel@ $B-en, -blar@
..ETY    (gj lty $Bkabel@ og fr $Bc$3able@ frå lat. $Bcapulum@ 'reip'
         av $Bcapere@ 'ta')
..DEF    $C1@ kraftig trosse av tau el. ståltråd
..DEF    $C2@ elektrisk leidning med kraftig isolering
=
NN001    kabelfarty
NN011    kabelfartøy
TR006
..OPP    $C~farty@ $C<<~fartøy>>@
..DEF    farty som legg sjøkabel
=
NN001    kabelfjernsyn
TR006
..OPP    $C~fjernsyn@
..DEF    fjernsynsoverføring til mottakaren delvis med
         hjelp av *kabel (2)
=
NN001    kabelgatt
NN002    kabelgat
TR006
..OPP    $C~gat(t)@
..DEF    rom i farty for tauverk
=
```

(Merknader til utskrifta:
Feltkodar: NN tyder "nynorsk", 001 tyder "første hovudform", 001a tyder "grammatisk kode til første hovudform", 011 tyder "første sideform", TR006 tyder "trykkdel prosjekt nr 006", "trykkdelkodane" som tek til med .. skulle vere sjølvforklarande. Trykkodar: £> tyder "nytt avsnitt i trykk", $B ..@ tyder "kursiv skrift", $C .. @ tyder "halvfeit skrift". << og >> står for hakeparentes.)

Det er fleire opplysningar som kan få plass "på toppen" enn dette. Det skal fyllast ut med bøyingskodar. Kanskje noko formalisert etymologi, og i alle høve formalisert semantikk bør ein freiste å "pine" ut av data. Formalisert syntaks skulle også gjerne vore der, men ordboka har få eller ingen opplysningar om dette.

Feltinndelinga i den prosjektavhengige delen er "lausare". Her må ein tillate variasjonar frå prosjekt til prosjekt.

Innskrivinga

Innskrivinga har vi freista å gjere så enkel som råd med å bruke eit enkelt feltkodesystem. Dei første postane i dømet ovanfor vart skrivne inn slik:

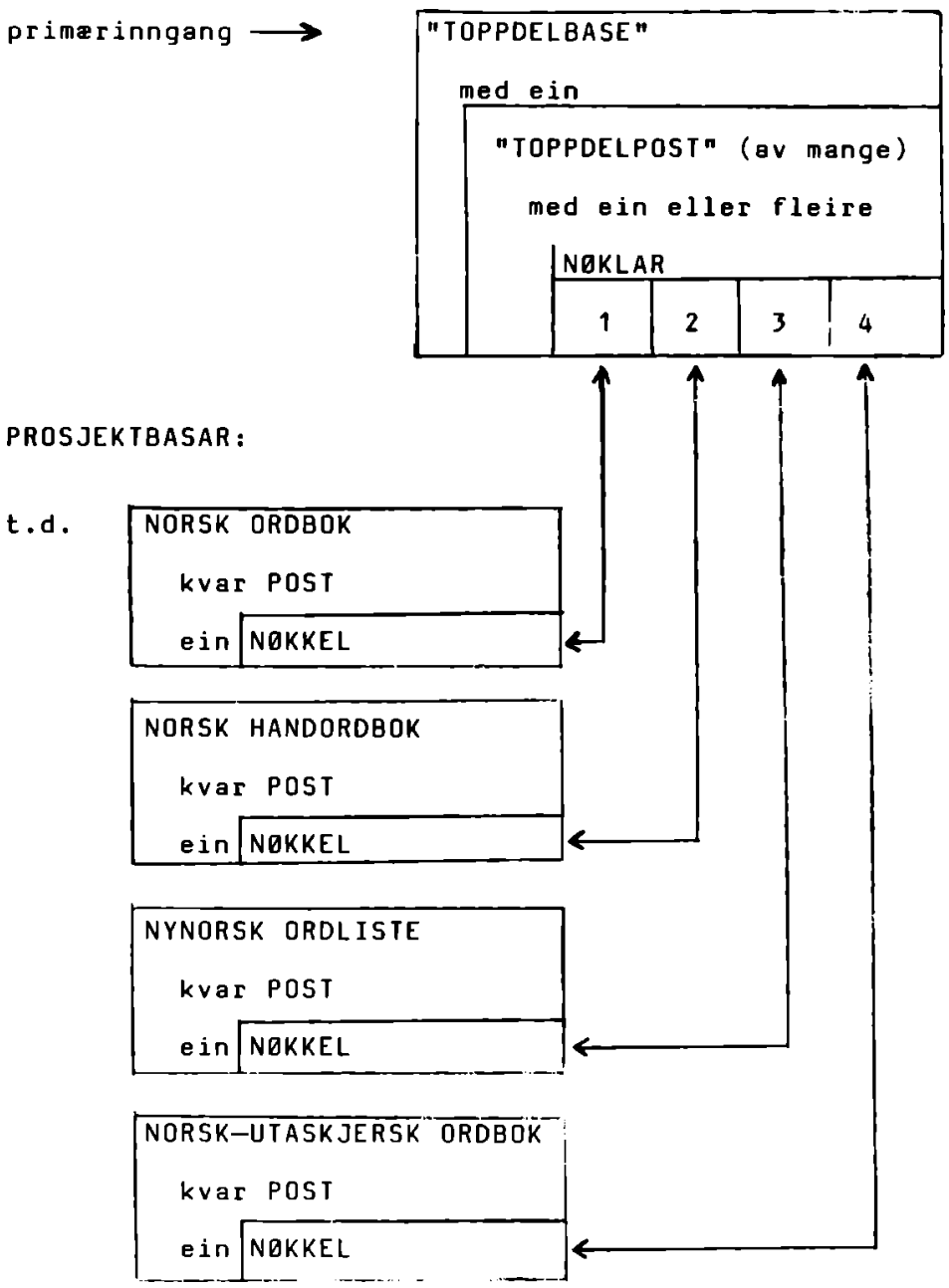
```
O kabret>te
G f2
E (truleg sm o s norr tilnamn $Bkakbretta@ kanskje
 smh med $BI kabbe@ 'klump')
D ostevelling av innkokt myse
=
O kabel
B -en, -blar
E (gj lty $Bkabel@ og fr $Bc$3able@ frå lat. $Bcapulum@ 'reip'
 av $Bcapere@ 'ta')
D1 kraftig trosse av tau el. ståltråd
D2 elektrisk leidning med kraftig isolering
-
```

Ein sparer mykje skrifttypekoding med å gjere det slik. Som ein ser er felta O, G, B slått saman i ..OPP-feltet i utskrifta på førre sida. Opplysningane er her fullstendig analyserte til "toppdelen", og den finare feltinndelinga er ikkje lenger nødvendig.

Litt framtid

Når eit prosjekt er ferdig, tek ein med seg "toppdelen" til neste prosjekt. "Trykkdelen" får leve sitt eige liv, men via ein identifikasjonsnøkkel som skal få plass i begge delane, kan ein seinare kople dei saman.

Det kan godt vere at ein flik av framtida vil sjå slik ut:



Tove Fjeldvig
Institutt for privatretts avdeling for EDB-spørsmål
Niels Juels gate 16 - Oslo 2

UTVIKLING AV ENKLE METODER FOR TEKSTSØKING MED SØKEARGUMENTER I NATURLIG SPRÅK

1. Innledning

Det er spesielt 2 grunner til at det kan være ønskelig med spørsmål i naturlig språk i et fulltekstsøkesystem:

a) Søkeargumenter i naturlig språk stiller ingen eller få krav til forkunnskaper hos brukeren, og vil derfor gjøre det lettere for både nybegynnere og tilfeldige brukere å anvende systemet.

b) Enkelte søkeargumenter lar seg best formulere i naturlig språk. F.eks. at en jurist i en gitt sak blir presentert for en dom og ønsker å kontrollere hvorvidt det finnes andre dommer som angår samme spørsmål. I et slikt tilfelle vil søkeargumentet kunne bestå av f.eks. et sammendrag av dommen kombinert med med "FINN DOKUMENTER SOM LIGNER".

Nå vil imidlertid en uerfaren bruker også kunne anvende dagens tekstsøkesystemer uten all for mye veiledning, men da kun på det helt enkleste nivå. For å kunne anvende systemet effektivt og oppnå gode resultater, kreves lang erfaring og godt kjennskap til hvordan man kan utnytte systemets finesser.

De siste årene har det vært en økende interesse for søkesystemer med muligheten for spørsmål formulert i naturlig språk. Forskningen har primært vært rettet mot såkalte "spørsmål-svar" -systemer hvor man søker etter et konkret svar på et problem - og ikke sekundært informasjon i form av dokumenter. I mindre grad har forskningen vært rettet mot dokumentgjenfinningssystemer. Vi kjenner til bare noen få eksempler på dette området (som prosjektene CONDOR (tysk), SYSTEX (fransk), POLYTEXT (svensk) og RESPONSA (israelsk).

Vi fant det interessant å se nærmere på muligheten for spørsmål i naturlig språk i dokumentgjenfinningssystemer, og spesielt rettet oppmerksomheten mot muligheten for en slik strategi som et alternativ til eksisterende søkestrategier. Vi valgte derfor å konsentrere arbeidet om enkle og lite ressurskrevende metoder og legge vekten på brukerkrav som responstid, bruk av lagringsplass og - ikke minst - effektiv oppdatering av "databasen".

Prosjektet ble initiert i begynnelsen av 1979 og er planlagt avsluttet ved utgangen av dette året. Arbeidet blir ledet og gjennomført av undertegnede, og prosjektet mottar økonomisk støtte fra Norges Teknisk- Naturvitenskaplige Forskningsråd.

2. Gjennomføring

Prosjektet har vært gjennomført som en serie med kontrollerte forsøk i tekstsøking

> Et kontrollert forsøk i tekstsøking går ut på at man for en gitt dokumentsamling definerer et sett med spørsmål som er aktuelle for samlingen. For hvert spørsmål går man gjennom alle dokumentene i hele samlingen og merker av de dokumenter som er relevant i forhold til spørsmålet (fasiten). Spørsmålene og fasiten defineres av de samme personer, og vi har lagt stor vekt på at vedkommende har godt kjennskap til dokumentsamlingen og dens fagområde.
>
> Spørsmålene danner grunnlaget for den maskinelle søkingen, og søkeresultatet blir sammenlignet med fasiten. Dette gir oss muligheten til å måle hvor mange av de relevante dokumenter som er funnet ved den maskinelle søkingen (recall), og hvor mange av de funne dokumenter som er relevante (presisjon).

Kontrollerte forsøk gjør det mulig til enhver tid å få feedback-informasjon om effekten av endringer i de valgte metoder. Dette bidrar til at vi kan styre forskningen i det vi tror er en riktig retning.

For å få spredning i dokumentmaterialet, har alle forsøk vært gjennomført på 3 ulike dokumentsamlinger:

a) 350 uttalelser fra Skattedirektøren, ca. 82 000 termer,

b) 1020 sammendrag av lagmannsrettsavgjørelser i familieskifte- og arverett, ca. 190 000 termer,

c) 1270 tinglysningsavgjørelser, ca. 218 000 termer.

Til hver dokumentsamling er det knyttet ca. 30 spørsmål med referanser til relevante dokumenter. Spørsmålene er definert av jurister med solid forankring innenfor dokumentsamlingens rettsområde. Referansene til de relevante dokumenter er stilt opp på grunnlag av en manuell gjennomlesing av alle dokumenter i hele samlingen.

3. Problembeskrivelse

Det ideelle resultat i et tekstsøkesystem (dokumentgjenfinningssystem) er å nå fram til alle og bare dokumenter som er relevant i forhold til brukerens problemstilling. Å konstruere et system som til enhver tid oppnår et slikt resultat, er ikke mulig. For det første fordi ordene i seg selv - eller språket - ikke er entydig, og for det andre fordi to forskjellige brukere ofte kan ha ulik forståelse av spørsmålet (søkeargumentet).

I de tradisjonelle tekstsøkesystemer, som f.eks. STAIRS, IMDOC, LEXIS og NOVA*STATUS, velger man bevisst søketermer som representerer idéene i problemstillingen, og som man forventer å finne i de relevante dokumenter. Dette vil ikke være tilfellet i et system basert på spørsmål i naturlig språk. Brukeren vil her velge ord og formuleringer med sikte på å bli forstått av et annet menneske. Man står med andre ord ovenfor ulike situasjoner i de to tilfellene, og dette er en av årsakene til at jeg tror at det er vanskelig - om ikke umulig - å utvikle et system basert på spørsmål i naturlig språk som gir like gode søkeresultater som et system av den første typen.

Av ressursmessige hensyn valgte vi å konsentrere arbeidet rundt selve spørsmålet og i så liten grad som mulig ta i bruk informasjon som krever nærmere analyse av dokumentmaterialet. Figur 1 på neste side gir et bilde av hvordan vi forestilte oss søkestrategien, og på denne bakgrunn ble prosjektarbeidet splittet i følgende hoveddeler:

a) Identifisering av søketermer og fraser i spørsmålet,
b) Utvidelse av søkeargumentet med synonymer,
c) Valg av regler for utvelging og rangering av dokumenter,
d) I hvilken grad tilbakeføring av informasjon til brukeren (feedback-informasjon) kan bidra til å øke søkeeffektiviteten til systemet.

Fram til i dag har oppmerksomheten vært festet til de 3 første punktene. I dette notatet vil jeg konsentrere meg om arbeidet under punkt a) og b).

4. Identifisering av søketermer og fraser i spørsmålet

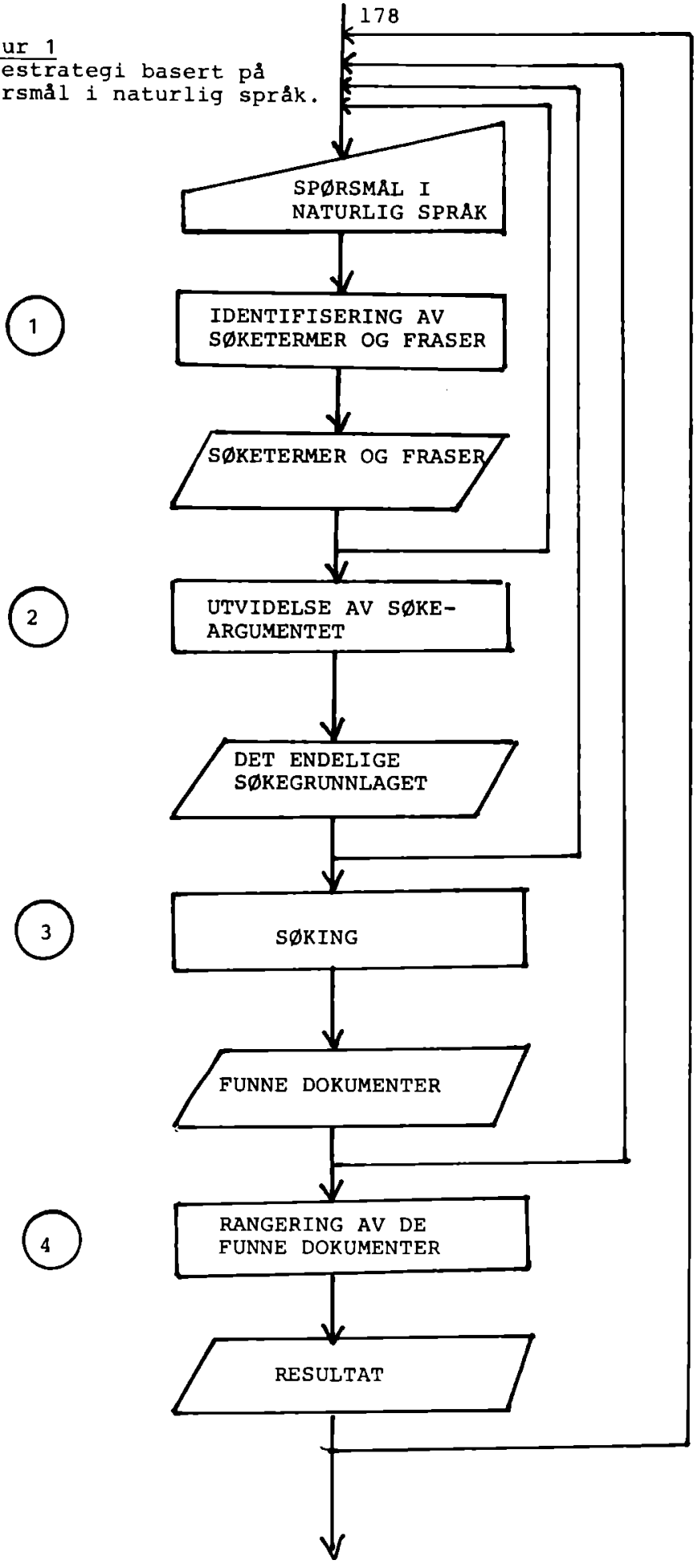
En søketerm skal ha den egenskapen at den kan bidra til å skille relevante dokumenter fra irrelevante. Betrakter man et spørsmål i naturlig språk, vil man finne at det er svært få ord som egentlig har denne egenskapen. De fleste ordene vil - betraktet isolert fra konteksten - være lite karakteristiske for selv meningsinnholdet i spørsmålet, men de vil ha en viktig funksjon i kommunikasjon med et menneske. La oss f.eks. betrakte spørsmålet:

"FINNES DET NOEN DOKUMENTER OM NORSKE TEKSTSØKESYSTEMER?"

Her vil kun ordene NORSKE og TEKSTSØKESYSTEMER ha betydning for søkeprosessen, mens de øvrige ordene vil bare skape støy i søkeprosessen. De har derfor fått betegnelsen støyord.

En søking kan også baseres på bestemte ordkombinasjoner som f.eks. et navn, et uttrykk eller en henvisning. For å kunne identifisere denne type ordstrenger i spørsmålet - eller fraser som vi har valgt å kalle dem i mangel på et bedre uttrykk - krever det enten kjennskap til disse på forhånd eller en nærmere analyse av spørsmålet.

Figur 1
Søkestrategi basert på spørsmål i naturlig språk.



Vi har foreløbig valgt å begrense arbeidet til de enkelte termer i spørsmålet og ikke ta hensyn til hvilke relasjoner som foreligger mellom dem. Et unntak er gjort når det gjelder tall, men dette vil jeg komme tilbake til nedenfor.

I steden for å forsøke å identifisere søketermene direkte, valgte vi å gå den motsatte vei å betrakte termer som ikke er egnet som søketermer - dvs. støyord.

Manuelt forsøk

For å få en nærmere innsikt i hva som karakteriserer støyord og omfanget av dem, ble det gjennomført et manuelt forsøk i å identifisere dem.

To personer fikk i oppdrag hver for seg å gå gjennom alle ulike termer i hele dokumentsamlingen og merke av de termer som ikke var egnet som søketermer. Begge hadde lang erfaring med tekstsøkesystemer og kjente godt til både prosjektet og den aktuelle dokumentsamling.

Forsøket var interessant på mange måter. For det første viste en sammenligning av resultatene fra de to, at knapt halvparten av de utvalgte termene var felles. I følge senere samarbeid kom det fram at dette i liten grad skyldes uenighet, men i større grad unøyaktigheter og mangel på fantasi. De hadde ofte oversett termer som helt opplagt var støyord eller glemt å få med alle grammatikalske varianter av ordene. Siden forsøket var basert på en skjønnsmessig vurdering av det enkelte ord, måtte man ofte forestille seg aktuelle problemstillinger før man kunne ta stilling til ordet. Mangel på fantasi var derfor en av årsakene til denne uoverensstemmelsen.

Resultatet viste bare hvor vanskelig denne type arbeid er - selv innenfor en nokså homogen dokumentsamling. En term kan gjerne være relevant i forhold til en problemstilling, men totalt uten interesse for en annen.

For det andre viste forsøket at en manuell utvelgelse av støyord på denne måten er svært tidkrevende - selv for små dokumentsamlinger. Det var faktisk nødvendig å gå gjennom hele ordlisten opp til flere ganger før den endelige støyordlisten forelå.

Eksperimentet ble gjennomført for 2 ulike dokumentsamlinger med henholdsvis ca. 7000 og 9000 ulike termer. Begge forsøkene ga de samme erfaringer.

Støyord i relasjon til ordklasser

Resultatene av det manuelle forsøket førte til at det ble rettet enda større oppmerksomheten om utviklingen av maskinelle metoder for identifisering av støyord. Det synes å være klart at hyppigheten (frekvensen) til en term og antall dokumenter termen forekommer i (spredningen), er av stor betydning for hvorvidt termen er egnet som søketerm eller ikke. Forekommer en term f.eks. i alle dokumentene, er det klart at den vil mangle evnen til å skille relevante dokumenter fra irrelevante.

Oversikten over støyord pekte også på interessante sammenhenger mellom ordklasser og støyord. Jeg skal nedenfor se litt nærmere på disse.

De mest typiske støyord finner vi blant ord som preposisjoner, pronomen, konjunksjoner, interjeksjoner og artikler. Dette er alle ord som forekommer svært hyppig og som i seg selv er lite meningsbærende. Fordelen med disse ordene er at de forekommer i et begrenset antall og kan derfor lett defineres én gang for alle.

På lik linje med disse ordene finner vi også en rekke adverb som SÅ, OFTE, DA, NÅR, NETTOPP, GANSKE, VISST osv. Dette er en langt større ordgruppe og som vanskelig lar seg gjenkjenne uten en gjennomlesing av alle ordene i hele samlingen. Litt hjelp vil man imidlertid også kunne ha av en frekvensordliste

En annen type adverb er de som står direkte knyttet til verbet og forteller noe om hvordan ting skjer. På samme måte vil adjektiv stå direkte knyttet til substantivet og forteller noe om hvilke egenskaper det har. Felles for disse ordene er at de kan bidra til å karakterisere en problemstilling hvis de betraktes i sammenheng med de ordene som de står knyttet til, men vil de egne seg som søketermer alene?

Forsøksresultatene viste imidlertid at de spilte en positiv rolle i søkeprosessen. De bidrog til at søkeordfrekvensen i de relevante dokumenter økte mer enn i de irrelevante. Dermed ble det lettere å skille ut de relevante dokumenter. På den annen side bidrog de også til at flere irrelevante dokumenter ble funnet, men denne ulempen kan man til dels unngå ved å se bort fra dokumenter som er funnet bare på grunnlag av adjektiv eller adverb.

Adjektivene vil i mange tilfeller kunne la seg identifisere automatisk enten ut fra særegne suffikser eller på grunnlag av deres posisjon mellom en artikkel og et substantiv (forutsatt at disse er kjent). Den første metoden er blitt testet i prosjektet.

En interessant type fraser er de som inneholder tall eller tallord. Problemet med tall som søketermer, er at de forekommer svært hyppig og i en rekke ulike sammenhenger - ikke alle like interessant for tekstsøking. F.eks. ved søking i et juridisk dokumentmateriale vil tall spille en sentral rolle som del av en lovhenvisning eller dato, men mindre interessant som punktbenevnelse. Siden vi i dette prosjektet ikke har planer om å se nærmere på muligheten for å søke innen et gitt intervall (f.eks. at antall arvinger i et arveoppgjør skal være mindre enn 3), er det også mindre interessant å betrakte tall som spesifikasjon av et kvantum.

Vi valgte å konsentrere arbeidet om tilfeller hvor tallet forekommer som en del av identifikasjonen til en frase, f.eks. "SIDE 9", "PARAGRAF 4" eller "ÅRET 1918". Det er interessant å merke seg at også ordene i denne type fraser (som SIDE, PARAGRAF og ÅRET) forekommer så hyppig, at de i enkelte tilfeller kan ha en negativ effekt på søkeresultatene hvis de behandles som individuelle søketermer.

I ett av våre forsøk med tall-fraser viste det seg at bare ved å knytte ordet PARAGRAF til paragrafnummeret, fikk vi redusert antall funne irrelevante dokumenter med 14% uten å miste noen av de relevante.

I arbeidet med å utvikle en metode for automatisk gjenkjenning av denne type tall-fraser, kjørte vi ut en oversikt over alle tallforekomster i hele dokumentmaterialet. I følge denne oversikten forekom tall i ca. 90% av tilfellene enten rett foran eller rett bak ordet (eller ordene) som det var tilknyttet. Så vi bort fra alle ord som forekom sammen med tall 4 ganger eller mindre, sto vi igjen med en liten gruppe ord av typen ÅR, PARAGRAF, LEDD, PROSENT, JANUAR etc. Dette er alle ord som betraktet isolert sett - i allefall i vårt datamateriale - er typiske støyord, men som i sammenheng med et tall vil kunne utgjøre en viktig del av søkeargumentet. Forsøket viste med andre ord at det var mulig å gjenkjenne automatisk de mest vanlige tall-fraser av denne type med enkle metoder.

De fleste søketermer finner vi blant substantiv og i langt mindre grad blant verb. Dette henger kanskje sammen med at substantiv ofte beskriver ting og idéer, mens verb sier noe om handlingen.

Blant verbene finner man i første rekke hjelpeverb og uselvstendige verb som typiske støyord. Også verb som ANSE, UTTALE, ANTA, NEVNE etc. er å betrakte som støyord da de sier ingenting om hva som omtales, men på hvilken måte det omtales. Disse verbene vil i de fleste tilfeller forekomme såpass ofte, at de lett lar seg skille ut i en spredning- eller frekvens-ordliste.

Enkelte andre verb vil ha like stor evne til å karakterisere idéer som et substantiv. Dette er ord som ofte er avledet av et substantiv, og som forekommer i et langt mer begrenset omfang enn de øvrige.

Substantivene vil nesten alltid ha evnen til å karakterisere ting eller idéer, og de er derfor godt egnet som søketermer. I svært homogene dokumentsamlinger hender det imidlertid at man står ovenfor substantiv som er så typiske for temaet som behandles, at de ikke lenger kan bidra til å skille relevante dokumenter fra irrelevante. Eksempel på denne type substantiv er BARN og FARSKAP i en samling med bare farskapssaker, og SKATT i en samling med bare skatteavgjørelser. Disse ordene lar seg lett skille ut i en frekvensordliste, men man skal ikke alltid stole helt på frekvensen eller spredningen i bedømmelsen om en term er egnet som søketerm eller ikke. F.eks. i en samling med familierettsavgjørelser, hvor farskapssaker kanskje utgjør 80-90%, vil ord som BARN og FARSKAP spille en sentral rolle i karakteristikken av et farskapsproblem. De vil kunne bidra til å avgrense søkingen til nettopp denne delen av basen, og av den grunn heller bli tildelt større vekt enn de øvrige søketermer.

De fleste verb og substantiv vil kunne gjenkjennes på grunnlag av særegne suffikser. Dette er også forsøkt implemenert i vårt forsøkssystem, og resultatet vil vi komme tilbake til nedenfor.

Studiet av støyord i relasjon til ordklasser har pekt på enkelte sammenhenger som kan være nyttig ved automatisk identifisering av støyord. Nå er imidlertid disse sammenhengene ikke like entydig for alle ordklasser, og av den grunn har vi foreløbig valgt å ikke legge for stor vekt på ordklasser ved generering av støyordlister.

Informasjon om hvilken ordklasse et ord tilhører, vil i mange tilfeller kunne bestemmes ut fra suffiksen til ordet. Vi gjennomførte et forsøk, hvor vi ut fra en liste med preposisjoner, konjunksjoner, interjeksjoner etc. (jfr. ovenfor) og de mest karakteristiske suffikser for hver ordklasse, forsøkte å bestemme automatisk ordklassen til ordene. Resultatet viste en presisjon på ca. 90%.

Maskinell metode for identifisering av støyord

Ved utvikling av en maskinell metode for identifisering av støyord tok vi primært utgangspunkt i frekvensen og spredningen, og bare i enkelte tilfeller ordklassen. Det falt seg naturlig å betrakte deskriptorer - og ikke det enkelte ord - da ordenes form (bøyning) er uten interesse i denne sammenheng.

Med en deskriptor mener vi her en gruppe av ord som alle er avledet av samme grunnform. En deskriptor kan derfor gjerne omfatte både substantiv, verb, adjektiv etc. - f.eks. deskriptoren "ARV, ARVEN, ARVENE, ARVE, ARVER, ARVENE, ARVING". Grupperingen av deskriptorer foregikk maskinelt, og resultatet vil jeg komme tilbake til under avsnitt 5.

Deskriptorene ble tildelt en vekt som var beregnet på grunnlag av deskriptorens frekvens og spredning i dokumentsamlingen. Vekten ble beregnet ut fra følgende spredningsmål (jfr. Rosengren 1970):

$$K = n \left(\frac{\sum_{i=1}^{n} \sqrt{x_i}}{n} \right)^2$$

n: antall dokumenter
x_i: frekvensen til deskriptoren i dokument i

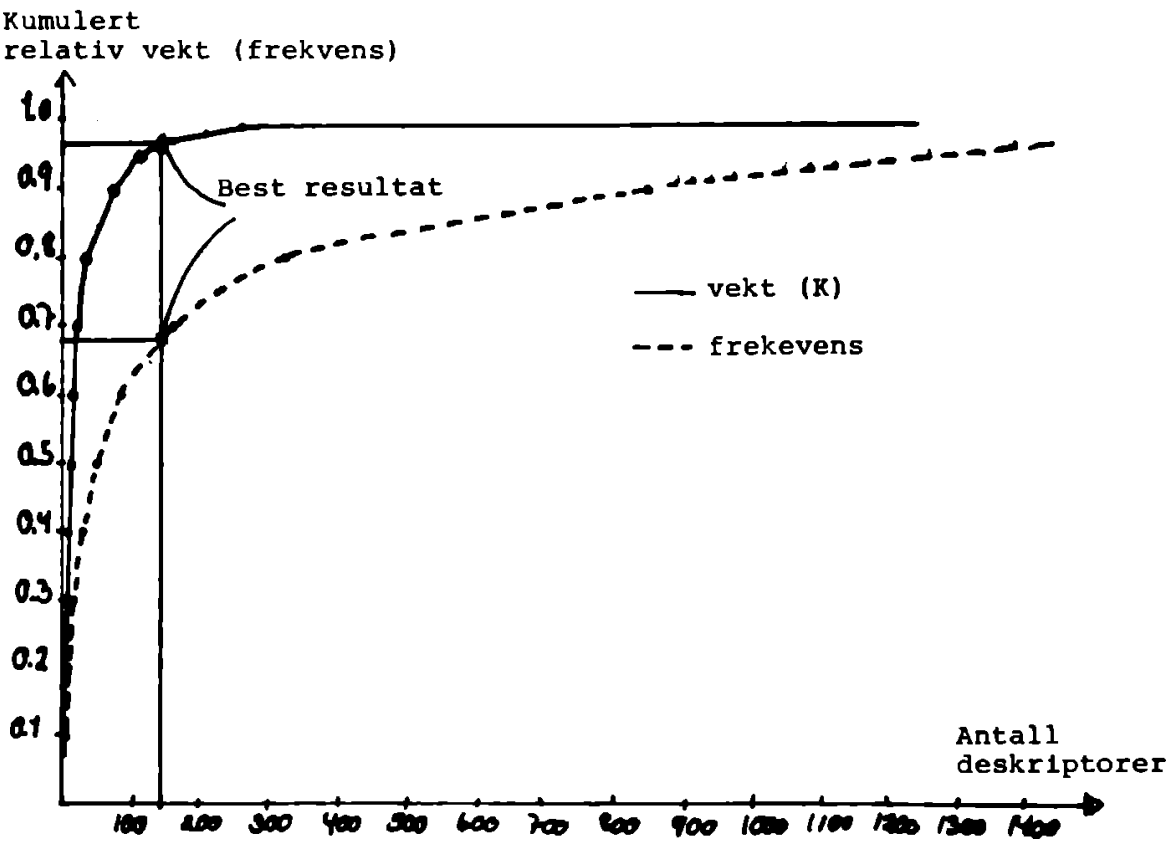
og øker med deskriptorens hyppighet og antall dokumenter den forekommer i. Vi forsøkte også å korrigere deskriptorens frekvens i et dokument (x_i) med lengden på dokumentet, men dette hadde liten innvirkning på resultatet i vårt tilfelle.

Vektene ble anvendt til å rangere deskriptorene, slik at de med høyst vekt kom øverst på listen. Denne listen viste seg på mange måter å gi et godt utgangspunkt for gjenkjenning av støyord - og langt bedre enn f.eks. en rangering basert på frekvensen alene.

For det første ga den en ganske bra rangering av deskriptorene i forhold til hvor godt de var egnet som søkeord. I figur 3 har vi gjengitt deler fra en slik liste basert på samlingen av tinglysningsavgjørelser. Øverst på listen finner vi de mest typiske støyord som preposisjoner, konjunksjoner, hjelpeverb, pronomen etc. Lenger nede framkommer tall, vanlige verb og substantiv som er svært karakteristiske for temaet i dokument-samlingen. Beveger vi oss langt nok ned, kommer vi til de mer karakteristiske ord.

For det andre fikk vi en meget gunstig fordeling av verdien på vektene med tanke på å kunne automatisk gjenkjenne hvor grensen går mellom støyord og ord som er bedre egnet som søkeord. Det viste seg nemlig at det var en relativt liten gruppe med deskriptorer som oppnådd en høy vekt, mens de fleste andre fikk en liten vekt. F.eks. for en av dokumentsamlingene utgjorde de 200 deskriptorer øverst på listen hele 98% av den totale summen av alle vektene (totalt ca. 11 200 deskriptorer). Målt i frekvens, utgjorde disse bare 73% av det totale antall termer i hele dokumentsamlingen. Dette kommer også til uttrykk i figuren nedenfor, hvor det nærmest er en knekk på kurven ved overgangen fra de hyppige deskriptorer til de mer skjeldene. Figuren viser antall deskriptorer i relasjon til den kumulerte relative vekten (avt. frekvensen).

Figur 2



Figur 3 Liste over deskriptorer sortert på vekten (K)
(basert på en samling med tinglysningsavgjørelser)

		VEKT
1.	I	5206220
2.	VÆRE, VAR, ER ..	5169546
3.	AV	4541176
4.	AT	44523941
5.	til	.
6.	OG	.
7.	PRG, PRG'S, ..	.
8.	SOM	.
	.	
	.	
	.	
	.	
14.	TINGLYSE, TINGLYSING, ..	.
15.	PÅ	.
16.	BLI, BLIR, ..	
	.	
	.	
24.	EIENDOM, EIENDOMMER, ..	914649
	.	
	.	
37.	GI, GIR, ..	.
	.	
43.	ANTA, ANTAR, ..	313116
44.	SIDE	.
45.	3	.
46.	BESTEMMELSE, ..	.
	.	
	.	
	.	
73.	TA	.
74.	BURDE, ..	.
75.	GJØRE, ..	.
	.	
	.	
	.	
131.	KREVE	.
	.	
	.	
	.	
199.	BOSETTE, ..	.
200.	BETALE, ..	13846

Det ble gjennomført forsøk med alternative støyordlister som omfattet et ulikt antall deskriptorer fra toppen av listen - f.eks. én støyordliste kunne omfatte de 80 øverste deskriptorene og én annen 100.

Det beste resultatet ble oppnådd ved å sette grensen akkurat i det området hvor kurven knekker, jfr. figur 2. Deskriptorene vi da fikk med utgjorde ca. 68% av det totale antall termer i hele dokumentsamlingen, og summen av vektene viste ca. 97% av den totale sum for alle vektene. Disse prosentandelene var omtrent de samme for alle våre 3 dokumentsamlinger (jfr. avsnitt 2).

Ved å eliminere tall og deskriptorer som bare omfattet substantiv, sto vi igjen med en nokså tilfredstillende støyordliste. I tillegg supplerte vi den med utelatte preposisjoner, konjunksjoner og pronomen. Også utelatte adverb, som lot seg gjenkjenne på grunnlag av suffiksen, ble lagt til listen.

Hele prosessen med å få identifisert støyord foregikk maskinelt, og det endelige resultatet viste at det var ytterst få ord man kunne reise tvil om hørte hjemme på en støyordliste. Sammenliknet med den manuelle støyordlisten omfattet den maskinelle langt færre ord. Allikevel viste søkeresultatene at bruk av den maskinelle listen ga vel så gode resultater som bruk av den manuelle.

5. Utvidelse av søkeargumentet

Termene i et spørsmål vil ofte være tilfeldig valgt og bestemt ut fra den formulering de er en del av. For å kunne komme så nær en fullstendig beskrivelse av meningsinnholdet i spørsmålet som mulig, burde hver av de utvalgte søketermer ha blitt supplert med synonymer. Dette forutsetter imidlertid informasjon om hvilke termer som er synonymer, og vi er her inne på et problem som ikke bare gjelder naturspråkbasert søkestrategier, men også tekstsøkesystemer generelt.

Enkelte systemer løser synonymproblemet ved hjelp av en synonymtesaurus, andre ved at brukeren selv definerer sine synonymer. Det arbeides i dag også med metoder basert på at systemet selv skal bygge opp et begrepsnettverk som gjenspeiler den semantiske strukturen i dokumentsamlingen (jfr. CONDOR-prosjektet, f.eks. Banerjee 1977).

Foreløbig har vi begrenset prosjektarbeidet til den enkleste form for synonymer, nemlig ord som er grammatikalske varianter av samme grunnform - dvs. deskriptorer. I tillegg har vi sett litt nærmere på effekten av å splitte sammensatte ord i spørsmålet. I neste fase ønsker vi å gå et skritt videre, å studere i hvilken grad vi kan få systemet selv til å generere (og vedlikeholde) en synonymstruktur, f.eks. ved å trekke på erfaringer fra tidligere søk og utnytte den respons det vil kunne få ved å føre informasjon tilbake til brukeren (f.eks. gjennom en dialog).

Gruppering av deskriptorer

Grupperingen av deskriptorer foregår maskinelt på grunnlag av en liste med

a) vanlige norske suffikser,
b) sterke verb,
c) uregelmessige bøyninger for substantiv,
d) preposisjoner, konjunksjoner, artikler og pronomen,
e) tall og tallord.

Til hver av disse elementene er det knyttet informasjon om hvilken ordklasse de tilhører. Listen er framkommet primært gjennom prøving og feiling med de tilgjengelige data. Til å begynne med, undersøkte vi muligheten for å anskaffe en slik liste fra miljøer som arbeider med språklig databehandling i Norge, men det eneste vi oppnådde var en liste med 67 suffikser basert på Ibsens verker. Listen var utviklet av NAVF's EDB-senter for humanistisk forskning, men var dessverre noe spesiell for vårt formål.

I dag omfatter listen ca. 1000 elementer. Den kan selvfølgelig gjøres enda bedre ved å supplere den med ytterligere elementer, med det viser seg at over et visst nivå vil elementene etterhvert bli så spesielle, at det skal mange til før men gjennomsnittlig seg merker positive utslag på søkeresultatene. Det vil derfor være et kostnad/nytte spørsmål hvor langt man skal gå i denne utvidelsen.

Deskriptoren til et ord dannes ved at man først finner fram til grunnformen på ordet, og deretter søker i den inverterte filen etter ord med samme grunnform.

Grunnformen til et ord genereres ved hjelp av listen ovenfor, og i følge våre forsøksresultater lykkes dette i gjennomsnittlig 90% av tilfellene. Det bør da tilføyes at suffikslisten også omfatter genitivs s, ing-form og lignende suffikser som ikke alltid er like lett å hanskes med. En stor del av feilprosenten kan defor føres tilbake til disse.

Den siste prosessen - søking etter ord med samme grunnform - kan unngås hvis man i den inverterte filen bare har grunnformen til ordene. Dette vil imidlertid føre til at man mister informasjonen om ordenes opprinnelige form, og dermed også muligheten til å søke etter disse i teksten (f.eks. som del av fraser). I dag arbeider vi med å finne metoder som ut fra grunnformen på ordet automatisk kan generere alle grammatikalske bøyninger av det uten ordbok. Dette byr ikke på vesentlige problemer innenfor én og samme ordklasse, men gjør det vanskelig når deskriptoren omfatter ord fra flere ordklasser.

Automatisk trunkering

De fleste tekstsøkesystemer i dag gir muligheten til trunkering av søketermer. Den mest vanlige form for trunkering er høyre-trunkering hvor brukeren kan søke på alle termer som begynner med en gitt tegnstreng. Tidligere forsøk med trunkering har

vist man gjennom høyretrunkering får med gjennomsnittlig 75% av alle kontekstuavhengige synonymer til de trunkerte ordene (jfr. Harvold 1974). Man vil i første rekke få med alle grammatikalske varianter av ordet, men også sammensatte ord og i enkelte tilfeller ord som er irrelevant for problemstillingen.

Vi fant det interessant å studere nærmere automatisk trunkering som et alternativ til utvidelse med bare deskriptorer. Foreløbig har vi bare sett på effekten av å trunkere stammen på ordet, og resultatet har gitt et noe blandet inntrykk.

Trunkeringen førte på den ene siden til at flere relevante dokumenter ble funnet og til at søkeordfrekvensen i de relevante dokumenter økte. På den annen side førte den også til at flere irrelevante ord ble fanget opp, og dette økte antall funne irrelevante dokumenter. De gjennomsnittlige søkeresultatene viste allikevel minst like gode som en utvidelse med bare deskriptorer. Man skal heller ikke se bort fra at metoden for automatisk trunkering kan gjøres enda bedre ved å ta hensyn til f.eks. ordklasser og uregelmessige bøyninger.

Splitting av sammensatte ord

På norsk er det nokså vanlig å anvende sammensatte ord, og i ett av våre spørsmålssett besto faktisk 38% av de ulike søketermene av sammensatte ord. Vi fant det derfor interessant å undersøke i hvilken grad en splittelse av disse ordene vil påvirke søkeresultatet.

Et sammensatt ord vil for det første bestå av flere ord, og en splittelse vil derfor kunne bidra til en bedre representasjon av meningsinnholdet i spørsmålet. For det andre vil det i mange tilfeller være en mindre sjanse for å finne et sammensatt ord enn de individuelle ordene - blant annet fordi at et sammensatt ord ofte kan skrives om til en frase.

På den annen side vil en splittelse også kunne ha en negativ effekt. Selv om ett av de individuelle ordene er til stede i dokumentet, er det slett ikke sikkert at idéen bak det sammensatte ordet er til stede. Dette kan føre til at flere irrelevante dokumenter blir funnet, men ved å prioritere de dokumenter som inneholder alle de individuelle ordene, vil denne effekten kunne reduseres.

Forsøket med splitting av sammensatte ord pågår fremdeles, og for tiden venter vi på å få tilgang til en rutine som gjennomfører splittingen automatisk uten ordbok. I følge de forsøk vi har gjennomført fra til nå, synes det som om en splittelse vil gi bedre søkeresultater.

6. Valg av regler for utvelging/rangering av dokumenter

I dette notatet vil jeg ikke gå nærmere inn på de forsøk som er gjennomført med rangering av de funne dokumenter, da dette faller utenfor temaet for denne konferansen. Allikvel vil jeg kort understreke hvor viktig denne delen av søkestrategien er.

Ut fra en antagelse om at ethvert dokument som inneholder minst én søketerm har en sannsynlighet for relevans, vil det alltid være en tendens til at for mange dokumenter blir funnet. Dette kommer spesielt til uttrykk i et tekstsøkesystem basert på argumenter i naturlig språk, på grunn av usikkerheten omkring valg av søketermer.

Formålet med en rangering av de funne dokumenter er derfor å få plassert dokumentene i en slik rekkefølge for brukeren, at de dokumenter som har størst sannsynlighet for relevans, blir plasert øverst på resultatlisten.

En viktig del av prosjektetarbeidet har derfor vært - og vil fortsatt være - å utvikle effektive metoder for rangering. Resultatene i dag viser at i ca. 64% av tilfellene får vi plassert relevante dokumenter øverst på resultatlisten, men fremdeles gjennstår mange uprøvde idéer - f.eks. å utnytte informasjon om ordklasser.

7. Oppsummering, konklusjon og framtidsutsikter

Hovedformålet med dette forskningsprosjektet er ikke primært å utvikle så effektive strategier for tekstsøking som mulig, men å undersøke muligheten for enkle og lite ressurskrevende søkestrategier basert på argumenter (spørsmål) i naturlig språk.

Prosjektet har i første rekke gitt oss en god innsikt i hvilke problemer som oppstår ved bruk av søkeargumenter i naturlig språk. Erfaringene viser - nokså naturlig - at det største problemet er overgangen fra et spørsmål i naturlig språk til et søkeargument som kan danne grunnlaget for søkingen. I denne forbindelse er det lagt mye arbeid i studiet av metoder for

- identifisering av søketermer og fraser i spørsmålet,
- utvidelse av søkeargumentet med termer som er avledet av søketermene spesifisert i spørsmålet (f.eks. ord med samme grunnform).

Det er grunn til å påpeke at metodene som er utviklet, er svært enkle og krever nesten ingen andre data enn de som vanligvis eksisterer i et tradisjonelt tekstsøkesystem. (Et unntak er listen med grammatikalske bøyningsregler og ord som preposisjoner, konjunksjoner, pronomen etc.)

En sentral del av prosjektarbeidet har også vært studiet av effektive metoder for rangering av de funne dokumenter. Fram til i dag har disse metodene bare tatt utgangspunkt i statistiske data, men vi ser ikke bort fra at de kan gjøres enda bedre ved trekke inn lingvistiske data.

Som en konklusjon på vårt arbeide vil vi si at forsøksresultatene har gitt oss tro på at det er mulig å oppnå er akseptabelt søkeresultat med så enkle søkestrategier som her beskrevet. Hva som er et akseptabelt søkeresultat vil imidlertid avhenge av den enkelte bruker og den aktuelle søkesituasjon.

Prosjektet vil bli avsluttet ved utgangen av dette året. Mens arbeidet så langt har vært rettet mot de metodiske spørsmål, vil vi i tiden framover konsentrere oss om spesifikasjonen av en modul for naturspråkbasert søking.

Det er fra neste år av søkt om økonomiske midler til et nytt prosjekt hvor vi ønsker å gå et skritt videre i arbeidet med tekstsøkesystemer basert på spørsmål i naturlig språk. Formålet med dette prosjektet vil være å utvikle en "intelligent" preprosessor (forsats) til et tekstsøkesystem, som gjennom en dialog med brukeren kan bidra til et rikere søkeargument enn det en typisk bruker vil kunne spesifisere uten hjelp. Prosjektet har fått navnet FORT (FORsats til Tekstsøkesystem) og er estimert til 3 år.

REFERANSER

Allén, Sture/Thavenius, Jan (1970)
Språklig databehandling; Studentlitteratur, Lund.

Banjerjee, N. (1967) "CONDOR - Communication in Natural language with dialogue oriented retrieval systems"; Schneider/Hein 1977: 163-172.

Bing, Jon/Fjeldvig, Tove/Flataker, Ole Bjørn/Harvold, Trygve (1980) Lovspråk og juristspråk;
Skriftserien Jus og EDB nr. 41, Oslo.

Fjeldvig, Tove (1976) Kontrollert forsøk i tekst-søking på uttalelser fra Skattedirektøren;
NORIS (8) III, Skriftserien Jus og EDB nr. 16, Oslo.

Harvold, Trygve (1976) "Belysning av synonymproblemet i norske, formuerettslige lover";
Bing/Fjeldvig/Flataker/Harvold 1980: 127-144.

Prestel, B.M. (1971) "Datenverarbeitung im Dienste juristischer Dokumentation", EDV und Recht, vol 3.

Rosengren, I. (1970) "Prosjektet Modernt tysk tidnings-språk"; Allén/-Thavenlus 1970:61-76.

Schneider, W./Hein, A.-I. Sågvall (1977)
Computational linguistics in Medicine; North Holland, Amsterdam.

Knut Kleve og Espen Smith Ore
Universitetet i Oslo

"PATTERN RECOGNITION" I PAPYRUSFORSKNING

Problemet er å rekonstruere bokstaver som det bare finnes blekkrester igjen av på papyrusunderlaget. Rekonstruksjon ved hjelp av fotografiske metoder har vist seg nytteløs. Der hvor det ikke er blekk, er skriften totalt forsvunnet, uten å etterlate seg spor. Rekonstruksjon ved hjelp av EDB-metoder gjenstår.

Et bokstavfragment kan sammenlignes med modeller av fullstendige bokstaver (jfr. fig. 1). Hvis de forskjellige bokstaver alltid hadde vært likt skrevet, ville saken være enkel. Opplysninger om de fullstendige modeller kunne lagres på en slik måte at de og fragmentene uten videre kunne sammenlignes på en datamaskin. Bokstaver er figurer som kan beskrives i en så finmasket todimensjonal matrise som det er ønskelig eller praktisk mulig å bruke.

Bokstavene i papyrene er imidlertid håndskrevne og varierer i størrelse, sentrering i bildet, helningsvinkel osv. Derfor er det ikke nok å bruke et program som bare ser etter om en figur (fragmentet) kan være en del av én av en gitt mengde større figurer (modellbokstavene). Man må statistisk avgjøre hva som er de mest typiske kjennetegn ved bokstavene i en papyrustekst og prøve å finne tilsvarende kjennetegn i det aktuelle bokstavfragmentet.

Slike statistiske gjennomsnittsbokstaver kan brukes til å sammenligne bokstavene som figurer. Det må da legges inn et slingringsmonn når det skal avgjøres om figurer er like (jfr. sammenligning av "fuzzy sets").

Det er mest sannsynlig at den endelige avgjørelse etter sammenligningen av fragment og modeller må tas av et menneske. Tegnene kan lett variere så meget at det ikke vil være mulig for maskinen å avgjøre hvilken bokstav fragmentet er en del av (jfr. fig 2). men man kan tenke seg et program som svarer med et utvalg av mulige bokstaver etter at fragmentet er sammenlignet med listen av modeller. Deretter kan man selv sammenligne fragmentet med de alternativer maskinen har gitt.

Sammenligningen kan foregå på følgende måte: Ved hjelp av en grafisk terminal legger man bilder av fragmentet og de forskjellige modellene over hverandre, og benytter even-

tuelt også et program som kan forstørre, forminske og vri bildene.

Input/output

Innlesningen av bokstavene og fragmentene kan foregå på forskjellige måter:

Man kan selv tegne et rutenett over et (forstørret) fotografi av tegnet.

Man kan la maskinen lese inn tegnet direkte fra et fotografi ved hjelp av optisk leseutstyr.

Man kan bruke et digitaliseringsbord hvor man tegner over et bilde av tegnet i ønsket størrelse med en magnet-penn som sørger for at bildet blir lagret med den nøyaktighet vi ønsker.

Utmatingen av (rekonstruerte) bokstaver kan skje grovt med en tegnmatrise på en linjeskriver, eller man kan få det tegnet på en plotter.

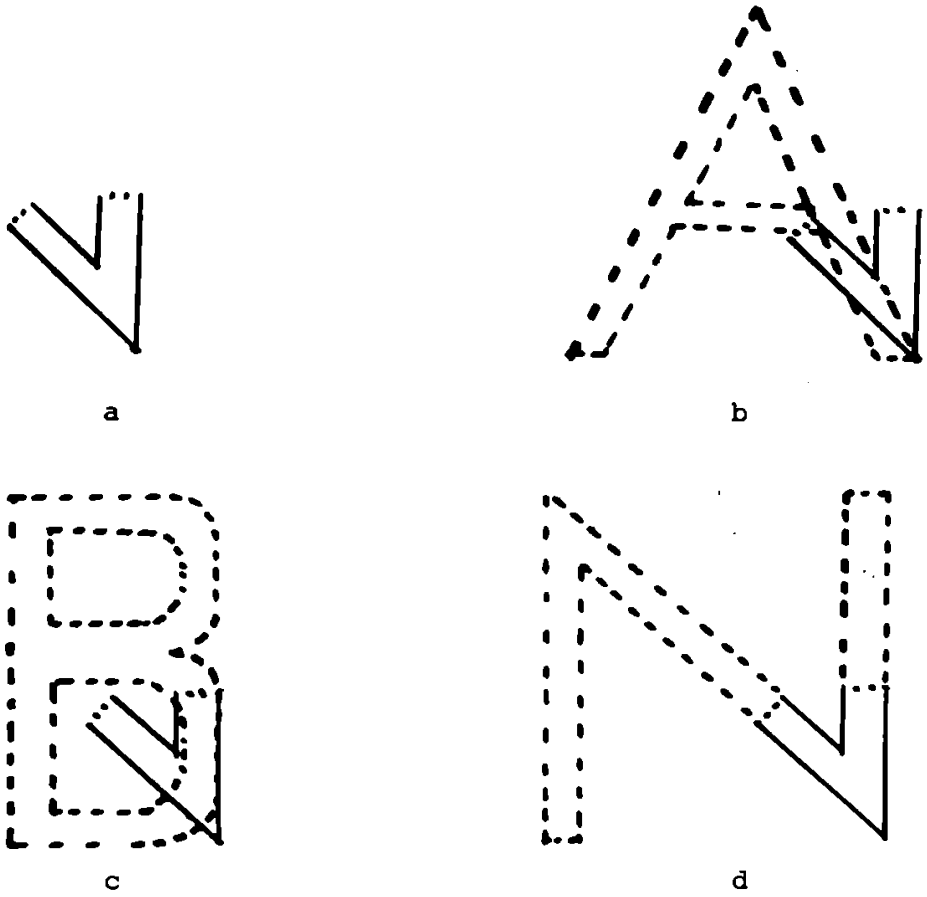
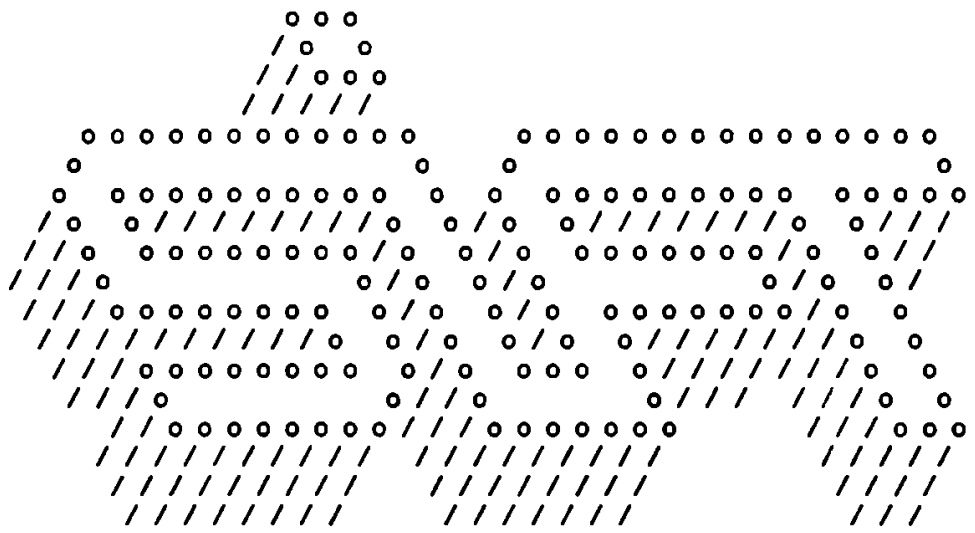


Fig. 1. a: fragmentet, b og c: sammenligning med modeller som ikke passer, d: fragmentet er en del av denne modellen.

Fig. 2. Her er det større likhet mellom modellbokstaven (prikket linje) A og bokstaven R enn mellom modell-A og skrevet A.

Øystein Reigem
NAVF's EDB-senter for humanistisk forskning



SIFT (Søking I Fri Tekst) - ET GENERELT INFORMASJONSSØKESYSTEM

Generelt om informasjonssøking og anvendelser.

Informasjonssøking eller information retrieval (IR) er en samlebetegnelse på forskjellige teknikker for gjenfinning av relevante undermengder av en større mengde informasjonsenheter. Disse enhetene kan være vitenskapelige artikler, sammendrag av artikler, brevene i en persons korrespondanse, opplysninger om museumsgjenstander, replikker fra et skuespill, lovtekster, domsavsigelser, overvåkingspolitiets personopplysninger m.m. Tradisjonelt forbinder man gjerne bruken av IR-systemer med søking i ustrukturert informasjon, dvs. vanlig tekst o.l. Det er imidlertid et klart behov for systemer som tilbyr søking i både ustrukturert og strukturert informasjon. (Med strukturert informasjon mener en faste datafelter som forfatter, navn på gjenstand, replikkinnehaver, dato for domsavsigelse, postadresse eller hårfarge.) Et godt IR-system trenger altså først funksjoner for å strukturere informasjonen systemet skal lagre, og dernest muligheter for å identifisere informasjonsenheter både ved ord i fritekstsammenheng og opplysninger i faste felter.

Bruken av IR-systemer spenner over svært store områder. Imidlertid går mange trekk ved anvendelsene igjen. Dette stiller bestemte krav til sentrale funksjoner og egenskaper ved generelle IR-systemer.

Store datamengder er karakteristisk for mange anvendelser. For å sikre en rask gjenfinning av informasjonsenheter - eller "dokumenter" som vi vil kalle dem - må systemet opprette og vedlikeholde ekstra datastruk-

turer. Den vanlige strategien består i å ha en såkalt invertert fil i tillegg til "dokumentfilen". Den inverterte filen inneholder en sortert liste over alle ord som forekommer i dokumentene, samt referanser til de enkelte forekomstene. Det går kvikt å slå opp i en sortert fil. Derfor går søking raskt, men det blir lett konflikter hvis en også ønsker en effektiv oppdatering. Dette ser en ved mange IR-systemer i dag.

Likevel er søkingen den kritiske faktoren i et IR-system. Siden IR-databaser pleier å være statiske, blir søking den hyppigste aktiviteten i systemet. Søking foregår gjerne vha. et søkespråk der man kan kombinere relevante ord med diverse operatorer. Typiske operatorer er logiske (og, eller, ikke), avstands- (søker på ord innen en viss avstand fra hverandre) og trunkeringsoperatorer (søker på starten / slutten av ord). Siden søking er så viktig og vanlig, bør søkespråket være kraftig, dvs. inneholde et variert reportoar av operatorer og funksjoner. Språket må også være lett å lære i sin enkleste form, slik at systemet kan nyttes av uerfarne brukere.

Den som en gang har brukt et IR-system, vet at gjenfinning av relevante dokumenter lett blir en prøve- og feileprosess. En forsøker seg gjerne fram med varianter av samme spørsmål, og en legger skjerpende betingelser på tidligere spørsmål. Også dette stiller krav til søkespråket, men når en skal kontrollere resultatet av hvert spørsmål, er det viktig å ha gode og brukervennlige funksjoner for "blading" i dokumentene. Spesielt i anvendelser der en har store dokumenter (større enn en dataskjerm, f.eks.), bør en ha muligheter for fokusering av avsnitt som inneholder søkebegrep eller visuell framheving av søkebegrep på skjermen / papiret.

Men uansett hvor sofistikert man lager et IR-system, vil det være spesielle anvendelser som ikke får alle sine behov dekket. Det er derfor viktig at systemet lett lar seg modifisere eller at det kan kommunisere med applikasjonsorienterte systemer. Dette krever en modulær oppbygning med veldefinerte grensesnitt utad og mellom delene. Dette er sjelden kost blant dagens IR-systemer. En slik oppbygning gir også muligheter for reduserte mikromaskinutgaver, eller oppdelte versjoner som kan kjøres på et nett av mikromaskiner.

Blant annet fordi en IR-database ofte har lengre levetid enn maskinutstyret den kjøres på, er portabilitet et viktig krav til et IR-system. (Et system sies å være portabelt dersom det ikke kreves mye arbeid å flytte det til en annen maskintype.) Også her lar eksisterende IR-systemer mye tilbake å ønske. Portabilitet fordrer disiplinert og gjennomtenkt programmering.

SIFT-prosjektet.

SIFT-prosjektet ble startet 1-1-80 og pågår under ledelse av Statens Rasjonaliseringsdirektorat. Prosjektet støttes av NTNF, og følgende institusjoner deltar:

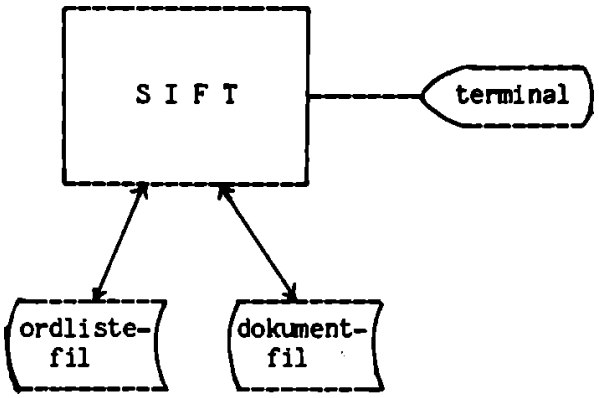
Norsk Data A/S
NAVF's EDB-senter for humanistisk forskning
LOVDATA
Institutt for privatrett, avdeling for EDB-spørsmål, UiO.

Målet med prosjektet er å utvikle et slagkraftig og fleksibelt IR-system som skal stilles til disposisjon for interesserte brukere. Under utviklingen bygger en på erfaringer med andre IR-systemer, spesielt den norske versjonen av det britiske STATUS-systemet, NOVA*STATUS. NOVA*STATUS har funnet anvendelse ved alle norske universiteter og i en rekke offentlige institusjoner.

SIFT-systemet.

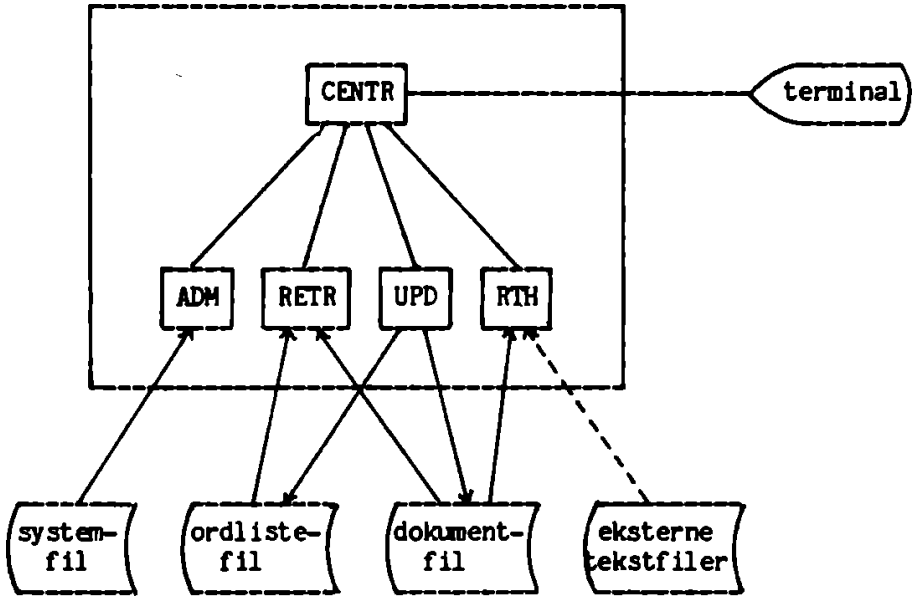
Basisversjonen av SIFT vil være et maskinuavhengig flerbrukersystem. Systemet styres av kommandoer som brukerne gir. (Det motsatte ville være at systemet til enhver tid presenterte valgmuligheter.) Kommandoene har posisjonsbestemte parametre, dvs. at rekkefølgen er viktig. (For eksempel er SLETT DOKUMENT 2-5 en riktig kommando, mens SLETT 2-5 DOKUMENT er gal.) Imidlertid er endel egenskaper lagt inn for å hjelpe uerfarne brukere. Systemet tilbyr en relativt omfattende HELP-funksjon (forklaringer på alle kommandoer og parametre), muligheten for å bli spurt etter en og en parameter i en dialog med systemet, og en omfattende bruk av "defaultverdier" (standardverdier) for de parametrene som brukeren ikke gir.

Slik kan man tenke seg SIFT (og for så vidt mange andre IR-systemer):



Dokumentfilen inneholder selve dokumentene, opplysninger om relasjoner mellom disse og om intern dokumentstruktur. Ordlistefilen inneholder en sortert liste over ord som forekommer i dokumentene, samt referanser til selve forekomstene. Søking foregår stort sett ved oppslag i ordlistefilen.

Her er en mer detaljert og omfattende figur:



I tillegg til den interne strukturen til SIFT, er det kommet flere filer med på figuren. Systemfilen inneholder mest mulig av systemparametre, slik at optimalisering av systemet for en bestemt anvendelse skal kunne foretas med få eller ingen inngrep i selve programmene. Her ligger også alle kommandonavn og meldinger slik at en lettvint kan generere versjoner for brukergrupper med annen språkbakgrunn. De eksterne tekstfilene er tatt med for å antyde at SIFT også skal kunne håndtere dokumenter på filer utenfor selve systemet. Det er et viktig poeng hvis man tenker på utvidelser eller integrering med andre systemer.

SIFT er fullstendig modulært oppbygd. "På toppen" sitter sentralmodulen (CENTR) og styrer de 4 andre hovedmodulene - administrasjonsmodulen (ADM), gjenfinningsmodulen (RETR), oppdateringsmodulen (UPD) og modulen som tar seg av søkeresultater (RTH). Hovedmodulene er oppbygd av undermoduler på samme hierarkiske måte som systemet selv, disse undermodulene av undermoduler igjen, osv.

All kommunikasjon med brukerne går gjennom sentralmodulen. Internt i systemet blir alle kommandoer omformet til "kommandopakker" som starter opp aktiviteter i en eller flere av de 5 hovedmodulene. Kommunikasjonen mellom sentralmodulen og de andre modulene går gjennom helt standardiserte grensesnitt. Derfor kan deler av systemet lett skiftes ut med eller modifiseres til applikasjonsorienterte versjoner. De ulike delene av systemet vil til og med kunne operere på forskjellige maskiner.

Av hovedmodulene er gjenfinnings- og oppdateringsmodulene de mest sentrale. Det er disse som leser og skriver på ordlistefilen. Denne filen er i SIFT organisert som et såkalt B-tre. Nodene i treet ("sidene" i ordlisten) er gitt en nokså sammensatt struktur. Denne organiseringen

løser langt på vei konflikten mellom effektiv søking og oppdatering, og gir dessuten mulighet for å optimalisere systemet med hensyn på en av de to prosessene.

Strukturer.

En samling sammenhørende dokumenter utgjør det vi vil kalle en database. Når et dokument innlemmes i en database, blir teksten organisert i felter, avsnitt, setninger, fraser og ord etter de kriterier brukeren har satt for den typen dokument. (En database kan inneholde flere typer dokumenter.) Referanser til lite meningsbærende ord kan sløyfes i ordlisten for å spare plass. Uønskede dokumentavsnitt kan utelukkes fra ordlisten og forstyrrer dermed ikke søkingen.

I en database kan brukeren gruppere dokumentene i mengder, og dokumenter kan defineres som historiske versjoner av hverandre. Søking kan foretas i flere databaser samtidig.

I tillegg til strukturering av databasene og de enkelte dokumentene, kan brukeren bygge opp tesauri til bruk i søkingen. Hun kan definere egne relasjoner og siden knytte sammen ord vha. disse relasjonene. Søkingen på en term kan så utvides til også å omfatte termene som står i bestemte relasjoner til denne. Tesaurusrelasjonene kan være av 4 typer: 1) Nettverksstrukturer, dvs. relasjoner av typen beslektede termer. 2) Trestrukturer, dvs. relasjoner av typen mer og mindre generell term. 3) Likeverdige, dvs. relasjoner av synonymtype. 4) Relasjoner mellom ord og forklaring på hvordan de brukes.

Søking.

Søkespråkets elementer er ord, maskete ord, fraser, operatorer og navn på felter og brukerdefinerte mengder av dokumenter.

En frase er en sekvens av ord definert som en sammenhørende enhet. Brukeren kan velge om de enkelte ordene i en frase skal være søkbare.

Masking av søkeord vil si at søkeordene inneholder spesielle tegn som står for mer eller mindre vilkårlige tegnstrenger. Eksempler: Dersom * er definert til å stå for en hvilken som helst tegnstreng, vil en søking etter *problem* finne alle dokumenter med ord som inneholder tegnstrengen problem, f.eks. problematikk, kommunikasjonsproblem, problemenes, problem osv. Dersom £ står for ett siffer, vil en søking etter 19££ finne alle dokumenter med firsifrede tall som begynner på 19.

Operatorene faller i 4 grupper: Logiske operatorer, avstandsoperatoren, operatorer som refererer til innholdet av navngitte felt og klasseoperatoren.

Blant de logiske finner vi de vanlige ELLER, OG, OGIKKE og IKKE. I tillegg kommer AVSN og IKKEAVSN, SETN og IKKESETN som er OG- og OGIKKE-operatorer innen samme avsnitt / setning. Eksempel: Spørsmålet usa SETN

nøytronbombe gir oss alle dokumenter der disse to ordene forekommer i samme setning.

Avstandsoperatoren setter grenser for hvor langt ord kan stå fra hverandre. Eksempler: Spørsmålet produ* /3/ nøytron* gir oss alle dokumenter hvor to ord som begynner på produ og nøytron står i høyst 3 ords avstand fra hverandre. Spørsmålet produ* nøytron* krever at ordene står etter hverandre med produ... først.

Innhold av felt kan gjøres til søkekriterium vha. operatorene LIK, MI, ML, ST, SL og MELLOM. Operatorene står for henholdsvis "lik", "mindre enn", "mindre eller lik", "større enn", "større eller lik" og "mellom". Sammenligningen foretas tegn for tegn i tekstfelt, og for tallet som helhet i tallfelt.

Ofte kan I-operatoren være bedre å bruke enn LIK. Dersom forfatter er et felt, vil spørsmålet asbjørnsen I forfatter gi alle dokumenter hvor ordet asbjørnsen forekommer i forfatterfeltet, selv om det skulle stå noe annet der også.

Dersom et spørsmål gir en viss mengde dokumenter som resultat, kan SAMME-operatoren brukes til å inkludere flere dokumenter i resultatet, nemlig alle dem med samme verdi i et bestemt felt, eller alle dem som ligger i samme brukerdefinerte mengder.

Klasseoperatoren kan gis eksplisitt eller den kan gis implisitt i klassekommandoen. I tillegg til vanlig søking, tilbyr nemlig SIFT såkalt klassesøking. Brukeren gir da et sett med spørsmål. Hvert spørsmål identifiserer en klasse. Dokumentene rangeres etter hvor mange klasser de tilhører, dvs. hvor mange spørsmål de tilfredsstiller. Brukeren kan få en mer eller mindre detaljert oversikt over klassefordelingen. Klassesøking er lett å bruke og har ofte vist seg å gi vel så gode resultat som kompliserte spørsmål basert på bare logiske operatorer.

SIFT kan vise brukeren alle ord eller termer som tilfredsstiller et bestemt uttrykk, f.eks. et masket ord eller en tesaurusrelasjon. Brukeren kan så på sin side bruke et utvalg av ordene som basis for nye spørsmål.

Brukeren kan lettvint referere til tidligere stilte spørsmål og også inkludere disse i nye. Spørsmål eller deler av spørsmål som går igjen kan brukeren definere som såkalte makroer. En makro gis et navn som den refereres ved. En makro kan ha argumenter, dvs. "åpne rom" for varierende deler. Brukeren kan bygge opp sine egne filer med makroer som kan hentes inn etter behov. En kan også ta vare på alle sine vanlige spørsmål på denne måten.

Resultathåndtering.

SIFT har et godt sett med kommandoer for å "bla" i dokumentene. Blading foregår både i mengder av dokumenter og innen lange dokumenter. Aktuelle mengder av dokumenter er: resultatet av et spørsmål, de historiske versjonene av et dokument, en brukerdefinert mengde eller databasen som

helhet. Det finnes en printkommando som gir utskrift på linjeskriver eller fil. Brukeren kan selv definere formater for visning og utskrift ved å spesifisere et utvalg av felter og avsnitt. "Highlight"- og fokuseringskommandoene framhever søkeordene og deres omgivelser, og gjør det dermed lettere å sjekke relevansen av resultatet.

Dokumentene i en database har en innbyrdes ordning bestemt ved generering/oppdatering. Resultatet av et spørsmål vil være en undermengde ordnet i samme rekkefølge. Men brukeren kan også få dokumentene rangert etter hyppighet av søkeord eller sortert på angitte felter.

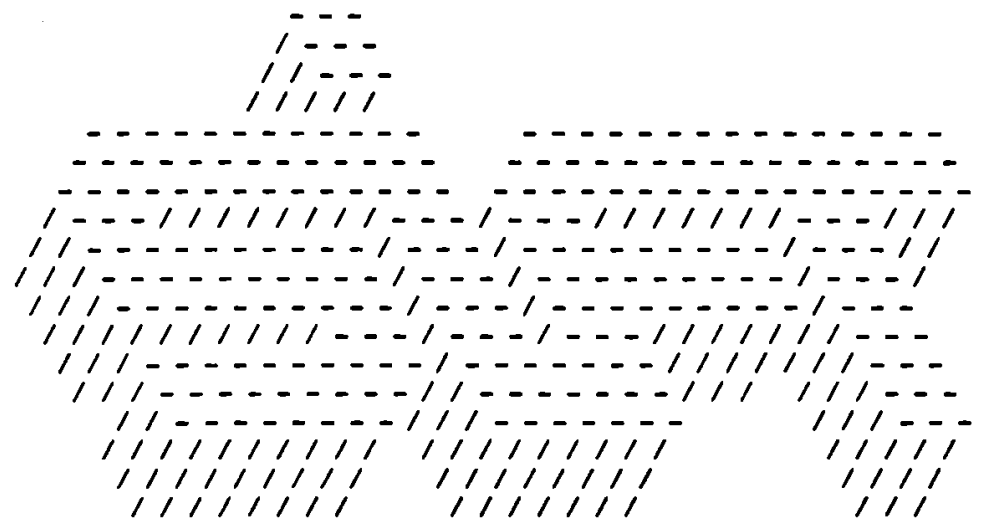
I tillegg til dette skal den endelige versjonen av SIFT inneholde ikke-interaktive, men mer avanserte sorterings- og utskriftsfunksjoner. Disse skal også omfatte frekvenslistinger på grunnlag av ordlistefilen.

Framdrift. Tilgjengelighet.

SIFT-prosjektet er delt i flere trinn. Det nåværende arbeidet går på en prototyp for NORD-maskin (SIFT-1). Prototypen planlegges ferdig innen utgangen av 1981. Andre trinn er utviklingen av en maskinuavhengig versjon (SIFT-PORTABLE). Disse første versjonene vil ikke inneholde alle egenskapene nevnt her. Tredje trinn består imidlertid i å utvide SIFT-PORTABLE til en fullstendig versjon (SIFT-COMPLETE). Siden kan en tenke seg ytterligere utvidelser som f.eks. skjermorientert kommandohåndtering, spørsmål i naturlig språk, lagring av dokumentrelasjoner vha. et ordinært databasesystem osv.

I samsvar med prosjektets målsetting vil all programvare og alle rapporter bli stilt til disposisjon for interesserte parter gratis.

Norsk Data A/S vil sannsynligvis markedsføre en SIFT-versjon spesielt tilpasset NORD.



Hans Olav Egede Larssen
Oslo

SØR ELLER SYD

- norsk "dobbeltform" med betydningsdifferensiering?

Norsk sprog har gjennom århundrer vært preget av påvirkning fra to kanter, dels den særnorske innflytelse fra dialektene og via nynorsk, dels påvirkninger fra et opprinnelig dansk og senere fornorsket skriftsprog, riksmål eller bokmål.

I dagens norske skriftsprog har dette ført til en rekke "dobbeltformer". Et eksempel på dette er formene SØR og SYD. SØR kommer fra dialektene og stammer opprinnelig fra det gammelnorske SUðR. SYD er kommet inn ved påvirkning fra det opprinnelig danske skriftsprog.

Siden 1938 har SØR vært eneste tillatte form innen bokmål. SYD har imidlertid levet videre i det "uoffisielle" riksmål. Og ved kongelig resolusjon av 30. april 1981 - det såkalte liberaliseringsvedtaket - er (bl a) SYD kommet inn i den offisielle rettskrivning som valgfri form ved siden av SØR.

Grunnen til dette er vel først og fremst at SYD er blitt mye brukt, dels som norm i aviser mm, men også av enkeltpersoner i skrift og tale på helt individuell basis.

En grunn til at formen har overlevet, kan også være at den har klar flertallsstøtte i Europas germanske og romanske sprog, herunder flere verdenssprog:
dansk SYD, tysk SÜD, fransk SUD, italiensk SUD, rumensk SUD, nederlandsk ZUIDEN, samt det beslektede enkelsk SOUTH. Svensk har SØDER, men SYDÖST/-OST og SYD-AMERIKA, spansk har SUR, men SUDESTE og SUDAMÉRICA (men AMÉRICA DEL SUR), portugisisk har SUL og SUL-AMERICANO (s.amerikaner), men også der SUDESTE.

Det mest interessante er kanskje likevel hva som har skjedd innen norsk, i sproglige miljøer med tilknytning til riksmål.

I 1962 utga en sprogets kunstner, dikteren og forfatteren André Bjerke, sin lille bok "Hva er godt riksmål". Der lanseres teorien om betydningsdifferensiering mellom SØR og SYD:

"Det er fjern-nær-bestemmelsen som avgjør det nyanserte valg av ordformen hvor det er tale om geografiske avstander. SØR står for det hjemlige, det innenlandske, mens SYD refererer seg til det fjernere, det fremmede, det mer omfattende..... Det logiske skille er klart: det heter SØR og SØROVER i landet, SYD og SYDOVER på jordkloden..... Altså: SØR-TRØNDELAG OG SØR-NORGE, men SYD-AMERIKA, SYD-AFRIKA og SYDPOLEN."

Også i offisielt bokmål har de sammensatte formene SYDEN, SYDFRUKTER, SYDLANDSK, SYDLENDING vært eneformer, altså med betydningsvariant 'fjern'. SYDPOLEN har vært tillatt ved siden av SØRPOLEN.

For å få verifisert Bjerkes teori har forfatteren selv undersøkt Norges største avis, Aftenposten. Den største undersøkelsen ble gjort i 1976, da alt redaksjonelt stoff med unntak av det mer spesielle olympia-stoff o.l. ble gjennomlest i månedene juli og august.

En rekke praktiske og økonomiske vanskeligheter gjorde det denne gang umulig å få lagret materiale til kjøring med EDB. Forfatteren kunne dermed heller ikke benytte sitt selvlagede Program SPROG for å finne og liste ut de enkelte forekomster av SØR/SYD. Dette var desto mer beklagelig fordi Program SPROG er særlig velegnet for ekstensiv utnyttelse av store mengder tekst-data.

Gjennomlesningen ga likevel en "nærkontakt" med stoffet som var særlig verdifull ved den første undersøkelse av et materiale så stort at statistisk sikre og tilstrekkelig nyanserte konklusjoner kunne trekkes.

I alt ble SØR brukt i 180 artikler i tidsrommet juli-august, mens SYD ble brukt i 784 artikler, en del ganger sammen med SØR.

Etter betydning ble inndelt i 3 hovedklasser. Kriterier var her:

1. Origo. Hvor ligger det som omtales i forhold til Norge eller skribentens norske lokalmiljø?

2. Radius. Hvor stor radius har det området som er i fokus? I "Moss ligger S for Oslo" er radius begrenset, i "Afrika ligger S for Norge" er radius, avstanden mellom de to steder, ganske betydelig.

Hovedregel - detaljer kan ikke gis her - var at betydningen ble kategorisert som H1 dersom origo var fjernt eller radius var stor, H3 dersom origo var innen Norge og radius lokal eller i hvert fall innen Norges grenser. En mellomkategori, H2, omfattet særlig kort radius (f eks tvers over gaten, vanlige avstander innen et urbant nærmiljø), samt tilfeller hvor en lokal/norsk avstand ble spesifisert uttrykkelig og hvor det da altså ikke var noe behov for avstandsmarkering. Klassifikasjonen begrunnes altså ut fra det inntrykk at formen SØR i riksmål kan brukes til å markere lokal radius i motsetning til lang eller uspesifiserbar radius - noe som er overflødig når radius er spesifisert på annen måte.

Hovedresultatet av undersøkelsen er:

Hovedklasse	Relativ andel SØR
H1 ('fjern')	8,2%
H2 (mellomkategori)	16,2%
H3 ('nær')	41,6%
I alt	18,7%

Formen SØR ble altså brukt 5 ganger så ofte når betydningen var H3 ('nær') som når betydningen var H1 ('fjern').

Resultatet er klart statistisk signifikant. Utsagnskraftig er endog at SØR ble brukt minst 4 ganger så ofte i betydning H3 ('nær').

Det er helt klart at en slik forskjell ikke kan skyldes tilfeldige "glipp" i et setteri pålagt å brukt den "konservative" form SYD etter avisens "private" normering.

Avisens medarbeidere var imidlertid pålagt å bruke formen SYD i navn på fremmede land. Men selv om alle slike navn kuttes ut av materialet, blir det fortsatt en riktignok svakere, men likevel klar og signifikant tendens i samme retning.

Inndeling ble foretatt i 8 hovedtyper etter bruksmåte av SØR/SYD ("spilltype"). Den generelle tendens ble gjenfunnet for de 5 typer for hvilke materialet var rimelig stort, for 3 av typene var det klar signifikans når alt stoff medregnes. For en av disse og 2 andre typer var materialet for lite eller ufullstendig.

Det obligatoriske SYD i navn på fremmede land (og deler av disse) sto i klar kontrast til bl.a. S-NORGE, hvor det var 65% SØR-NORGE mot 35% SYD-.

For h.typen "lokalisering/bevegelse i forhold til lokalt origo/lokal basislinje" var kontrasten mindre, med noe større andel SØR i 'fjern' bet., men likevel signifikant: "Landområder SYDØST og SYDVEST for Norge (H1, 'fjern') og "et nedlagt verksted like SYD for Åneby" (H2, spesifisert avstand) kontra "et nedlagt verksted SØR for Åneby", men også "SYD for Åneby" (H3, 'nær').

For "lokalisering innen et begrenset område" - herunder topografisk definerte områder som dalfører, elver mm. - var tendensen noe svakere pga relativt høy andel SØR også i bet. 'fjern'. For egenprodusert stoff + telegramstoff var forskjellen likevel signifikant.

For "global orientering" i forhold til det jordiske koordinatsystem dominerte SYD i 'fjern' bet.: "SYDPOLEN", "SYDLIGE breddegrader", "SYDLIGE halvkule"; "SYDLIGE stjernehimmel" (S. for himmelens ekvator), osv osv. (For bet. 'nær' var det bare en tvilsom observasjon av SØR).

For "Kringsjå/panorama" som en observatør ser det (f eks fra en fjelltopp) var materialet alt for lite.

H.typen "S.lige av en tellbar mengde enheter" synes med noe forbehold dominert av SYD, uansett 'fjern' eller 'nær': "Japans SYDLIGSTE øy", "den SYDØSTRE toppen" (på Hovedøya).

Inndeling etter stofftype ledet frem til en viktig konklusjon: Det var avisens egne medarbeidere som i det egenproduserte stoff praktiserte den mest gjennomførte betydningsdifferensiering. Telegramstoff hadde nesten samme tendens. I innsendt stoff var tendensene svakere, i innsendte artikler ble SØR brukt mer enn ellers der betydningen var 'fjern'; i "Lesernes mening" ol. så det derimot ut til at alt ble redigert til SYD. Værmeldinger hadde omtrent 50% SØR om vindretning - i kontrast til egenprodusert stoff hvor SYD dominerte i "avstandsnøytral betydning".

Bare 10 artikler i hele materialet inneholdt SØR/SYD brukt både i betydning H1 ('fjern') og H3 ('nær'). I 5 av disse var det forskjell i valg av ordform slik at SØR minst en gang er blitt brukt som "nærhetsmarkering". Dette er godt i samsvar med den generelle tendens i materialet og skulle antyde at i hvert fall en del av Aftenpostens medarbeidere har latt betydningsvariant få innflytelse på valg av ordform - mer eller mindre bevisst.

LITTERATUR

BJERKE, André: Hva er godt riksmål? Riksmålsforbundet, Oslo 1962, 2. utg. 1967.

LARSSEN, H.O. Egede: I serien STATISTISKE METODER I LINGVISTIKK:

SL 1 Statistiske metoder i lingvistikk. (Stensilert, 1969).

SL 2 Statistiske metoder i lingvistikk 2. (Stensilert, 1969).

SL 3 KJØD/KJØTT og statistikk. Del 1-4. Matem. inst., Univ. i Oslo. (Stensilert 1971-77).

SL 4 Ord, statistikk og betydningsforskjeller (OPPTAK/OPPTAGELSE). (Håndskrevet, 1972).

SL 5 Ordassosiasjoner og statistikk. (Håndskrevet utkast, 1972).

SL 6 Avistekst og urtekst. (Håndskrevet utkast, 1976).

SL 7 SØR og SYD. Forskjeller i bruksmåte og betydning. To undersøkelser av Aftenposten (1970-71 og 1976).

SL 8 En undersøkelse av enkelt- kontra dobbeltbestemt form i Aftenposten: PÅ DEN ANNEN SIDE kontra PÅ DEN ANDRE SIDEN. Matem. inst., Univ. i Oslo. (Stensilert, 1977).

LARSSEN, H.O. Egede: SL 9 FUNGERE eller FUNKSJONERE? En undersøkelse av Aftenposten, 1970 og -71. Matem. inst., Univ. i Oslo. (Stensilert, 1977).

SL 10 Statistiske analyser av betydningsdifferensiering mellom to ordformer. "The State of the Art". Oslo, 1980.

LARSSEN, H.O. Egede: Program SPROG. Computer-program for tekstutskrivning til semantisk-statistiske analyser. (Kopiert, 1970).

LARSSEN, H.O. Egede: Program SPROG. Utvidet versjon, de nye rutiner pr. 30.11.1970. (Kopiert).

Sture Allén
Språkdata
Göteborgs universitet

VAD ÄR DATALINGVISTIK?

Det som inte kunde bli av för min del i Köpenhamn kanske kan realiseras i Trondheim i stället: att inleda en diskussion om ämnesområdet. Nu kan jag också anknyta till min genomgång av frågan vid sommarens forskarkurs i Reykjavik. Här följer några förberedande synpunkter.

Efterhand som ämnet har vuxit fram under de senaste decennierna, har man haft lite olika tankar om dess innehåll. En del har velat ge det en mycket vid latitud, alltför vid enligt min mening. Det skulle då innefatta alla användningar av datamaskiner, vid vilka programmet utnyttjar någon lingvistisk princip. Andra har velat begränsa det till utveckling av lingvistiska algoritmer, vilket jag ser som alltför inskränkt. Det finns naturligtvis fler ståndpunkter.

För egen del har jag tidigt betonat att det rör sig om utveckling och tillämpning av algoritmiska metoder i syfte att vinna insikt i naturligt språk. Det är alltså fråga om ett språkvetenskapligt ämne av i princip allmän natur.

När det gäller att karakterisera området närmare, kan man ta sin utgångspunkt i begreppet språkbrukare. I det språkliga flödet gäller det för språkbrukaren att analysera och förstå språkyttringar som riktar sig till honom och att själv producera språkyttringar i förekommande fall.

Vilka resurser behöver språkbrukaren för att kunna utföra detta? Först och främst måste han ha lingvistisk förmåga. Det innebär att han känner till språkets lexikaliska och grammatiska uttrycksmedel och dess textuella grundprinciper.

Han måste också ha vad man kan kalla kognitiv förmåga. Det innebär bland annat att han kan organisera inkommande upplysningar och dra slutsatser ur dem. Det innebär exempelvis också att han kan planera de språkyttringar som skall produceras. Intressanta resultat i de här avseendena har kommit fram på området artificiell intelligens under senare år.

Vidare måste språkbrukaren ha en viss stilistisk förmåga. Det krävs att han har en uppfattning om vad som är vanligt och ovanligt i olika sammanhang, eftersom man exempelvis inte uttrycker sig likadant i ett privatbrev som i ett högtidstal.

Han måste också ofta kunna dra nytta av redan existerande texter eller textfragment för att göra sig gällande i språksituationen.

Till detta kommer att han måste ha kunskaper om världen, låt oss kalla det encyklopedisk förmåga. Två viktiga aspekter gäller huvuddragen i världens byggnad (riktningar, material osv.) och inträffade händelser i urval.

Man kan göra tankeexperimentet att placera ett programsystem på språkbrukarens plats i språksituationen. Det kommer då att kräva resurser av liknande slag som språkbrukaren utnyttjar för att kunna analysera och generera språkyttringar. Det är som jag ser det inom denna ram som datalingvistiken rör sig.

Språkvetenskapen förfogar idag inte över de redskap som krävs på något av de fyra områden jag nämnde, inte heller över de algoritmer som är en förutsättning för programsystemet. Modellen är i själva verket ett incitament till forskning på vida fält inom den angivna ramen. Det är högst ovisst om man någonsin når ända fram. De problem som skapas är emellertid av mycket stort intresse. Från språkvetenskaplig synpunkt knyter intresset sig i första hand till den lingvistiska och den stilistiska förmågan. Under överskådlig tid får man tänka sig system som strävar åt modellens håll som interaktiva.

En intressant aspekt är att datamaskinen kan användas för att bygga upp de resurser som programsystemet behöver. Detta kan ske på olika sätt. Man kan experimentera med exempelvis lingvistiska regelsystem för att utveckla dem. Man kan göra studier av enskilda verk eller stora textmängder för att få ett grepp om de faktiska språkförhållandena.

I grunden är det fråga om att språket betraktas som en process. Detta får konsekvenser på både det metodologiska och det teoretiska planet. En språklig teori framstår i detta ljus som en teori om analys och syntes av naturligt språk med hjälp av en uppsättning språkbrukarresurser. I själva verket befinner vi oss kanske i ett paradigmskifte, en övergång från en strukturorienterad till en processorienterad inriktning.

ALFABETISK FORFATTERREGISTER

DELTAKERLISTE

Allén, Sture	Språkdata	Göteborg
Amble, Tore	RUNIT	Trondheim
Andersen, Torben Arboe	Jysk Ordbog	Århus
Bilý, Milan		Lund
Brodda, Benny	Institutionen før Lingv.	Stockholm
Christensen, Rolf	Slavisk Institut	København
Eide, Bjørn	Nordisk institutt	Bergen
Evensen, Lars Sigfred	Inst. for anv. språkv.	Dragvoll
Faber, Dorrit		København
Falkedal, Kirsten		København
Findreng, Ådne	Germanistisk institutt	Dragvoll
Fintoft, Knut	Lingvistisk institutt	Dragvoll
Fjeldvig, Tove	Avd. for EDB-spørsmål	Oslo
Fonnes, Ivar	HF-data	Oslo
Gavare, Rolf	Språkdata	Göteborg
Golden, Anne	Nordisk institutt	Oslo
Grinstead, Eric	Centralinst.for Nord.Asienf.	København
Gustavson, Helmer	Riksantikvarieämbetet	Stockholm
Hagan, Holger	Inst. for språk og litt.	Tromsø
Hageberg, Arnbjørg	Norsk leksikografisk inst.	Oslo
Hagland, Jan R.	Nordisk institutt	Dragvoll
Hansen, Annemette Egerod		København
Hansson, Hasse	RECKU	København
Hauge, Jostein H.	NAVFs EDB-senter	Bergen
Heggstad, Kolbjørn	Nordisk institutt	Bergen
Hein, Anna Sågvall	Centr. før datorlingvistik	Uppsala
Hellvig, Lars	Arrheniuslaboratoriet	Stockholm
Hjorth, Ebba	Inst. for navneforskning	København
Hjulstad, Håvard	Norsk leksikografisk inst.	Oslo
Hofland, Knut	NAVFs EDB-senter	Bergen
Holmboe, Henrik	Inst. for lingvistik	Århus
Hovdenak, Marit	Norsk leksikografisk inst.	Oslo
Hvenekilde, Anne	Nordisk institutt	Oslo
Jahr, Mette-Cathrine	Britisk institutt	Oslo
Jensen, Klaus M.		København
Killingbergtrø, Laurits	Norsk leksikografisk inst.	Oslo
Koch, Gregers	DIKU	København
Købeke, Mikael		København
König, Magnhild Vollan	Nordisk institutt	Bergen
Landfald, Aagot	Norsk språkråd	Oslo

Larssen, Hans Olav Egede		Oslo
Lauvhjell, Arne	Norsk leksikografisk inst.	Oslo
Leistad, Geirr I.	Norsk senter for informatikk	Oslo
Leunbach, Gustav	Danmarks pædagogiske inst.	København
Lien, Eirik	Edb-tjenesten	Dragvoll
Linnarud, Moira		Lund
Ljunger, Magnus	Engelska institutionen	Stockholm
Lorentzen, Lise	Romansk institutt	Dragvoll
Lødrup, Helge	Nordisk institutt	Bergen
Löfström, Jonas	Språkdata	Göteborg
Mjaavatn, Per Egil	Nordisk institutt	Dragvoll
Mundt, Marina	Nordisk institutt	Bergen
Nielsen, Max	Datacenteret	Odense
Nordlie, Sigurd	Norsk leksikografisk inst.	Oslo
Olsen, Thorkil Damsgaard	Inst. for nordisk filologi	København
Olsen, Jørgen		København
Ongstad, Sigmund		Løvenstad
Pedersen, Birte Hjorth		Vanløse
Pedersen, Karen Margrethe	Inst. for dansk dialektf.	København
Rasmussen, Peter		København
Reigem, Øystein	NAVFs EDB-senter	Bergen
Rindal, Magnus	Nordisk institutt	Bergen
Ro, Anne Karin	Engelsk institutt	Dragvoll
Roldsgaard, Lars		København
Rosén, Valentina	Centr. før datorlingvistik	Uppsala
Ruus, Hanne	Inst. for nordisk filologi	København
Simensen, Erik	Leksikografisk institutt	Oslo
Skarsten, Roald	EDB-seksjonen v/HF	Bergen
Stålhane, Tor	RUNIT	Trondheim
Thorvaldsen, Gunnar	Reg.sentralen for his.data	Tromsø
Tietz, Susanne Boberg		København
Vestbøstad, Per	Norsk tekstarkiv	Bergen
Wedel-Jørgensen, H.C.		København
Weise, Lis		Søborg
Wikberg, Kay	Inst. for språk og litt.	Tromsø
Worren, Dagfinn	Norsk leksikografisk inst.	Oslo
Aass, Kristin	Norsk leksikografisk inst.	Oslo