KLAUS SCHUBERT

Kunskap om världen eller kunskap om texten?

En metod för korpusstödd maskinöversättning

Abstract

World Knowledge or Text Knowledge?
A Method for Corpus-based Machine Translation

As the scope and the quality demands on machine translation systems increase, the developers tend to direct their efforts not only on automating the translation process itself but also on automating the more labour-intensive subprocesses of the system development. This is the reason for a tendency towards more and more automated techniques of knowledge acquisition. Whereas commercial systems at present normally have dictionaries or knowledge banks which were generated in at best semi-automatic corpus-based or corpus-inspired ways, some of the more advanced research projects attempt to approach fully automatically generated knowledge banks. From this idea it is a logical step to using a corpus directly as a knowledge source for the machine translation process. For the DLT system of BSO in Utrecht a method is developed in which the translation process relies entirely on a Bilingual Knowledge Bank as its one and only source of translation-relevant knowledge. The Bilingual Knowledge Bank consists of parallel corpora of texts in a given source/target language and DLT's intermediate language Esperanto. The texts are represented as dependency trees with the sentence trees being linked up by text-grammatical pointers (e.g., for deixis, reference, event chains). Corresponding elements in the parallel versions of the text are combined to form translation units. The translation process is carried out by means of various sets of generalization rules which apply the specimen translation units to a given text to be translated. Such generalizations are made at the levels of monolingual syntax, metataxis (syntactic transfer) and semantics/pragmatics. Since the knowledge acquisition process is not carried out before the translation function requires a certain bit of information, it can be dynamically steered by the information contained in the part of the text already translated.

Proceedings of NODALIDA 1989, pages 218-228

1 Kunskapskällor för avancerad maskinöversättning

Att översätta är en krävande intellektuell handling. Försöken att bygga maskinöversättningssystem kan betraktas som ett strävande efter en automatisering av denna ytterst komplexa verksamhet. Ett par decenniers erfarenheter med denna strävan har visat att det inte räcker att automatisera bara själva översättningsprocessen. Redan i uppbyggningen av ett maskinöversättningssystem ingår så komplexa arbetssteg att även dessa måste utföras i stor utsträckning automatiskt eller åtminstone på ett avancerat datorstött sätt. Givetvis riktar sig detta sekundära automatiseringsintresse i första hand på de mest arbetsintensiva stegen i systemutvecklingsprocessen. Detta är för de flesta systemens vidkommande de lexikografiska (samt terminografiska) arbetsmomenten. Mera allmänt sagt gäller det i dessa moment att genom lexikografi eller på annat sätt inhämta den kunskap som behövs för översättningsprocessen och att göra kunskapen tillgänglig för datorsystemet. Jämför man de maskinöversättningssystem som har byggts eller projekterats sedan ett fyrtiotal år tillbaka, så kan man iaktta en utveckling i kunskapskällorna som står i direkt samband med nödvändigheten att automatisera själva kunskapsinhämtningsprocessen. Utvecklingen gäller både kunskapskällans innehåll och kunskapsinhämtningssättet. Tar man med även tilltänkta framtida innovationer så förlöper utvecklingen (något förenklat) så här:

Ordbok
(handgjord, baserad på intuition)
↓
Ordbok
(handgjord, korpusinspirerad)
↓
Ordbok/Kunskapsbank
(halvautomatisk korpusstödd framställningsprocess)
↓
Kunskapsbank
(helautomatisk korpusstödd framställningsprocess)
↓
Korpus i direkt bruk
(automatiska kunskapsinhämtningsmetoder)

De första två utvecklingsstegen var mycket vanliga i maskinöversättningens första decennier. Dagens system har för det mesta uppnått ett sådant omfång och sådana kvalitetskrav att rent handarbete har blivit ogörligt. Fullständigt handgjorda ordböcker förekommer däremot även i dag i maskinöversättningssystem som är relativt små, dvs system som antingen är avsedda enbart för experimentellt bruk eller som är inskränkta till en mycket snäv ämnesdomän.

Större system med friare textsort som är beräknade för praktiskt bruk har ofta en kunskapsinhämtningsmetod som motsvarar det tredje utvecklingssteget: Ett programsystem analyserar en korpus som vanligtvis redan är försedd med tillagd disambigueringsinformation och genererar ur korpusmaterialet lexikoningångar i maskinöversättningssystemets speciella format. Ingångarna granskas sedan av en människa och godtas eller korrigeras och kompletteras. I stället för (eller vid sidan om) en korpus av löpande text används ibland även vanliga ordböcker i bokform som görs tillgängliga för databehandling genom optisk inläsning eller konvertering av datafiler från sätt- och tryckmaskiner.

2 Automatisera systemutvecklingen?

Mig veterligen har det hittills inte marknadsförts något maskinöversättningssystem som bygger på en mera framskriden teknik än den jag här beskriver som det tredje steget. Däremot försöker man i somliga av de mest avancerade nu pågående forsknings- och utvecklingsprojekten att uppnå det fjärde eller till och med det femte steget. Resonemanget är enkelt: Granskningen av automatiskt genererade lexikoningångar är ett tidsödande och därmed dyrt arbete. Det ligger därför nära till hands att rikta automatiseringsintresset igen på det mest arbetsintensiva momentet och försöka att göra granskningsarbetet överflödigt. Om detta vore möjligt utan att ge avkall på översättningskvaliteten, skulle mycket vara vunnet.

Sådana automatiskt framställda lexikoningångar utgör det fjärde utvecklingssteget. Innan man satsar på detta, lönar det sig att föra tankeexperimentet vidare. Om det skulle visa sig vara möjligt att helautomatiskt generera lexikoningångar med utgångspunkt i en korpus, så betyder detta att den information man behöver för att kunna översätta finns i korpustexten och kan hämtas därifrån på ett helautomatiskt sätt. "Helautomatiskt" betyder i detta sammanhang framför allt att ingen kunskap behöver läggas till av människan. Om detta är så, då kan man eventuellt lika gärna låta bli att framställa ingångar och i stället anlita korpusen direkt som kunskapskälla. Detta är det femte steget i kunskapskällornas utveckling.

Tabellen ovan presenterar det femte steget som en fortsättning eller vidareutveckling av det fjärde, men med tanke på programsystemens storlek och snabbhet undrar man kanske om det inte snarare är ett steg tillbaka. Om det fjärde och det femte steget är likvärdiga, kan det då överhuvudtaget ha någon mening att diskutera det femte där man är tvungen att lagra en hel korpus i stället för några redundansfria ingångar? Är inte den kompaktare lösningen utan vidare att föredra? Jag beskriver nedan en lösning som siktar på det femte steget, så att det är på sin plats att skaffa sig klarhet om det precisa förhållandet mellan helautomatisk ingångsgenerering och direkt korpusbruk. Om det fjärde steget är möjligt, så betyder det att man kan generera för översättningsbehov tillräckliga lexikoningångar ur en korpus med hjälp av en på förhand fastställd uppsättning regler. Tankeexperimentet går ut på att man ur en given korpus får fram samma information, oavsett tidpunkten på vilken man tillämpar reglerna.

Det spelar alltså i detta avseende ingen roll om reglerna används innan eller medan det föreligger en konkret översättningsuppgift. Med andra ord, om man överhuvudtaget kan inhämta den nödvändiga informationen helautomatiskt, så har man friheten att välja om man vill förlägga inhämtningsprocessen till den förberedande systemutvecklingen eller till själva översättningsprocessen.

Det lönar sig inte att föra detta tankeexperiment vidare om inte det femte steget erbjuder väsentliga fördelar jämfört med det som är möjligt redan på det fjärde. Man måste ha mycket övertygande argument när man vill avstå från möjligheten att undangöra en så svår delprocess som korpusstödd kunskapsinhämtning onekligen är redan i utvecklingsfasen och i stället uppskjuta den till själva översättningsprocessen i runtime. Det enda giltiga kan vara ett argument som bygger på viktig tillagd information som blir tillgänglig först när översättningsprocessen har kommit igång. Bara om kunskapsinhämtningsprocessen kan styras eller avsevärt förbättras genom kunskap eller villkor hämtade ur den text som är under bearbetning, då kan det löna sig att tänka på en lösning på femte steget.

I den lösning jag skisserar i avsnitt 3 t o m 5 är kunskapskällan och den redan översatta delen av texten representerade i samma ytnära format. Bl a detta gör det möjligt att genomföra frekvensberäkningar, probabilistiska kontextjämförelser och liknande delprocesser på ett specifikt sätt som är anpassat till den konkreta kontexten och som dynamiskt tar med i beräkningen den kunskap som kan inhämtas ur den redan behandlade textdelen. På detta sätt blir den kunskapsbehandlingsprocess som stöder översättningsfunktionen i hög grad styrd av ett välavvägt samspel mellan den allmänna och den för tillfället mest relevanta speciella kunskapskällan.

Utöver fördelar som kan uppnås genom en kunskapsinhämtningsprocess i runtime finns det ytterligare en anledning att intressera sig för det femte steget. Denna anledning har i beskrivningen ovan någorlunda dolts av framställningssättet i den femstegiga utvecklingen. Jag har hittills bara diskuterat det femte steget under förutsättning att det fjärde är genomförbart, och jag har i tämligen allmänna ordalag talat om den information man kan inhämta med de två antydda metoderna: På det femte steget är denna information minst likvärdig med den man får på det fjärde, och det finns anledning att anta att det därutöver är möjligt att inhämta tillagd information som bara är tillgängligt på det femte steget. Detta resonemang får emellertid inte dölja den kvalitativa skillnad som ändå består mellan det fjärde och det femte steget. Skillnaden blir tydlig när man går närmare in på i vilken form informationen lagras. På fjärde steget utvärderas korpusen för att generera lexikoningångar. Processens utdata är alltså en fastlagd representation för den inhämtade kunskapen. Kunskapen representeras sålunda på ett explicit sätt. En lexikoningång skall vara tillämplig på vilka som helst förekomster av uppslagsordet. (I stället för ett uppslagsord kan det givetvis vara fråga om en annan enhet, t ex ett morfem, ett syntagm osv.) När man genererar lexikoningångar, är det meningen att uppnå en så allmängiltig och täckande beskrivning av uppslagsordet som möjligt (eller några få sådana). På femte steget däremot används korpusen direkt som kunskapskälla, och en korpus är av en kvalitativt annorlunda karaktär än en lexikoningång. Medan en lexikon-

ingång skall vara allmängiltig i den mån detta är möjligt, innehåller en korpus enbart exempel. Medan alltså ett system av fjärde steget går från exemplen i den underliggande korpusen genom härledningsregler till en i denna speciella bemärkelse allmängiltig lexikoningång och därifrån genom tillämpningsregler till den konkreta översättningsuppgiften, så läggs på femte steget ett omedelbart förband mellan exemplen och uppgiften. Det allmängiltiga mellansteget kan falla bort.

Detta är en väsentlig iakttagelse. För att bevisa tillämpligheten av det femte steget behöver man alltså inte förutsätta att det fjärde är möjligt. Det räcker att bevisa att man ur korpusexempel direkt kan härleda den information som behövs för översättningsprocessen.

3 DLT:s tvåspråkiga kunskapsbank

För maskinöversättningssystemet DLT projekteras numera en korpusstödd kunskapsbehandlingsmetod som strävar efter att närma sig skalans femte steg.

Innan jag tar upp metoden något mera i detalj kan ett par inledande ord över DLT vara nödvändiga. *Distributed Language Translation* (DLT) är namnet på ett maskinöversättningsprojekt som bedrivs av det nederländska mjukvaruföretaget Buro voor Systeemontwikkeling (BSO/Research) i Utrecht, delvis med statligt anslag. Efter en förstudie (Witkam 1983) inträdde DLT år 1985 i implementeringsfasen. Den första prototypen blev färdig 1987, den andra 1988. DLT skall bli ett mångspråkigt system, bl a för tillämpningar i datakommunikationsnät. Under utgångsspråksanalysen förs en systeminitierad disambigueringsdialog med användaren. Dialogfrågorna ställs på utgångsspråket och det krävs ingen posteditering, så att användaren inte behöver känna till målspråken. Förbindelselänken mellan utgångs- och målspråken är mellanspråket esperanto.

De första prototypversionerna översätter från engelska genom esperanto till franska. Som kunskapskällor anlitar de tre morfosyntaktiska ordböcker (engelska, esperanto, franska), två tvåspråkiga metataxordböcker (engelska-esperanto, esperanto-franska; om termen *metatax* jfr Schubert 1987) och en enspråkig lexikal kunskapsbank (esperanto). De olika framställningsprocesserna låg mellan det första och det tredje steget på skalan (jfr om prototypens arkitektur: Schubert 1986; om kunskapsbanken: Papegaaij 1986).

Utvärderingen av erfarenheterna med prototyperna har lett fram till en vidareutveckling av kunskapskällorna som betyder en ingripande förändring i systemet DLT:s sätt att fungera. DLT är (redan i prototypversionerna) ett modulärt system. Det består av språkparsmoduler som alltid har esperanto på den ena sidan. En text som översätts till ett enda målspråk passerar på så sätt två sådana språkparsmoduler. Det är bl a på grund av denna arkitektur som systemet kan betraktas som ett dubbelt direkt översättningssystem (Schubert 1988). I DLT:s tredje systemversion, som befinner sig i planerings- och modellimplementeringsfasen, har alla kunskapskällor som ingår i samma språkparsmodul sammanfattats till ett enda system. Detta system har fått namnet *Tvåspråkig kunskapsbank.*

DLT:s Tvåspråkiga kunskapsbank består av en parallell korpus, dvs en och samma text parallellt på två språk (original och översättning, även två översättningar från ett tredje språk). I det pågående provimplementeringsarbetet ingår paren engelska-esperanto och franska-esperanto. I den slutgiltiga implementeringen av den tredje systemversionen skall minst två språk komma till; senare versioner projekteras för två paket av sex språk var, varefter flera språk kan läggas till efter behov tack vare DLT:s modulära mellanspråksarkitektur. Jag illustrerar den Tvåspråkiga kunskapsbanken nedan med språkparet danska-esperanto.

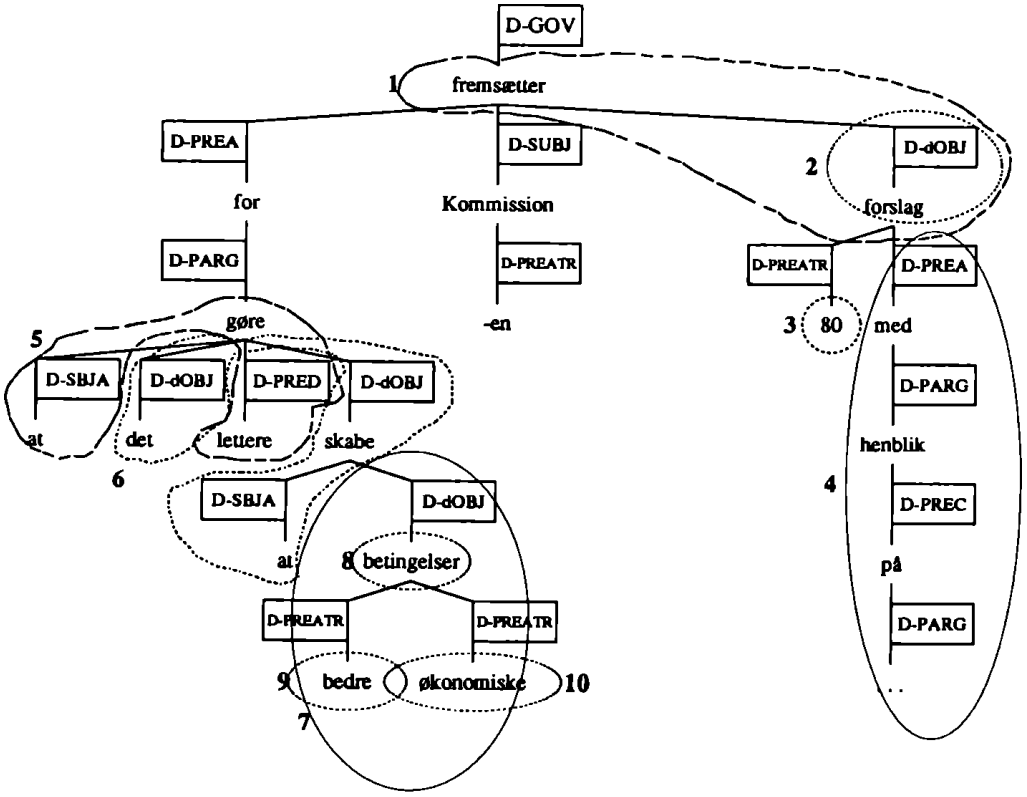
Korpustexterna lagras i den Tvåspråkiga kunskapsbanken i disambiguerad form. Den grundläggande representationsformen är dependenssyntaktiska träddiagram, utdata av en parser (jfr Schubert 1987: 28–129). Dessa är syntaktiskt oambiguösa. Textgrammatiska pekare (deixis, referens, skeendekedjor m m) förbinder satserna och meningarna till sammanhängande texter. Vid sidan om dessa enspråkiga markörer är texterna försedda med speciella tvåspråkiga pekare som bygger upp översättningsenheter. En översättningsenhet är ett ord, en ordgrupp eller bara ett morfem med dess motsvarighet i det andra språket. Markörerna för de syntaktiska relationerna som är utsatta i dependensträdet ingår även i översättningsenheten. En större översättningsenhet kan innehålla mindre enheter. Enheterna är dock inte mindre än att den översättningsmotsvarighet de innehåller kan användas även i andra kontexter.

4 En illustration

Som illustration använder jag följande mening på danska och esperanto. Träddiagrammen bygger på dependenssyntaxerna som har utarbetats enligt DLT-modell för danska (Ingrid Schubert 1989) och esperanto (Schubert 1989) där de här använda etiketterna för dependensrelationer förklaras i detalj. Av utrymmesskäl tar jag bara en enda mening och visar bara den övre delen av träddiagrammen. Trädavsnitt med samma nummer i båda träden utgör översättningsenheter. För översiktlighetens skull markerar jag långt ifrån alla översättningsenheterna. Textgrammatiska pekare utelämnas helt.

> For at gøre det lettere at skabe bedre økonomiske betingelser, fremsætter Kommissionen 80 forslag med henblik på at nedbryde markedsskrankerne og sætte virksomhederne i stand til fuldt ud at drage fordel af den europæiske dimension.
>
> Por faciligi la kreadon de pli bonaj ekonomiaj kondiĉoj la Komisiono faras 80 proponojn, kiuj celas faligi la barilojn de la merkato kaj ebligi al la entreprenoj maksimume eluzi la avantaĝojn de la eŭropa dimensio.

(Med hänsyn till entydighet i ordstrukturen markeras morfemgränserna i DLT:s mellanspråk som är fullständigt agglutinerande. Morfemtecknen ses i esperantoträdet med har utelämnats här.)



Figur 1:

I denna mening förekommer både enkla och komplexa översättningsenheter. Till de enklare hör ettordsenheterna:

[10] økonomiske — ekonomiaj

Översättningsenheten

[2] forslag — proponojn

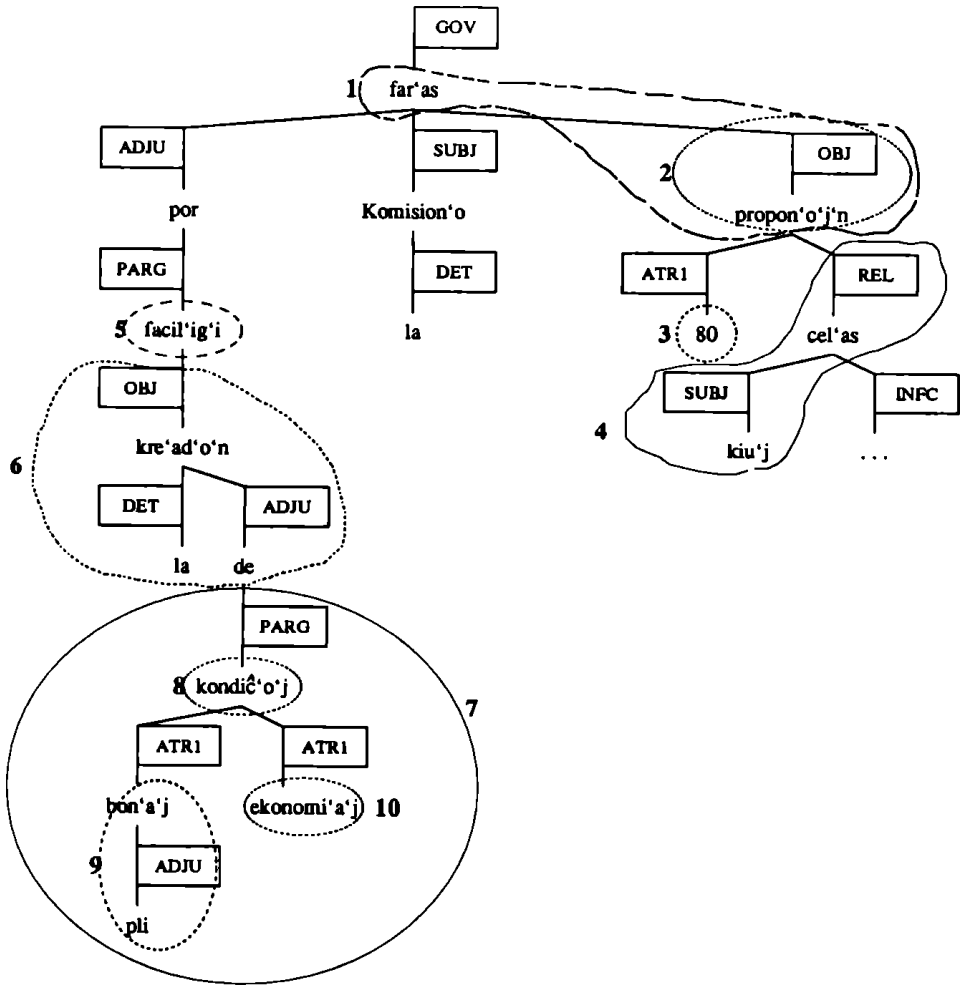
innehåller på esperantosidan en ackusativ (*-n*), så att en giltig motsvarighet bara föreligger när objektetiketten tas med i enheten.

Observera att *forslag* motsvarar *proponojn* men att *fremsætter* inte motsvarar *faras*. Verbet *fari* betyder 'göra' och kan inte betraktas som brukbar översättning av *fremsætte* i andra kontext. Därför omfattar enheten flera ord:

[1] fremsætter forslag — faras proponojn

En mera komplex motsvarighet är

[5] at gøre lettere -- faciligi



Figur 2:

Även mera ingripande syntaktiska förskjutningar kan iakttas i denna mening:

[4] med henblik på — kiuj celas

Detta är en övergång från ett prepositionellt tilläggsled (D-PREA) i danska till en relativ bisats (REL) i esperanto (*kiuj* 'vilka', *celas* 'avser').

Översättningsenheten

[6] det at skabe — la kreadon de

illustrerar bra att korpusinformationen har exempelkaraktär. Givetvis översätter man inte alltid infinitivkonstruktionen *det at skabe* med en substantiv (*la kreado* 'skapandet'), men formen kan mycket väl tänkas förekomma i vissa andra kontexter. Informationen är alltså inte allmängiltig, men illustrerar bara en av många möjliga översättningar för detta syntagm.

5 Översättarens kunnande i maskinöversättningssystemet

Det esperanto-danska meningsparet är godtyckligt valt. Att illustrationen utöver enkla fall som [10] eller [3] även innehåller så många strukturellt och lexikalt intressanta motsvarigheter är ett tecken på att en korpusstödd översättningsmetod har tillgång till en mängd konstruktioner och motsvarigheter som i denna form aldrig upptas i vanliga ordböcker. Här återspeglas i implicit form översättarens kunnande.

I DLT:s kunskapsbank ingår en stor mängd exempel på hur vissa textbitar i konkreta tillfällen har översatts av fackkunniga översättare. "Uppfinnaren" av den (patentanmälda) Tvåspråkiga kunskapsbanken, Victor Sadler, beskriver i detalj hur man kan översätta med hjälp av denna kunskap (Sadler 1989). Hans framställning, som inte kan återges här, vill jag i korthet komplettera med två aspekter som kan belysa denna korpusstödda översättningsmetod med utgångspunkt i resonemanget om den femstegiga skalan i maskinöversättningssystemens utveckling. Den första aspekten beträffar uppbyggningen av en Tvåspråkig kunskapsbank och den andra tillämpningsreglerna.

I diskussionen om DLT:s förnyade systemarkitektur bör två funktioner hållas klart åtskilda: å ena sidan kunskapsbehandlingen under översättningsprocessen i runtime och å andra sidan kunskapsinhämtningen. Den senare processen, i vilken uppbyggningen av den Tvåspråkiga kunskapsbanken ingår, är förlagd till DLT-"fabriken", medan själva översättningsprocessen givetvis äger rum hos användaren. Även om det naturligtvis är önskvärt att automatisera kunskapsinhämtningsprocessen i hög grad, är det dock inte uteslutet att utföra manuella ingripanden och att anlita specialisthjälp som inte är tillgänglig under översättningsprocessen. På ett liknande sätt kan inhämtningen också utnyttja större och annorlunda datorsystem än användarmodulerna. Under översättningsprocessen är den enda mänskliga hjälp systemet kan få de svar som ges i den interaktiva disambigueringsdialogen. Dialogfrågorna ställs på utgångsspråket och måste kunna besvaras av språk- och datavetenskapliga lekmän. Denna skillnad förklarar hur det är möjligt att den Tvåspråkiga kunskapsbanken i systemutvecklingsfasen förses med all den utomspråkliga information jag ovan har nämnt: oambiguösa syntaktiska träd, textgrammatiska pekare, översättningsmotsvarigheter m m. I DLT-fabriken genereras dessa strukturer automatiskt, t ex genom parsning, i den mån detta är möjligt och sedan granskas och kompletteras de av människor. Mänskligt arbete behövs framför allt för identifieringen av översättningsmotsvarigheterna. Medan DLT-användaren kan vara en enspråkig person som bara förstår utgångstexten, kräver kunskapsinhämtningsprocessen specialistarbete, bl a av yrkesöversättare. Den Tvåspråkiga kunskapsbanken är alltså inte inskränkt till den kvalitet i betydelseanalysen som i dagens läge kan uppnås med helautomatiska kunskapsbehandlingsprocesser, utan den har tillgång till mänsklig fackkunskap.

Den andra aspekt som jag vill nämna här gäller generaliseringsfunktionen. Som ett speciellt slags korpus innehåller den Tvåspråkiga kunskapsbanken ex-

empel på faktiskt bruk av ord, uttryck och översättningsmotsvarigheter. Men varje korpus är nödvändigtvis "för liten", dvs en exempelsamling, hur stor den än blir, kan aldrig innehålla varje användningsmöjlighet av varje enstaka ord och varje enstaka konstruktion (jfr Lehrberger/Bourbeau 1988: 129). Statistiken visar att ungefär hälften av alla lemman i en korpus förekommer bara en enda gång i löptexten. De regler med vars hjälp korpusinformationen tillämpas på konkreta översättningsuppgifter (parsning, syntaktisk transfer, lexikal transfer, semantisk-pragmatisk disambiguering osv) bör därför generalisera med utgångspunkt i korpusexemplen. Sådana generaliseringsfunktioner behövs på minst tre plan:

1. enspråkig syntax (parsning [jfr Zuijlen 1989], morfosyntaktisk syntes);

2. metatax (syntaktisk transfer);

3. semantik-pragmatik (mätning av betydelseavstånd m m).

I överensstämmelse med DLT:s arkitekturprinciper följer dessa generaliseringsregler implicitetsidén, vilket bl a innebär att de undviker att anlita en explicit allmängiltig mellanrepresentation som den som skulle behövas i ett system på skalans fjärde steg.

I tankeexperimentet ovan antog jag att en väsentlig drivkraft för att låta utvecklingen gå vidare från det tredje till det fjärde och femte steget är intresset i att ersätta det mänskliga arbete som är nödvändigt i kunskapskoderingsprocessen med automatiska processer. Den lösning jag skisserar ovan realiserar denna vidareutveckling, men den saknar ändå inte arbetsmoment där information läggs till av människor. Människan, systemutvecklaren, har alltså inte rationaliserats bort. Men DLT:s korpusstödda översättningsmetod innebär att det bidrag specialisterna lämnar till kunskapsinhämtningen motsvarar i mycket högre grad än vid tredje utvecklingssteget ett vanligt sätt att resonera över språk. Ber man en specialist att ge exempel på språkbruk i sitt ämnesområde eller att granska föreslagna formuleringar så är uppgiften enklare och svaren pålitligare än när man är tvungen att be om en allmängiltig metaspråklig redogörelse. De som medarbetar i DLT:s systemutvecklingsfas kan därför i högre utsträckning vara specialiserade i ämnesområdet och i översättning och behöver i mindre grad koncentrera sig på teoretisk grammatik eller lexikografi.

6 Kunskap om världen eller kunskap om texten?

Sålunda tillåter utvecklingen av maskinöversättningssystemet DLT med sin nya systemstruktur ett nyartat svar på frågan huruvida grundvalet för maskinöversättningsändamålet bör vara (utomspråklig) kunskap om världen eller (inomspråklig) kunskap om texten. DLT:s svar är att det är kunskap ur texter.

Litteratur

Lehrberger, John, Laurent Bourbeau. 1988. *Machine translation.* Amsterdam/Philadelphia, Benjamins.

Papegaaij, B. C. 1986. *Word expert semantics.* Dordrecht/Riverton, Foris.

Sadler, Victor. 1989. *Working with analogical semantics. Disambiguation techniques in DLT.* Dordrecht/Providence, Foris.

Schubert, Ingrid 1989. A dependency syntax of Danish. Dan Maxwell, Klaus Schubert [utg.]. *Metataxis in practice. Dependency syntax for multilingual machine translation*:39–67. Dordrecht/Providence, Foris.

Schubert, Klaus. 1986. Linguistic and extra-linguistic knowledge. *Computers and translation*, 1:125–152.

Schubert, Klaus. 1987. *Metataxis. Contrastive dependency syntax for machine translation.* Dordrecht/Providence, Foris.

Schubert, Klaus. 1988. The architecture of DLT—interlingual or double direct? Dan Maxwell, Klaus Schubert, Toon Witkam [utg.]. *New directions in machine translation*:131–144. Dordrecht/Providence, Foris.

Schubert, Klaus 1989. A dependency syntax of Esperanto. Dan Maxwell, Klaus Schubert [utg.]. *Metataxis in practice. Dependency syntax for multilingual machine translation*:207–232. Dordrecht/Providence, Foris.

Witkam, A. P. M. 1983. *Distributed Language Translation.* Utrecht, BSO.

Zuijlen, Job M. van 1989. Probabilistic methods in dependency grammar parsing. *International Workshop on Parsing Technologies*:142–151. Pittsburgh, Carnegie-Mellon University.

Klaus Schubert
BSO/Research
Postbus 8348
NL-3503 RH Utrecht
Nederländerna
schubert@dlt1.uucp