

När kan man lita på maskinöversättning?

Aarne Ranta

Institutionen för data- och informationsteknik
Chalmers tekniska högskola och Göteborgs universitet

aarne@chalmers.se

Sammanfattning

Artikeln ger en introduktion till grunderna i automatisk översättning. Den hjälper läsaren att förstå vilka fel som är typiska och varför, och hur man ska använda t.ex. Google Translate så att man får ut bästa möjliga resultat.

Översättning med dator

Dagens system för automatisk översättning är opålitliga. Tidigare översattes t.ex. ”jag är svensk” till ”ich bin ein Amerikaner” (jag är en amerikan) på tyska i Google Translate. Just det här felet är borta nu, men det finns mycket annat som fortfarande blir fel. Syftet med den här uppsatsen är att ge en förståelse för hur fel som detta uppstår. Vi ska ta en titt på de grundläggande teknikerna och fråga oss vad som är svårt och vad som är lätt i maskinöversättning.

Trots sådana översättningsproblem är det fler och fler som använder sig av maskinöversättning. Förväntningarna varierar: om man översätter till ett språk som man inte alls förstår, kan det vara svårt att se om något blir fel. Man kan också tänka efter modellen ”datorn måste ju ha rätt”. Då blir förväntningarna på en korrekt översättning lika självklara som när man skriver $2+2$ i ett kalkylatorprogram och väntar sig svaret 4.

För de flesta är det dock klart att maskinöversättning måste tas med en nypa salt. Den ger ett ungefärligt resultat, som duger väl för att ge en uppfattning av innehållet i en främmande text. Men den räcker knappast till för att producera en version av det egna företagets hemsida som man lugnt kan lägga ut på webben och ta ansvar för. För sådana uppdrag anlitar man fortfarande riktiga översättare.

Google Translate kan knappast jämföras med kalkylatorer. En bättre jämförelse är sökprogram, sådana som Googles egen. Google-sökning brukar ge hyfsat bra resultat, men man kan aldrig vara säker på att just de sidor som bäst motsvarar sökorden hamnar överst. I själva verket når jämförelsen mellan Google-översättning och Google-sökning ännu längre: dessa program använder i slutändan samma teknik, nämligen statistik om språket. Resultatet – översättningen av en text eller listan med sökresultat – produceras som en gissning. Denna gissning är mycket kvalificerad då den är baserad på material med miljoner eller miljarder av ord. Men den är ändå en gissning, och inte en säker beräkning lik $2+2=4$.

Varför är det så här? Om man tänker på sökning, så är det svårt att föreställa sig något bättre sätt att få fram en lista på de tio bästa dokumenten från hela webben som motsvarar ett sökord. Ingen människa har kapacitet att göra detta för hand. Men för översättning vet vi ju att människor är bättre. De enda fördelarna med maskinöversättning är att den är billigare och snabbare. Om man verkligen vill, och har resurser, kan man anställa folk och därmed lösa översättningsproblemet utan att nöja sig med ungefärliga resultat.

Men återigen: varför är det så? När det gäller att beräkna $2+2$ finns det ingen märkbar skillnad mellan människor och datorer. Men när man ska beräkna en division mellan 100-siffriga tal är datorn både snabbare och pålitligare. Skulle datorn kunna ha samma fördel när man översätter hundratusen sidor av Wikipedia och inte bara en företagshemsida på femtio rader?

Svaret är både ja och nej. Just nu finns det inget datorprogram som ens kan översätta företagshemsidor med pålitlig kvalitet. Men Google kan översätta hur många sidor som helst, med en kvalitet som blir allt bättre. Och det finns datorprogram som kan göra saker som liknar översättning. En *kompilator* är ett program som översätter programkod, t.ex. Java-kod, till maskinkod, dvs. till ”ettor och nollor”. Tack vare kompilatorer behöver programmerare inte själva skriva

```
0000101 00011011 01101000 00011100 01100000 00111011
```

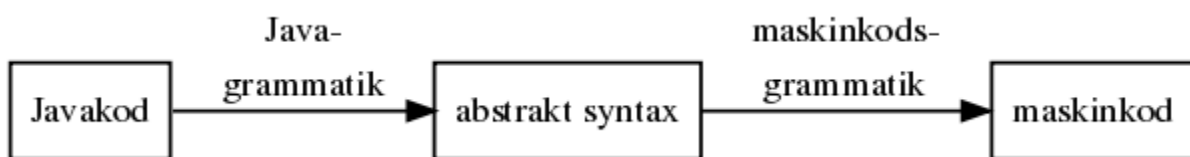
utan något som är lättare för en människa att förstå och få rätt:

$$x = 2 * y + z$$

Och programmeraren kan lita på att kompilatorn översätter Java-koden till maskinkod som betyder exakt samma sak. I själva verket är kompilatorn en *kalkylator*. Den är mycket mera komplicerad, men använder samma principer och har samma precision.

Översättning med grammatik

Den grundläggande tekniken i kompilatorer är *grammatik*. Varje programspråk har en grammatik, dvs. ett regelsystem som säger hur språket är uppbyggt. Det första steget i en Java-kompilator är att använda Java-språkets grammatik för att analysera programkoden. Det sista steget är att använda maskinspråkets grammatik för att producera ettorna och nollorna. Däremellan sitter en *abstrakt syntax*, ett formelspråk som kompilatorn använder internt för att representera den gemensamma *meningen* (dvs. *betydelsen*) i Java-koden och maskinkoden, se figur 1.



Figur 1. En abstrakt syntax representerar meningen i Java-koden och maskinkoden.

Vi ska återkomma till den abstrakta syntaxen och förklara hur den representerar mening. Det vi behöver veta nu är att kompilatorn garanterat producerar maskinkod som betyder samma sak som programmerarens Java-kod – garantin kommer från den gemensamma abstrakta syntaxen.

För de flesta är grammatik något som är känt från läroböcker i språk. En tysk grammatik är en samling regler för det tyska språket – något som man måste lära sig för att tala, skriva och förstå tyska. En stor del av en grammatik består av *böjningsregler*, som producerar olika former av ord. Om vi hoppar direkt till det extrema, kan vi ta finsk grammatik som exempel. Finska substantiv har minst 26 olika former, och man kan komma upp till 3 744 om man räknar på ett lämpligt sätt. Dessa former produceras med komplexa, men fullständigt precisa regler. Om vi tar ordet ”yö” (natt) som utgångspunkt, får vi följande 26 former:

yö, yön, yötä, yöksi, yönä, yössä, yöstä, yöhön, yöllä, yöltä, yölle, yöttä, yöt, öiden, öitä, öiksi, öinä, öissä, öistä, öihin, öillä, öiltä, öille, öittä, öineen, öin

En finsktalande person har ingen som helst svårighet att känna igen och producera vilken som helst av formerna – även om hon aldrig sett den förut. Ett intressant drag i det valda exemplet är att den regelbundna böjningen ändrar den första bokstaven i ordet. Därför kommer man inte långt om man använder en ordbok som enda redskap. För att veta vad ”öistä” betyder måste man lista ut att det är en form av ”yö” och sedan slå upp på bokstaven ”y” i stället för ”ö”. Sedan måste man känna igen att formen är plural elativ, och räkna ut betydelsen ’från nätterna’.

Datorn kan böja ord lika bra som människan. De flesta världsspråk och europeiska språk har datorprogram som kan analysera och producera ord i alla deras former. Men det behövs mer än ordböjning för att en grammatik ska kunna översätta. Det behövs även regler för hur man kombinerar ord till satser och meningar. Dessa regler kallas för *satslära* eller *syntax*.

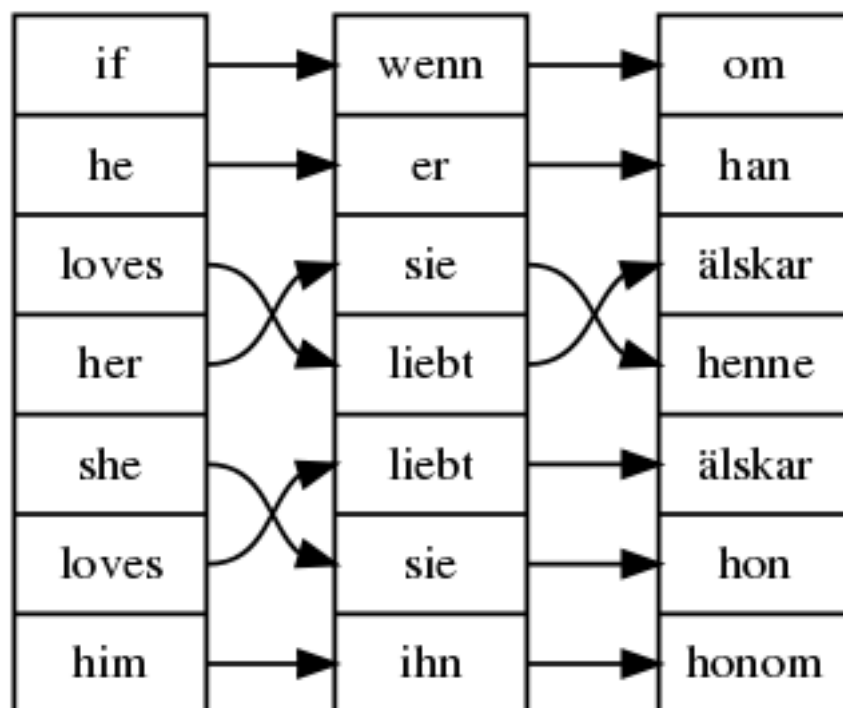
Huvudfrågorna i satsläran är *ordföljd* och *kongruens*. Kongruensreglerna säger vilka former av ord som används i vilka sammanhang, medan ordföljdsreglerna bestämmer ordens ordning. Dessa regler varierar stort från ett språk till ett annat. Tyskan, till exempel, har kon-

gruens mellan verb och subjekt, som svenskan inte har (fast den tidigare haft det). Man säger således

han blir rik → er wird reich
de blir rika → sie werden reich

Verbet ”blir” på tyska är ”wird” i singular och ”werden” i plural. Men exemplen visar även att svenskan har kongruens mellan subjekt och adjektiv, vilket tyskan inte har: både ”rik” och ”rika” blir ”reich”.

Tyskan ger som bekant bra exempel på ordföljdsreglerna också. Om vi sätter en mening på engelska, svenska och tyska sida vid sida, får vi de motsvarigheter mellan orden som visas i figur 2.



Figur 2. Ordföljd i en mening på engelska, svenska respektive tyska.

Figur 2 visar att engelskan har ordföljden subjekt–verb–objekt i båda satserna, medan tyskan och svenskan har verb–subjekt–objekt i den andra satsen och tyskan dessutom subjekt–objekt–verb i den första.

Syntaxreglerna är inte svåra för datorer. I översättningsprogram är nyckeln den abstrakta syntaxen. En villkorssats, som i bilden ovan, har en abstrakt syntax som kan betecknas med formeln

(Cond (Pred subj1 verb1 obj1) (Pred subj2 verb2 obj2))

Formeln är uppbyggd med konstanten Cond, som betecknar en villkorssats, och konstanten Pred, som betecknar *predikation*, dvs. sammansättningen av ett subjekt, ett verb och ett objekt till en sats. Formeln ovan består av två predikationer, som kombineras till en villkorssats.

Olika språk omvandlar den abstrakta syntaxen till olika ordsekvenser, samt använder sig av olika ord för det abstrakta begreppet Cond:

engelska: <i>if</i>	subj1 verb1 obj1 subj2 verb2 obj2
tyska: <i>wenn</i>	subj1 obj1 verb1 verb2 subj2 obj2
svenska: <i>om</i>	subj1 verb1 obj1 verb2 subj2 obj2

Det finns program som kan en hel del om syntaxen hos europeiska språk och världsspråk. De är oftast forskningsprototyper snarare än produktionsmogna system. Men principerna är klara, och syntax, hur hopplös den än kan kännas för en svensk som vill lära sig tyska eller finska, är något som kan hanteras med en kalkylator.

Ett grammatikbaserat maskinöversättningssystem skulle kunna fungera på följande sätt:

1. Analysera orden i källtexten.
2. Hitta den abstrakta syntaxen.
3. Ersätt källspråkets ord med målspråkets ord.
4. Välj de korrekta formerna i målspråket och producera dem med målspråkets ordföljd.

Den här proceduren är tyvärr fortfarande för enkel. Den *kan* ge bra översättningar, men bara i sådana fall där texten kan översättas ord för ord, med oförändrad syntaktisk struktur – såsom i figur 2 ovan.

I många fall krävs det att den syntaktiska strukturen ändras för att översättningen ska bli korrekt. Ett typiskt exempel är ”jag gillar det här”, där ”jag” är subjekt och ”det här” är objekt. Den bästa italienska översättningen är ”questo mi piace”, där ”questo” (det här) är subjekt och ”mi” (mig) är objekt! Man måste således ändra den syntaktiska strukturen när man översätter – omvandla subjekt till objekt och vice versa.

Men sådana här fall kan hanteras med en abstrakt syntax som är abstraktare än den grammatiska strukturen. Den ska representera *begreppet* ’gilla’ och säga vem som gillar och vad. Vi kan skriva (Like x y) för denna begreppsstruktur: som en matematisk formel som uttrycker att x gillar y . Sedan kan vi ange översättningsregler, som visar hur begreppet översätts till en syntaktisk struktur, som i sin tur producerar en ordsekvens med varje språks regler för predikation:

engelska: (Pred x like y)	x like y
italienska: (Pred y piacere x)	y x piace
svenska: (Pred x gilla y)	x gillar y

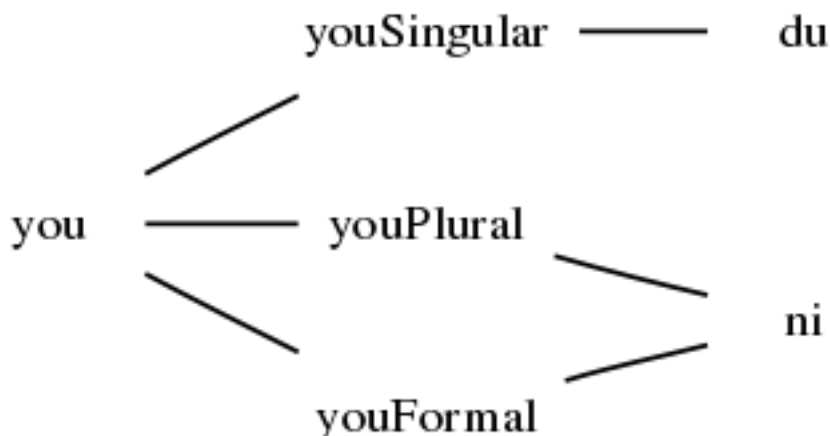
I predikationen ingår kongruensregler som skapar olika former av verbet, subjektet och objektet. Italienskan har även variation i ordföljd, där placeringen av objektet x beror på om x är ett pronomen. Allt detta är välkänt och kan hanteras med ett datorprogram.

Flertydighet

Det svåraste problemet i översättning ligger varken i böjningen eller i syntaxen. Problemet är *flertydigheten* – att en text ofta kan tolkas på olika sätt. Varje tolkning har sin egen abstrakta syntax, som i sin tur kan ha sin egen översättning i ett annat språk. Detta är sällan något problem i kompilatorer, eftersom programspråk vanligen är konstruerade för att vara entydiga. Men naturliga språk är flertydiga på ett nästan obegränsat sätt.

Låt oss ta ett mycket enkelt exempel. Engelskan har ordet ”you”, som på svenska har två motsvarande ord, ”du” och ”ni”. De möjliga översättningarna är i själva verket tre, eftersom ”ni” uttrycker både plural och formellt tilltal. Således har den engelska satsen ”you are rich” tre möjliga översättningar: ”du är rik”, ”ni är rika” och ”ni är rik”. Dessa tre varianter av

”you” kan representeras med tre begrepp i den abstrakta syntaxen. Figur 3 visar vilka flertydigheter som uppstår i engelska och svenska gentemot den abstrakta syntaxen.



Figur 3. Flertydigheter i engelska och svenska gentemot den abstrakta syntaxen.

Även om de enskilda orden är flertydiga, blir de ofta entydiga i en *kontext*. Till exempel, ”you are a student” måste vara singular, medan ”you are students” måste vara plural. Utifrån dessa exempel kan man luras att tro att det hela beror på substantivets numerus (”student” vs. ”students”). Men det beror också på substantivets betydelse: ”you are a couple” kan bara översättas ”ni är ett par”, medan ”you are nuts” kan betyda ”du är galen”, ”ni är galna” eller ”ni är galen”, eller kanske under vissa omständigheter ”ni är nötter”.

Det är kontextens roll i tolkningen av flertydighet som är det riktigt svåra i maskinöversättning. Böjning och syntax kan programmeras på ett precist och uttömmande sätt. Men vad som exakt menas med ett ord eller en sats kan bero på många faktorer. I de enklaste fallen är dessa faktorer syntaktiska och kan härledas från kongruens med andra ord i samma sats (såsom i ”student” vs. ”students”). Men de kan också komma från satser långt före eller efter, från andra ords tolkningar (såsom med ”nuts”), och till och med från något helt utanför texten. Översättningen av satsen ”you are rich” beror i slutändan på vem eller vilka den sägs till, och detta kan vara underförstått snarare än uttryckt i texten.

Översättning med statistik

Vi har sett att en dator kan programmeras så att den böjer orden rätt och bygger korrekta satser. Att man i princip kan göra detta borde vara begripligt för vem som helst som kan sin grammatik och har programmerat. Men i praktiken krävs det mycket arbete, och det kan vara omöjligt att bygga grammatiker som täcker språk i alla deras variationer och ändringar, och som dessutom kan tolerera alla grammatikfel som kan förekomma i riktiga texter.

Det finns dock ett annat sätt att bygga översättningsprogram: med statistik. Denna teknik behöver inga kunskaper om språk och inget mödosamt arbete med grammatiker. Den är därför den dominerande tekniken idag, och används i synnerhet i Google Translate. Google Translate täcker för närvarande över 50 språk, och nya läggs till löpande. Till exempel, efter jordbävningen på Haiti år 2010 kunde Google på några månader skapa en modul för översättning mellan haitisk kreol och de tidigare språken i systemet.

Men hur kan man översätta utan grammatik? De två huvudidéerna är *ordlinjering* (eng. word alignment) och *n-gram*. Dessa tekniker är så allmänna att de kan tillämpas på vilka *sekvenser* som helst, t.ex. i biokemi på sekvenser där proteiner spelar samma roll som ord i översättning. Därför behöver teknikens användning ingen kunskap om språk. I ordlinjering

har man som utgångspunkt två *parallella texter*, d.v.s texter som är översättningar av varandra. Genom att undersöka vilka ord som förekommer på motsvarande ställen i de två texterna kan man med viss sannolikhet säga att ett ord är en översättning av ett annat.

Bilden ovan (figur 2) med ”om han älskar henne älskar hon honom” är ett litet exempel på en parallell text på engelska, tyska och svenska. Vad pilarna i bilden visar är precis ordlinjeringar: att ”wenn” motsvarar ”om”, ”liebt” motsvarar ”älskar” och så vidare. De korsande pilarna i bilden visar att ord som motsvarar varandra inte behöver förekomma på exakt samma ställe.

Ordlinjeringarna i figur 2 ovan har producerats av en grammatik. Men i statistisk översättning har man bara texterna att utgå ifrån. Det behövs mycket text för att kunna bestämma linjeringarna på ett någorlunda pålitligt sätt, speciellt om det blir korsande pilar. En annan komplikation är att olika språk kan använda olika antal ord för att uttrycka samma sak. Till exempel blir ”huset” på svenska ”the house” på engelska. Då måste man dra pilar från ett ord till två samtidigt, vilket är ännu svårare att lista ut från bara texterna.

Som regel behöver bra ordlinjering en parallell text med minst en miljon ord på vardera språk. Även då kan man ta fel. Det berömda exemplet med ”svensk” och ”Amerikaner” får en möjlig förklaring utifrån den här tekniken. De parallella texterna kan ha haft ”svensk” och ”Amerikaner” på motsvarande ställen, eftersom den svenska texten har varit anpassad för svenskar och den tyska texten för... nej, så kan det inte ha gått till. Då skulle man ju ha ”Deutscher” i stället för ”Amerikaner”! Men nu har vi ändå fått hälften av en förklaring.

Den andra hälften avslöjas när man låter Google Translate översätta ”jag är svensk” till engelska och får resultatet ”I am American”. Det som händer mellan svenska och tyska är att översättningen går via engelska! Detta kan låta fullständigt obegripligt om man tänker på att tyskan faktiskt är närmare svenskan än engelskan. Men man kan ändå förstå varför Google har gjort så: det finns inte så mycket parallell text mellan svenska och tyska som det finns mellan engelska och de flesta andra språk.

Enligt Franz Och, som leder Googles översättningsprojekt, görs mer än 90 % av Google-översättningar via engelska. Ett skäl är att man då har den bästa tillgången till parallella texter för olika språk. Man kan bara tänka på hur mycket parallella texter det finns mellan t.ex. estniska och baskiska på nätet – knappast mer än Bibeln. Ett annat skäl är att användningen av en *interlingua*, ett mellanspråk, gör det mycket lättare att bygga ut systemet. För att översätta mellan varje språkpar bland 50 språk behövs 2 450 översättningsprogram. Men om ett av språken används som interlingua kan alla dessa byggas som kombinationer av bara 98 program.

En annan sak som snabbt avslöjar att engelskan används som interlingua i Google Translate är att både ”du” och ”ni” brukar bli ”du” på tyska – och till och med på norska, där övergången via engelska känns ännu konstigare om man tänker lingvistiskt. Men som sagt: interlingua är den bästa tekniken om man vill skapa system som hanterar riktigt många språk. Den används såväl i grammatiska som i statistiska översättningssystem. Men om man kan engelska, skall man som användare av Google Translate helst välja översättning till eller från engelska i stället för svenska för att få det bästa resultatet. Något förenklat, om Google-översättning får 50% av texten rätt i varje steg (t.ex. tyska–engelska eller engelska–svenska), så får man bara 25% rätt om man översätter hela vägen från tyska till svenska!

Nu kan vi förstå varför ”svensk” blev ”American” i Google Translate: ordlinjeringen har funnit de här orden på samma plats tillräckligt ofta. Av samma skäl kan ”Stockholm” bli ”New York” och ”10 000 kronor” bli ”10,000 dollars”. Ibland är resultaten helt obegripliga: det finska ordet ”öille” (för nätterna) blir ”compensating those Member States” på engelska och ”kompensera de medlemsstater” på svenska. Där har den automatiska ordlinjeringen kanske sett bara en förekomst av ordet och misslinjerat den totalt.

Glesa data (sparse data) är faktiskt en av de stora utmaningarna för statistisk översättning. Det kan hända att ett visst ord aldrig har förekommit i de texter som systemet har sett. Detta är fallet i Google Translate med tre av de 26 böjningsformerna av finskt ”yö”. Då får man ordet självt som ”översättning”.

Om ordlinjering är de statistiska metodernas motsvarighet till ordböcker, är n -gram deras syntax. Utifrån grammatiken vet vi att ”jag” är formen som används som subjekt och översätts med ”I” på engelska, medan ”mig” är objektformen och översätts med ”me”. Ett rent statistiskt system vet ingenting om subjekt och objekt. Men det har sett tillräckligt med engelsk text för att veta att ”I see him” och ”he sees me” är möjliga ordsekvenser, medan ”me see him” och ”he sees I” nästan aldrig förekommer. Sådana sekvenser av tre ord kallas *trigram*. De är n -gram med $n=3$.

När ett statistiskt system producerar översättningen, föredrar den texter som består av de längsta och sannolikaste n -grammen i målspråket. På grund av problemet med glesa data måste n vara ganska litet för att det ska finnas en chans för de flesta n -grammen att alls förekomma. Som regel är en text på tio miljoner ord i ett språk en förutsättning för att man ska kunna bygga en bra n -grammodell för språket, och då talar man om n som är högst 4 eller 5. Enkla experiment med Google Translate bekräftar återigen att systemet använder n -gram. Låt oss översätta från svenska till engelska och lägga till ord mellan ”we” och ”rich”:

we are rich → vi är rika
we are not rich → vi är inte rika
we are not very rich → vi är inte särskilt rik
we became rich → vi blev rik

Här har vi det! Med ordet ”är” emellan ryms ”vi” och ”rika” med ett och samma trigram, som dessutom säkert är ganska vanligt, och kongruensen blir rätt. Men med tre ord emellan blir avståndet för långt. Också med bara ett ord, ”became”, som är tillräckligt ovanligt, försvinner förbindelsen mellan ”vi” och ”rik”.

Nu är kanske inte grammatik med rätt kongruens så viktigt om man bara vill få en uppfattning om vad en text säger. Men läget blir ett annat om ord på långt avstånd bidrar till betydelsen. Detta händer ofta med tyska. Verbet ”umbringen” (döda) är ett exempel på s.k. samman-satta verb, där partikeln ”um” och huvud verbet ”bringen” kan hamna långt ifrån varandra. Och då ändras Google-översättningen:

er bringt dich um → he is killing you
er bringt dich gerne um → he brings up to you

Den första meningen blir rätt, men den andra borde vara ”he kills you with pleasure”.

Hybridmetoder

Vi har sett att grammatiska och statistiska översättningsmetoder har olika styrkor. Med grammatik kan man känna igen ordformer som aldrig har använts förr, och man kan garantera att översättningen blir grammatiskt korrekt. Med statistik kan man bygga översättningssystem automatiskt, snabbt och billigt. Man kan dessutom översätta vad som helst: man får alltid ett resultat, vilket ofta är bättre än ingenting. En grammatik kan ofta misslyckas totalt med ett för programmerare så bekant ”syntax error”.

De mest omfattande översättningsprogrammen som finns tillgängliga på webben är statistiska: Google Translate och Microsoft Bing. Systran är ett äldre program, som är baserat på regler snarare än statistik, men inte på ren grammatik utan mera på heuristik, kvalificerade gissningar. Apertium är ett grammatikbaserat program, vars specialitet är nära besläktade

språkpar, t.ex. svenska och danska. Mellan nära besläktade språk kan man översätta med mycket få grammatiska ändringar. Detta innebär att man kan bygga översättningssystem snabbare än med statistik, eftersom det är svårare att hitta parallella texter än att skriva grammatiken. Det är roligt och instruktivt att experimentera med alla dessa fyra system och börja med de små exempel som visats i den här uppsatsen.

Men kan man inte kombinera fördelarna med grammatik och statistik? Denna fråga är en av de viktigaste i dagens forskning kring maskinöversättning. Kombinationerna bär namnet *hybridmetoder*.

Kanske den vanligaste hybridmetoden är att låta en grammatik analysera ord till deras grundformer och tillämpa statistiska modeller på dessa, i stället för på oanalyserade ord. Denna hybridmetod är till stor hjälp med glesa data. Om t.ex. finska ord har i genomsnitt 100 former, och det finns 10 000 ord som kan förekomma i texter, finns det en miljon möjliga ordformer, en biljon möjliga 2-gram, en triljon möjliga 3-gram osv. Men om man i stället analyserar alla ord till grundformen, har man "bara" en biljon 3-gram – en miljonfaldig minskning! Då finns det i alla fall lite mera hopp om att hitta de viktigaste 3-grammen i en text.

En annan hybridmetod är att använda statistik för att bygga delar av grammatiker automatiskt. De i särklass talrikaste reglerna i en översättningsgrammatik handlar om enskilda begrepp, sådana som "x gillar y". En sådan regel behövs i princip för varje verb, och dessutom för alla olika användningar av varje verb: "x tycker om y" är ett annat begrepp än "x tycker att y" och "x tycker illa". Det kan behövas tiotusentals sådana regler, och det är mycket arbetskrävande att skriva dem för hand. Men eftersom dessa begrepp oftast kan uttryckas med korta meningar, finns det hopp om att de fångas av en statistisk modell med korta n-gram. Sedan kan man använda de andra reglerna i grammatiken för att bygga längre meningar, där statistiska modeller inte fungerar så bra.

Vi kan göra ytterligare ett experiment med Google Translate. Vi vill veta hur "x becomes y" översätts till svenska. Vi väljer en kort och typisk mening som utgångspunkt och får ett korrekt resultat:

he becomes rich | → han blir rik

Vi generaliserar exemplet till en allmän regel i grammatiken:

(Become x y) → (Pred x bli y)

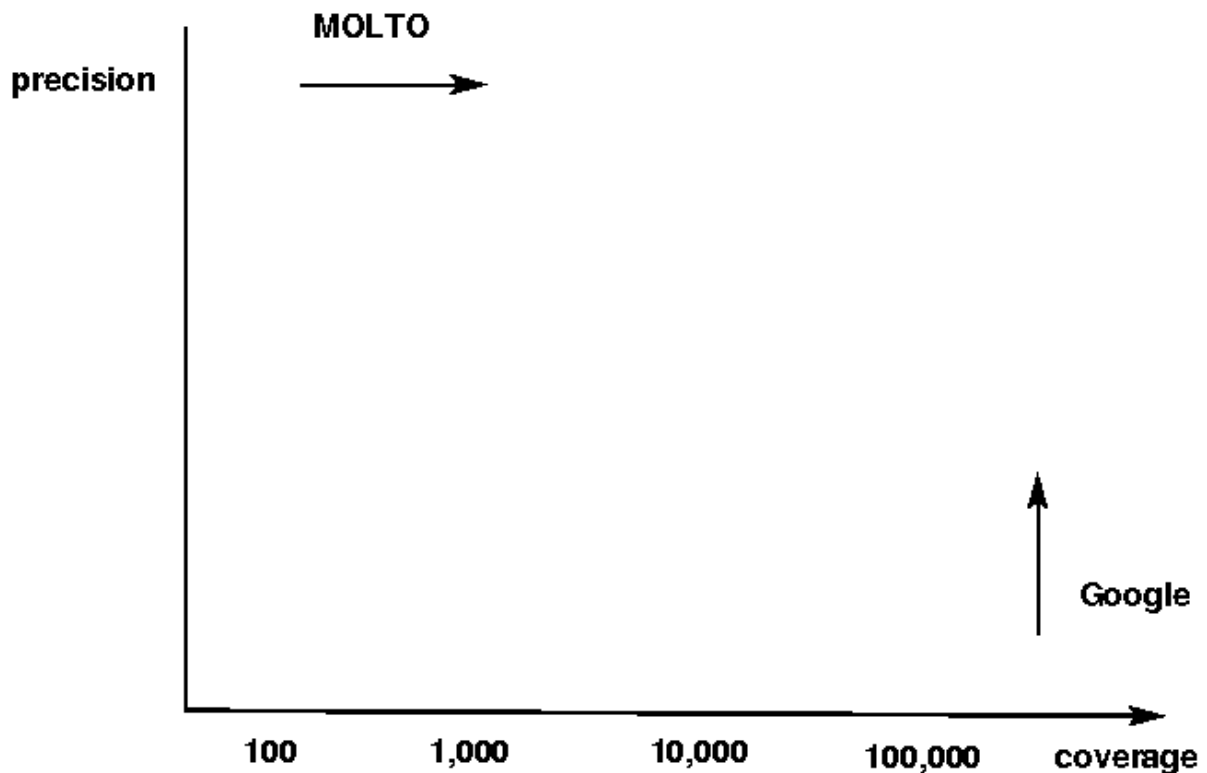
Den här regeln kan i kombination med andra regler i den svenska grammatiken översätta komplexa meningar, sådana som

if the company doesn't become rich, its owners will become poor

Den korrekta översättningen är lätt att finna med hjälp av de allmänna reglerna för kongruens och villkorssatser. Men ett rent statistiskt program kan knappast få alla detaljer rätt.

När kan man lita på maskinöversättning?

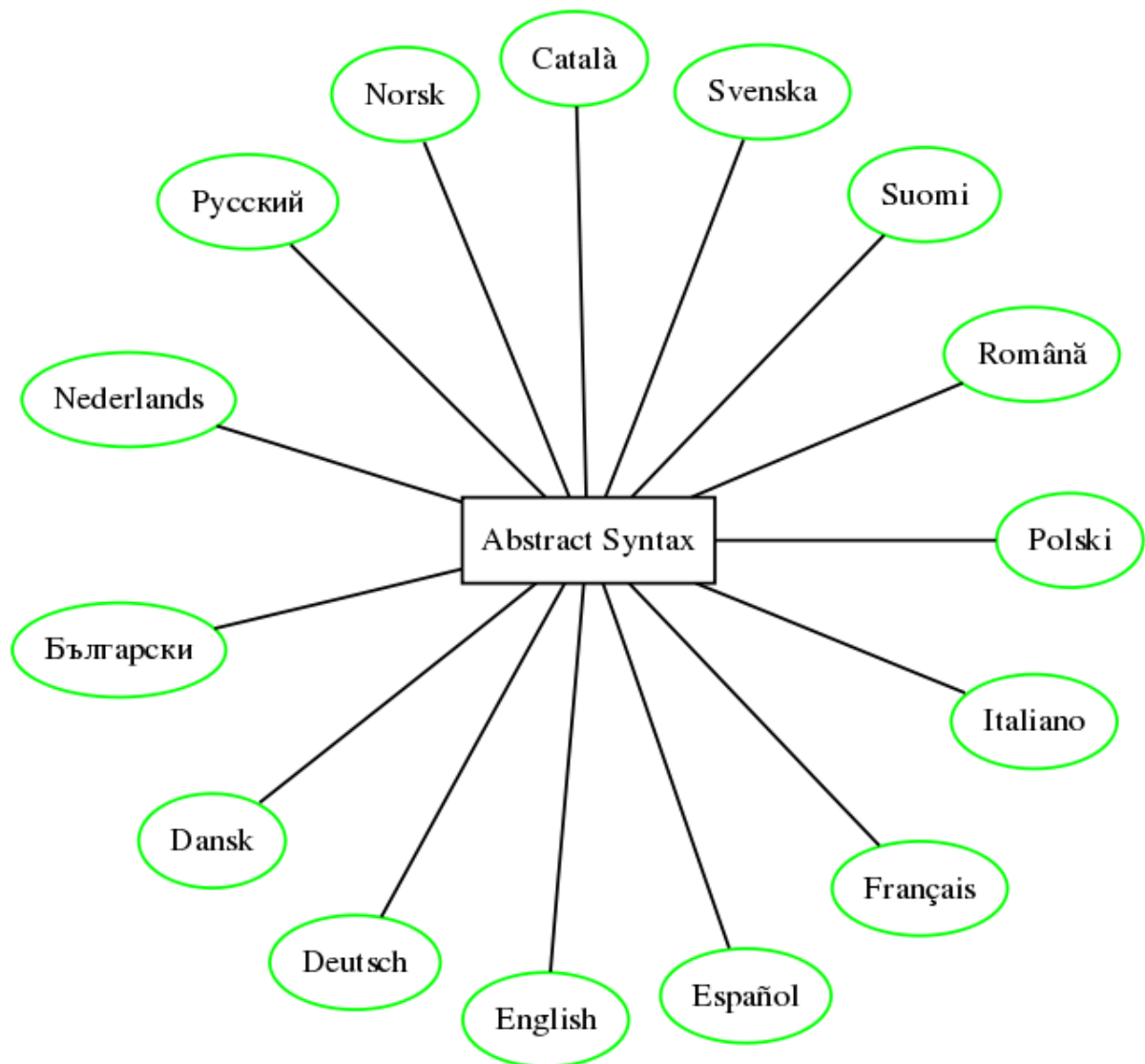
Allt vi vet om maskinöversättning tyder på att grammatik bör kompletteras med statistik för att öka dess *täckning*, och att statistik måste kombineras med grammatik för att öka dess *precision*. Diagrammet i figur 4 visar en jämförelse mellan Google Translate och MOLTO i nuläget. På x-axeln visas systemens täckning – hur många koncept som kan översättas. På y-axeln visas systemets precision.



Figur 4. Jämförelse mellan Google Translate och MOLTO i precision och täckning.

Diagrammet jämför det statistiska Google Translate med MOLTO, som har en teknik baserad på grammatik. MOLTO har som utgångspunkt *perfekt* precision och en täckning på endast hundratals begrepp, medan Google Translate täcker minst hundratusentals begrepp med en lägre precision. I framtiden kommer MOLTO att nå större täckning utan att tappa i precision, medan Google Translate kommer att öka sin precision med bibehållen täckning. Det är inte troligt att man någonsin kommer att kunna förena fullständig precision med fullständig täckning. Därför måste man i varje enskilt fall göra en avvägning.

MOLTO är baserat på den tidigare beskrivna kompilatormodellen för översättning. Översättningsprogram i MOLTO fungerar på begränsade områden, t.ex. matematiska övningar och farmaceutiska patent. Som första steg byggs en abstrakt syntax som fångar alla begrepp på området. Som andra steg anges regler för hur dessa begrepp uttrycks i de språk man vill översätta mellan. När ett nytt språk läggs till, räcker det att definiera dess relation till den abstrakta syntaxen. MOLTO använder en interlinguaarkitektur, precis som Google Translate. Figur 5 visar MOLTOs översättningsmodell och vilka språk som är med för närvarande.



Figur 5. MOLTOs översättningsmodell med de språk som går att översätta mellan.

Google Translate kan karaktäriseras som ett program för *informationskonsumenterna*, medan MOLTO är skapat för *producenterna*. Ett enkelt exempel uppvisar skillnaden. På grund av ordlinjeringen kan det hända att ett pris på ”99 euros” översätts till ”99 kronor”. Om översättningen skapas av en konsument som läser ett utländskt företags webbsida genom Google Translate, har företaget inget ansvar för prisinformationen. Men om den är utlagd av företaget självt, kan konsumenten kräva att faktiskt få varan till detta otroligt facila pris.

I ett översättningsprogram för producenter är precisionen ett väsentligt krav. Däremot behöver inte täckningen vara obegränsad. Det räcker att programmet känner till företagets sortiment och betalsätten, eller vad det nu är producenten behöver publicera. MOLTOs uppdrag är att underlätta byggandet av översättningssystem av denna karaktär.

Så: när kan man lita på maskinöversättning? Frågan kan tolkas på två sätt. Antingen: när i framtiden kommer maskinöversättning att vara så bra att man alltid kan lita på den? Eller: i vilka situationer i nutiden kan man veta att den maskinöversättning man använder är pålitlig? Svaret till den förra frågan är troligen ”aldrig”. Det kommer knappast en tid då en maskin kan översätta vad som helst med fullständig pålitlighet. Människor kommer alltid att behövas både som översättare och som utvecklare av bättre översättningssystem. Men den senare frågan har ett hoppfullare svar: visst kan man redan nu översätta med full kvalitet på begränsade områden, och dessa områden kommer i framtiden att vara bredare och fler.

Länkar

Google Translate: <http://translate.google.com>

Ett föredrag av Franz Och: http://www.youtube.com/watch?v=y_PzPDRPwIA

Bing Translator: <http://www.microsofttranslator.com/>

Apertium: <http://www.apertium.org/>

MOLTO-projektet: <http://www.molto-projekt.eu>

Författarens hemsida: <http://www.cse.chalmers.se/~aarne>

Tack

Tomas Brandberg, Bengt Nordström och Thomas Hallgren gav värdefulla kommentarer.
MOLTO finansieras av EU-anlaget FP7-ICT-247914.