

Datoriserad analys av sammansättningar i teknisk text

Stina Åberg
stinaka@stp.ling.uu.se

Examensarbete i datorlingvistik
Språkteknologiprogrammet
Uppsala universitet · Institutionen för lingvistik

3 december 2003

Handledare:
Anna Sågvall Hein, Uppsala universitet

Sammanfattning

Syftet med det här arbetet var att komplettera den svenska delen av MatsLex-databasen med information om vilka ord som är sammansättningar och de delar dessa utgörs av. Genom att korrekt kunna identifiera och segmentera en sammansättning kan ett översättningssystem öka andelen översatta ord.

Som ett första led i arbetet användes Uppsala Chart Parser, UCP, för att analysera samtliga ordformer i databasen. Utvärderingen av detta program visar att relativt många sammansättningar missas, men att ett par enkla åtgärder skulle kunna förbättra resultatet avsevärt.

Utifrån UCP-analyserna formulerades regler till ett program som väljer den bästa analysen för en ordform i de fall där UCP har kunnat analysera ordet på mer än ett sätt. Vissa analyser som plockades ut var felaktiga eftersom UCP genererade enbart felaktiga analyser för några sammansättningar samt att det fanns ett par undantag till varje regel i analysvalsprogrammet. 95 % av de analyser som plockades ut var dock korrekta.

11617 av 17557 ord i databasen visade sig vara sammansättningar, vilket stöder påståendet att sammansättningar är mycket vanliga i teknisk text. Dessutom visar det på hur viktigt det är att kunna hantera dessa på ett bra sätt i ett maskinöversättningssystem.

Innehåll

1 Inledning	2
1.1 Uppgiften	2
1.2 Syfte	3
1.3 Hypoteser	4
1.4 Avgränsningar	4
1.5 Översikt över uppsatsen	4
2 Bakgrund	6
2.1 Vad är en sammansättning?	6
2.1.1 Efterled	7
2.1.2 Förled	7
2.1.3 Fogen	8
2.2 Sammansättningsanalys	11
2.2.1 Tidigare arbete	11
2.2.2 UCP	13
3 Metod	15
3.1 Utgångspunkt/material	15
3.1.1 Indata	15
3.1.2 UCP-analyser	18
3.2 Algoritm för analysvalsprogrammet	20
3.3 Implementation/genomförande	27
3.4 Manuellt arbete	30
4 Resultat	31
4.1 UCP	31
4.2 Analysvalsprogrammet	31
5 Utvärdering av UCP	33
6 Utvärdering av analysvalsprogrammet	35
7 Sammanfattning	37
8 Utveckling/nästa steg	38
A Programkod	41
B Testkörning	54

1 Inledning

De flesta nybildade orden i modern svenska är sammansättningar. Somliga har med tiden blivit lexikaliserade, men dessa kan tas upp i ett lexikon och ställer därför inte till några problem i olika NLP-tillämpningar. Istället är det de för stunden nyskapade sammansättningarna som orsakar problemen. Eftersom det teoretiskt sett är möjligt att skapa en infinit mängd olika sammansättningar kan inget lexikon rimligtvis försöka ta upp dessa. Sammansättningar som hittas på för stunden och sedan aldrig används mer, som t.ex. tidningarnas (ibland något krystade) sammansättningar av typen *badhusmördarrättegången*, vill man heller inte ta med i ett lexikon även om så vore möjligt. Ett analysprogram som hanterar sammansättningar på ett bra sätt är därför en mycket viktig del av t.ex. ett maskinöversättningssystem.

Om samtliga ord som inte känns igen av datorn p.g.a. att de inte finns med i lexikonet skulle markerats som okända ord skulle den andel ord som programmet kan hantera minska avsevärt. Eftersom många av sammansättningarnas semantiska innebörd är likvärdig med kombinationen av delarnas semantiska innehåll går det ofta att komma fram till översättningen av en sammansättning genom att ha tillgång till de olika delarna. Genom att korrekt kunna identifiera en sammansättning och dess delar kan alltså ett översättningssystem öka sin andel översatta ord och framför allt minska andelen ord som inte översätts alls.

1.1 Uppgiften

För att få ett bra resultat vid maskinöversättning gäller det att ett analysprogram inte enbart kan hitta en sammansättning utan dessutom kan avgöra vilka delar sammansättningen består av. Detta är inte alltid en trivial uppgift. I många fall får en sammansättning ett flertal analyser¹. Ju fler led en sammansättning innehåller, desto fler olika analyser kan den få. Antalet analyser påverkas också av lexikonets storlek och hur många småord (t.ex. *en*, *al*) det innehåller.

Vissa sammansättningar kan vara svåra att segmentera även manuellt, eftersom det finns två eller flera tänkbara analyser. Exempel på ett sådant ord är *bildrulle*, som kan segmenteras antingen som *bil-drulle* eller som *bild-rulle* (exempel från Svenska Akademiens ordlista). I detta fall är det kontexten som avgör vilken betydelse av ordet det är frågan om. Eftersom kontextinformation är svår att lägga in i analysprogram måste man vid datamaskinell analys istället formulera regler för den mest sannolika segmenteringen. Undantag från dessa regler kommer då att få en felaktig analys, varför reglerna måste formuleras mycket noggrant.

¹I denna rapport innebär en *analys* en uppdelning av en sammansättning i mindre delar.

Det lexikon som undersökningen baserats på, MatsLex², är dock så pass litet och domänspecifikt att dessa fall av dubbla analysmöjligheter borde vara färre, eftersom det begränsade ordförrådet inom domänen gör att det inte finns så många analysmöjligheter. I bildomänen finns kanske inte ordet bild med och i fotosammanhang används inte ordet bil så ofta.

Uppgiften för examensarbetet är tredelad. Det första delmomentet går ut på att köra och utvärdera ett sammansättningsanalysprogram som genererar samtliga möjliga analyser av ett ord. Indata är grundformer ur en lexikal databas. Utifrån analysprogrammets resultat ska sedan regler formuleras till ett program som ska välja den bästa analysen. Till sist plockas huvudordet fram för denna analys för vidare manuella studier av huruvida det finns någon gemensam information för ord med samma huvudord.

Arbetet har gjorts vid institutionen för lingvistik på Uppsala universitet, med stöd av ETAP³.

1.2 Syfte

Syftet med examensarbetet är att ta fram information om vilka ord i den svenska delen av MatsLex-databasen som är sammansättningar och vilka deras huvudord är. Denna information ska läggas in i lexikonet för att kunna vara till nytta vid senare översättningsarbete. Ett möjligt användningsområde är att kontrollera att huvudleden i de fall där de är gemensamma för flera ord kan översättas på samma sätt i en text. Ordet *motor* ska t.ex. inte översättas med *motor* på ett ställe och med *engine* på ett annat i de fall där de båda förekomsterna syftar på samma sorts motor, utan man eftersträvar en konsekvent översättning. En annan tänkbar fortsättning är att undersöka huruvida det är möjligt att ställa upp generella regler för översättning av sammansättningar med gemensamt huvudord.

I databasen fanns sedan tidigare viss information om sammansättningar. Denna hade dock genererats automatiskt (se avsnitt 3.1.2) och var i vissa fall felaktig. Dessutom saknades informationen för vissa sammansättningar. Detta examensarbete ska bidra till att göra lexikonet mer konsekvent angående sammansättningsinformationen.

Eftersom resultatet från ett existerande sammansättningsanalysprogram som bygger på Uppsala Chart Parser (UCP) används som utgångspunkt ingår även en

²MatsLex är en flerspråkig lexikal databas som har utvecklats inom MATS-projektet. MATS = Methodology and Application of a Translation System. <http://stp.ling.uu.se/mats/>
För en beskrivning av MatsLex, se avsnitt 3.1.1.

³ETAP=ETablering och Annotering av Parallellkorpus för igenkänning av översättningsekvivalenter. <http://stp.ling.uu.se/etap/>
ETAP ingår i forskningsprogrammet "Översättning och tolkning som språk- och kulturmöte" som finansieras av Riksbankens jubileumsfond.

utvärdering av detta.

1.3 Hypoteser

Vid arbetets början gjordes följande antaganden.

- Antalet ord med bara två eller tre bokstäver är relativt få, vilket kan förhindra en övergenerering av antalet analyser för de sammansatta orden. Substantiviska småord som *el* och *ar* finns dock, vilket kan ställa till problem i form av att även vissa enkla ord felaktigt ges en sammansatt analys.

- Väldigt många av orden i databasen, åtminstone substantiven, ser ut att vara sammansättningar. Ungefär en fjärdedel av substantiven hade automatiskt analyserats och markerats som sammansättningar i ett tidigare skede (se avsnitt 3.1.1).

- Materialet bör kunna verifiera påståendet att sammansättningar är mycket vanliga i teknisk text.

- UCP bör klara av att analysera de flesta orden, men kan få problem med en del konstruktioner, t.ex. vissa fogeformer som *tidevarv*.

- Den domänspecifika vokabulären borde förhindra en övergenerering av analyser. Eftersom många enkla ord inte finns i lexikonet kan snarare för få analyser ges, vilket kan leda till att resultatet inte blir helt tillförlitligt. Ett ord som *munstycke* kan t.ex. inte få en sammansatt analys, eftersom ordet *mun* inte finns med i lexikonet. Ett lexikon som även innehåller mer generella ord och termer från andra domäner skulle ge fler varianter.

- De ord som kommer att få flera analyser är sannolikt långa ord, d.v.s. sammansättningar av fler än två lexem.

- Den korrekta analysen för varje ord bör vara lätt att plocka ut, eftersom det i många fall endast kommer att finnas en möjlig analys.

1.4 Avgränsningar

Endast ett analysprogram har studerats och endast svenska sammansättningar har plockats ut.

1.5 Översikt över uppsatsen

Uppsatsen inleds med ett bakgrundsavsnitt som inkluderar en beskrivning av begreppet sammansättning och tidigare arbeten med sammansättningsanalys. Därefter följer ett metodavsnitt över arbetet med att skapa ett program som väljer ut den bästa av samtliga möjliga analyser för ett ord. Resultatet av detta arbete presenteras sedan följt av utvärderingar av analysvalsprogrammet samt av den grammatik och de regler som användes för att generera de olika analyserna i utgångsläget. Uppsatsen avslutas med förslag på tänkbara utvecklingsområden.

I uppsatsen används tankstreck (–) för att där det behövs markera gränsen mellan för- och efterled. Vid sammansatta led markeras gränsen mellan de däri ingående delarna med lodstreck (|). Sålunda visar *broms–regler|krets* att gränsen mellan för- och efterled går mellan *broms* och *regler(a)* och att efterledet i sin tur är en sammansättning, med förledet *regler(a)* och efterledet *krets*. Vanligt bindstreck används som markering vid inskjutna bokstäver i fogen. Detta tecken finns även i sammansättningar som skrivs med bindstreck. Bindstreckets utgör då också gränsen mellan leden.

I de fall inget annat anges är exemplen autentiska och hämtade från MatsLex.

2 Bakgrund

Det har forskats relativt mycket kring sammansättningar, vad de är och hur de kan identifieras. Förhållandevis lite har dock skrivits om den datamaskinella aspekten. Detta område utforskas dock mer och mer och under tiden som detta arbete har pågått har t.ex. ordprediceringsprojektet FASTY⁴ utvecklat en funktion för sammansättningsprediktion, som delvis utnyttjat information som tagits fram under det här examensarbetet.

Detta bakgrundsavsnitt är uppdelat i två delar. Den första beskriver vad en sammansättning är och vilka regler som finns för hur den kan bildas. Den andra delen tar upp olika sätt att datamaskinellt hitta sammansättningar och ge dem en korrekt segmentering.

2.1 Vad är en sammansättning?

En sammansättning består av minst två fria morfem som tillsammans utgör en betydelseenhet med enhetlig böjning. Varje sammansättning kan skrivas om till en fras, t.ex. *dekorationsribba* → *ribba för dekoration*.

En sammansättning delas oftast⁵ upp i ett förled och ett efterled. Detta gäller även i de fall där fler än två ord satts samman, eftersom det då oftast rör sig om en sammansättning mellan ett enkelt led och ett led som i sin tur består av en sammansättning (*bromsreglerkrets* har strukturen *a+bc*, [*broms*][*reglerkrets*]), och är en sammansättning av det enkla ordet *broms* och sammansättningen *reglerkrets*).

Sammansättningarna kan delas in i olika typer beroende på relationen mellan leden. Den allra vanligaste sammansättningstypen är den *determinativa*, där efterledet är det semantiska huvudordet och förledet är dess bestämning, d.v.s. förledet är underordnat efterledet (*vridmotor* = ett slags motor). Det finns även *kopulativa* sammansättningar, där för- och efterled är jämställda, t.ex. *sötsur* (termer från Thorell (1981) och Svenska Akademiens grammatik (SAG), exempel från Thorell).

För kopulativa sammansättningar gäller att båda leden måste tillhöra samma ordklass. Oftast kan leden inte byta plats. Det fåtal sammansättningar som hör till denna grupp är ofta egennamn med bindestreck, t.ex. *Saab-Scania*.

⁴EU-projektet FASTY, Faster Typing for Disabled Persons, har skapat ett system för att predica ord och öka handikappades textproduktionshastighet.
<http://www.fortec.tuwien.ac.at/reha.e/projects/fasty/fasty.html>

⁵Flerledade sammansättningar finns, men är ovanliga. Ex.: *svensk-norsk-dansk*. (Liljestrands (1993) exempel).

2.1.1 Efterled

Efterledet kan vara antingen ett rotmorfem (*broms-krets*), en avledning (*broms-provning*) eller en sammansättning (*broms-regler|krets*).

Efterledet bestämmer ordklass och böjningsmönster för hela sammansättningen. Detta är intressant för det här examensarbetet eftersom resultatet av arbetet kan användas för att undersöka huruvida det även går att ställa upp gemensamma regler för sammansättningar med samma efterled. Det är känt att efterledet har stor inverkan på böjningen av ordet. I den här studien undersöks om efterledet även har samma påverkan i maskinöversättningssammanhang.

Den vanligaste ordklassen för efterled och därmed för hela sammansättningar är substantiv. Därefter följer adjektiv och verb.

2.1.2 Förled

Förledet i ett substantiviskt sammansatt ord är oftast ett substantiv, verb eller adjektiv i denna ordning, där substantivet är vanligast. Substantiviska och adjektiviska förled står oftast i grundform, men adjektivens neutrumform undviks. Vid verb faller ofta a:et. Övriga ordklasser förekommer endast sällan som förled och då i grundform.

Förledet kan vara ett rotmorfem (*dörrfäste*), en avledning (*kopplingsarm*) eller en av följande två sammansättningstyper:

1. Förledet kan stå som eget sammansatt ord (*axeltrycksfördelning* → *axeltryck*).
2. Förledet kan inte stå självständigt som sammansättning utan motsvaras då av en flerordsfras (*trepunktsbälte* → **trepunkts* → *tre punkter*, *engångssmörja* → **engångs* → *en gång*).

Grundregeln för den sistnämnda typen, sammansättningar med frasförled, är att de sätts samman av de ord vari de kan upplösas. Undantag från denna regel finns, t.ex. exemplet med *trepunktsbälte* ovan. Fogen mellan frasförled och efterled kräver foga-s i nästan alla fall. Den i databasen mest frekventa typen av ordfogningar (frasförledssammansättningar) är de där förledet har strukturen räkneord+substantiv i singular/plural+s. Samtliga exempel som givits ovan hör till denna grupp.

Substantiv får ofta en ny form då de ingår som förled i en sammansättning. Hur denna s.k. *fogeform* bildas avgörs av stamslutets fonetiska form. SAG presenterar följande former av förledet:

1. Det kan vara en stam (reducerad form): *lamp-* (*hållare*)
2. Det kan utgöras av grundformen (bibehållen form): *metall-* (*hylsa*)

3. Det kan förlängas med böjningsmorfer (utökad form): *tjänst-e-* (*vikt*)

Bibehållen grundform är den vanligaste typen av de tre.

Det vanligaste exemplet på en reducerad form är när ett trycksvagt -a faller. I vissa fall ändras även stavningen som t.ex. vid *trumma* → *trum*.

Ett annat exempel på en reducerad form är de substantiv som slutar på -are. Vid sammansättning faller -e: *strömställarbelysning*.

Vilken fogeform förledet får följer inte alltid ett regelbundet mönster. Ett och samma substantiv som förled kan t.ex. uppträda i olika fogeform i olika sammansättningar (*driftfall*, *driftskontroll*).

I vissa fall går det bra att lista ut hur en ny sammansättning ska fogas. *Färgband* och andra ord på *färg-* ger t.ex. en vink om att *färgkod* är en korrekt sammansättning. Men *tid-s-axel*, och *tid-punkt* ger ingen idé om hur *tid* och *justering* ska sättas samman.

2.1.3 Fogen

Det finns olika sätt att bilda en sammansättning. Det kan t.ex. ske genom juxtaposition, där de båda leden helt enkelt sätts ihop utan ”klister” emellan. Ibland behövs ett bindestreck. I andra fall, dock endast vid substantiviska förled, krävs ett foga-s eller foga-vokal mellan leden.

Foga-vokalerna, a och e, går i de flesta fall tillbaka på gamla genitivändelser. Dessa är bäst bevarade i sydsvenska dialekter: *bonn-a-gård*, *körk-e-torn* (Liljestrands (1993) exempel). I riksspråket antyder det dock en lite högre stilnivå eller ålderdomlig klang.

Vanligare är dock inskott av ett foga-s mellan sammansättningsleden. Huruvida en sammansättning ska ha foga-s eller inte är inte helt enkelt att avgöra. Enligt Delsing (2001) räknar man dock traditionellt med två huvudregler: en *fonetisk regel* som talar om att foga-s inte är förenligt med vissa språkljud, och en *strukturell regel* som säger att foga-s måste finnas om förledet i sammansättningen är flerledat.

Den fonetiska regeln punktats upp på ett tydligt sätt i Svenska Akademiens grammatik (1999), som säger att inskott av foga-s *inte* sker om förledet slutar på

1. s-ljud, sje-ljud eller konsonantgrupp där dessa ljud ingår (*broms-kloss*, *läckage-kanal*)
2. vokal (*gummi-band*)
3. obetonad vokal + r, l eller n (*nyckel-kod*)

Detta gäller oavsett om förledet är en rot eller ett i sin tur sammansatt led.

Den strukturella regeln säger att foga-s sätts in om förledet är flerledat, d.v.s. innehåller mer än ett ordled (ett morfem). Anledningen är att det vid sammansatt förled krävs en tydligare markering av gränsen mellan förled och efterled än vid ett enkelt förled. (Jfr *dragbil* och *dragbil-s-utrustning*). Avledningar är också flerledade och omfattas därför också av regeln.

Wennerberg (1962) anger följande principer för inskott av foge-s.

- Vid sammansatt förled (A|B–C):
Fogen mellan A och B får samma form som vid enkel sammansättning. I fogen mellan AB och C inkommer ett foge-s.
- Vid sammansatt efterled (A–B|C):
Fogen mellan B och C får samma form som vid enkel sammansättning. Fogen mellan A och BC får samma form som vid enkel sammansättning av A och annat ord.

Vid sammansättningar där efterledet är sammansatt tillkommer alltså ofta inget foge-s. Detta innebär att vi tolkar sammansättningar med eller utan s olika (jfr *träbensskydd* och *träbenskydd*) och kan avgöra vilka ordled som hör samman. Ett foge-s (eller avsaknaden av foge-s) kan sålunda signalera till ett analysprogram var gränsen mellan för- och efterled går.

Ibland blir det svårt att veta om den strukturella regeln ska tillämpas. Det inträffar till exempel då vi har problem att avgöra vart de olika ordleden hör. Delsing (2001) tar som exempel på detta upp en sammansättning av orden *kärnkraft + kraftverk*. Båda är väl inarbetade sammansättningar. Enligt regeln skulle vi få *kärnkrafts- kraftverk*. Detta förkortas dock till *kärnkraftverk*. Det kan se ut som om man utelämnat foge-s:et. Men det har man egentligen inte gjort, eftersom det som uppstått är en sammansättning med ett enledat förled *kärn-* och vid dessa har man aldrig foge-s.

Den fonetiska regeln är överordnad den strukturella. Detta innebär att man först kontrollerar om förledets slut har s- eller sje-ljud och i så fall utelämnas foge-s:et även om förledet är flerledat.

Sammansättningar av fyra eller fler morfem är inte särskilt vanliga. Dessa brukar istället skrivas som fraser. Det är dock teoretiskt möjligt att bilda oändligt långa sammansättningar. Reglerna är inte så komplicerade: Vid minst 2 led före huvudfogen följs den strukturella regeln, annars följs regeln för enkel sammansättning. Uppdelningen fortsätter sedan inom varje sammansättningsblock tills endast enkla led finns kvar.

Vid enkel sammansättning har vissa förled alltid foge-s, vissa aldrig, vissa varierar. Men det går inte att ställa upp regler för vad som gör att ett ord hör till en viss grupp. Delsing (2001:9) säger:

När ingen av reglerna gäller råder stor variation. Denna variation går tillbaka på de förhållanden som rådde i fornsvenskan före mitten av 1400-talet, då de flesta maskulina och neutrala ord hade genitivändelse på s medan feminina ord hade andra sätt att bilda genitiv. Foge-s:et utgår från genitiv-s:et och det är därför i princip bara gamla maskulina och neutrala ord som får foge-s när de uppträder som enkla förled.

Utöver dessa regler ser Liljestrand (1993) även en tendens till att foge-s ofta utelämnas vid tekniska termer samt en tendens till reducerad användning av foge-s rent generellt, d.v.s. även vid icketekniska nybildningar. I databasen finns flera exempel på ord som har former både med och utan s-inskott, t.ex. *tryckluftsbroms* – *tryckluftsbroms*, vilket skulle kunna stödja denna teori.

2.2 Sammansättningsanalys

Sammansättningar är karakteristiska för germanska språk. Engelskan har inte samma problem i NLP-sammanhang, eftersom man där främst använder sig av juxtaposition för sammansättningar. Vardera ordform slås då upp som vanligt i lexikonet och inget ord blir okänt. Däremot måste man ha med regler så att två led i en sammansättning inte översätts separat utan att två nomen i rad tolkas som en sammansättning.

Hutchins (1992) beskriver språk som t.ex. svenskan så här: ”Sammansättningar kan orsaka problem eftersom man ibland helt enkelt sätter ihop två ord, utan bindestreck eller mellanslag.” (min övers.) Svårigheten med att analysera sammansättningar ligger därför i segmenteringen och antalet möjliga segmenteringar. Det går inte att bara dela upp ord till höger och vänster utan man måste jämföra med lexikonet och se om de ingående delarna finns med där, så att programmet inte försöker att dela upp icke sammansatta ord.

Enligt Hutchins är sammansättningar ett syntaktiskt problem i engelskan, medan det i språk som svenska och tyska är ett morfologiskt problem. Han säger vidare att en hög andel sammansättningar kan ha inverkan på ett systems kapacitet. För ett sammansättningsrikt språk som svenskan är därför ett väl fungerande analysprogram extra viktigt.

2.2.1 Tidigare arbete

Hutchins (1992) sammanfattar sammansättningsdelen inom maskinöversättning bra i sitt uttalande ”In general, relatively little is written about morphological analysis in MT”. Det är mycket svårt att hitta information om vad som gjorts och för närvarande görs på detta område. Ett par av de projekt och undersökningar som utförts tas dock upp nedan.

Elzbieta Dura (1998) beskriver Karlssons (1992) metod för segmentering, inriktad speciellt på automatisk analys av produktiva svenska sammansättningar. I Karlssons program SWETWOL finns två huvudprinciper:

1. *Compound Elimination Principle* väljer i första hand de segmenteringsförslag som har minst antal sammansättningsgränser. Först genereras alla möjliga

kombinationer av ordparen. Därefter plockas alternativ bort. Färre uppdelningar = bättre. Dura tar upp exemplet *publikunderlag*, där analysen *publik+underlag* väljs hellre än analysen *pub+lik+under+lag*.

2. *Derivative Elimination Principle* väljer analyser med enkla led före analyser som innehåller avledda segment. Grundform betyder i detta fall att segmentet finns i lexikonet.

Båda principerna har som mål att "eliminate morphologically more complex readings in favour of simpler ones", d.v.s. att de mer komplexa analyserna väljs bort till förmån för enklare analyser.

I Karlssons undersökning täcktes hela 99,7 % av alla ord. SWETWOL tar dock inte med korta ord som *ö*, *ed*, *as* eller *os* i sammansättningar. På detta sätt missas sammansättningar där dessa ord ingår men övergenereringarna minskar, vilket kan vara en anledning till det bra resultatet.

Ett sätt att hitta sammansättningar på utan att använda sig av lexikon är att titta på s.k. *konsonantkluster*. Ett konsonantkluster är som namnet antyder en sträng av konsonanter. En av dem som studerat konsonantkluster som en metod för sammansättningsanalys är Benny Brodda. Han säger att "Vid en sammansättning uppstår (i bästa fall) en sådan konsonantkombination som omöjligen kan förekomma inne i ett enkelt ord. En sådan kombination kommer då både att utgöra en signal om att en sammansättning föreligger, och ger också ledtrådar om var sammansättningsgränsen går" (1979:29). Ett exempel på hur denna teori fungerar är klustret *m skl*, som inte kan förekomma i ett enkelt ord, utan är en indikering på att klustret kan vara en del av en sammansättning, t.ex. *bromskloss*.

Brodda har delat in dessa kluster i olika grupper och ställt upp en hierarki av successivt starkare medialklustermängder. Hierarkin har 6 nivåer och för varje steg ger klustren på just den nivån en tydligare signal om att det rör sig om en sammansättning. Grupperna 5 och 6 signalerar sålunda att det nästan alltid respektive alltid handlar om en sammansättning. Till den sista gruppen hör kluster som t.ex. *ntst* (kantsten) och *gkn* (stugknut).

Brodda ställer även upp en regel för var själva sammansättningsgränsen går. Den går ut på att om strängen M signalerat att det mellan vokalerna V1 och V2 i (V1 M V2) finns en sammansättningsgräns och M består av F(s)I (F = morfemfinalt kluster i förledet, I = morfeminitialt kluster i efterledet, s = foga-s) så letar man efter det längsta F respektive I som går att hitta.

Om man får flera olika segmenteringsförslag är det enligt Brodda bättre att välja det alternativ där F är så långt som möjligt och I så kort som möjligt (dock inte tomt). Detta går helt i linje med Karlssons teori som beskrivits tidigare i detta avsnitt. Genom att ta med statistik kan man ytterligare höja sannolikheten för att

välja rätt segmentering. Man väljer då den delning där produkten av de relativa frekvenserna för de ingående klustren når sitt maximum.

Ett sätt att minska antalet segmenteringar är enligt Dura att använda semantiska representationer. Detta alternativ blir dock väldigt kostsamt, varför Dura anser att Karlssons metod är bättre.

Det finns också sätt att plocka ut substantiv som *inte* är sammansättningar. Enstaviga ord kan t.ex. tas bort direkt.

2.2.2 UCP

Uppsala Chart Parser, UCP, är en chartparser som är skriven i LISP. Den har två delar – dels en parser som utför själva analysen, (UCP) och dels en språkbeskrivning (lexikon och grammatikregler) som parsern tolkar. Denna del har skrivits av Anna Sågvall Hein, som även har uppdaterat grammatikreglerna enligt de synpunkter som framkommit vid undersökningen.

UCP utför både morfologisk analys och syntaktisk analys. Resultatet av en meningsanalys är en hierarkisk beskrivning i form av en attribut-värdestruktur. För enstaka ord ges endast en morfologisk analys.

UCP:s grundläggande struktur är en chart. En chart består av ett antal noder med bågar emellan, som representerar delvis och helt igenkända delar av en fras. Om det går en båge från den första till den sista noden har hela meningen kunnat analyseras. Om det finns fler än en analys finns flera bågar. UCP:s uppbyggnad och innehåll beskrivs närmare i Sågvall Hein (1983), Ahrenberg (1984) och Dahllöf (1989).

UCP kan köras antingen på ett ord/en mening i taget eller på en hel fil. Den sistnämnda varianten användes för det här arbetet, vilket innebar att hela ordlistan kunde bearbetas på en gång.

Sammansättningsanalysen i UCP är i skrivande stund inte dokumenterad. En kortfattad beskrivning ges nedan.

Sammansättningsanalysen kan sägas bestå av två huvudkomponenter: Lexikon och regler. Processen går ut på att dela upp en ordform i tänkbara delar och söka efter dessa delar i lexikonet. Om samtliga delar finns upptagna uppfylls det första villkoret för en analys. Detta räcker dock inte om det inte finns regler som anger att delarna kan kombineras på det sätt som föreligger för den aktuella ordformen. Dessa regler finns definierade för varje böjningstyp (se avsnitt 3.1.1). Om både lexikon och regler godkänner den föreslagna uppdelningen presenteras denna som en analys i form av en attribut-värdestruktur. Det kan finnas flera olika uppdelningar av en ordform som uppfyller ovanstående krav. I dessa fall presenteras samtliga möjliga analyser för ordformen. Analysprogrammet kan ställas in

så att det visar en viss mängd attribut och döljer sådana som inte är relevanta för tillfället.

Sammansättningsanalysen i UCP har i ett första steg endast utvecklats för substantiv. Detta innebär i nuläget att endast sammansättningar som enbart består av substantiv tilldelas en sammansatt analys, t.ex. *fjäderarm*.

3 Metod

Själva arbetet kan delas in i olika delar:

- Material
- Regelframtagning
- Programmering
- Analys och utvärdering

Det första steget gick ut på att ta fram material att arbeta med. Det gällde dels en mer hanterlig version av innehållet i den svenska delen av MatsLex och dels de utdata i form av attribut-värdestrukturer dessa ord gavs från UCP. Därefter följde en gransknings- och kategoriseringsdel där jag utifrån exempel på attribut-värdestrukturerna bildade mönster och regler för vilken analys av flera möjliga som var den bästa för en viss typ av ordform. När sedan en algoritm tagits fram vidtog programmeringsarbetet. Det sista och mest omfattande steget inbegrep analys av dels UCP:s prestation och dels hur analysvalsprogrammet fungerade samt huruvida det går att utan manuell hjälp ta fram korrekt information om sammansättningar.

3.1 Utgångspunkt/material

Det material som arbetet utgick ifrån var MatsLex, den databas som skulle kompletteras. Informationen i databasen gavs till chartparsern UCP (se avsnitt 2.2.2) för sammansättningsanalys, vilket gav den information som sedan användes som indata till urvalsprogrammet.

3.1.1 Indata

MatsLex är ett domänspecifikt lexikon för maskinöversättning som har utvecklats inom MATS-projektet. Lexikonet utgörs av en relationell databas och innehåller lexikoningångar som används av det transferbaserade maskinöversättningsverktyget MULTRA⁶. Exempel på ingångar i databasen visas nedan.

```
fordon      fordon  BORD  fordon.nn  NOUN  NNNXIB  N  mt  95
fordonets  fordon  BORD  fordon.nn  NOUN  NNNSDG  N  mt  95
```

⁶MULTRA = Multilingual Support for Translation and Writing. Det är en prototyp för transferbaserad maskinöversättning av domänspecifika texter från svenska till engelska och tyska.

Fälten innehåller, från vänster till höger: ordform, stam, mönsterord, lemma, ordklass, morfosyntaktisk kod, stilkod, korpusbeteckning samt korpusår.

Mönsterordet anger ordets böjningsmönster. Genom att endast definiera varje böjningstyp en gång och sedan hänvisa till denna definition slipper man lagra informationen explicit för varje ordform. I exemplet ovan kan man se att *fordon* har samma böjningsändelser som *bord*. Mönsterordet ACCEPT används för lexikongångar som inte böjs och inte behöver översättas utan förs över direkt till målspråket.

Lemmabeteckningen utgörs av ordets grundform följt av dess ordklassbeteckning, t.ex. *bil.nn*. Vid ett tillfälle genomfördes en automatisk analys av orden i databasen och de ord som då analyserades som sammansättningar tilldelades ett lemma med sammansatt form bestående av de ingående ordens lemman åtskilda av plustecken, t.ex. *garanti.nn+avdelning.nn*.

Morfosyntaktisk och semantisk information lagras ofta i särdragsstrukturer. I MatsLex används istället sammanfattande koder, vilket gör att inga särdragsstrukturer behöver lagras explicit i databasen. Koder finns för att uttrycka morfosyntaktiska särdrag. Koder för lemmaspecifika särdrag, semantiska särdrag samt valensrelaterade särdrag definieras i filer.

Den *morfosyntaktiska koden* för ett substantiv innehåller information om, från vänster till höger, ordklass (NN), genus (Neutrum eller Utrum), numerus (Singular eller Plural), species (Indefinit eller Definit) samt kasus (Basic eller Genitiv). Underspecificering i form av ett 'X' används i de fall båda värdena passar in, som vid *fordon* ovan, vilket kan vara både singular och plural.

Utgångsmaterialet för undersökningen var sålunda en databas. För att kunna köra samtliga ord i databasen genom UCP togs en lista över de olika orden fram. Eftersom denna undersökning endast behandlar sammansättningar och den informationen är densamma för samtliga former av ett paradigm plockades endast grundformen ut. Detta gjordes genom att samtliga lemman i databasen plockades ut, lemmabeteckningen kapades av och dubletter togs bort.

Vissa ord som inte var relevanta för undersökningen raderades:

+ Fraser, t.ex.

sätta tillbaka
stå stilla
tid och kraft
till höger och vänster

+ Partikelverb, t.ex.

pressa fast
rikta in
rulla ut

+ Diverse tecken, t.ex.

Ø
xxxx
d!o
+/-10

+ Konstruktioner av ord som inte är sammansättningar, t.ex.

sv=svenska
r/min
miles/+/-
ABS/EDC
CAN-/färdskrivarsignal
Packning/O-ring

+ Bindestreckskonstruktioner som inte är sammansättningar, t.ex.

om-10a

+ Konstruktioner med siffror, t.ex.

om1asv
ft3
ZF809
0.01x

+ Rena sifferuttryck, t.ex.

4
4.78

+ Ord som har mönsterordet ACCEPT, t.ex.

A+A
BENDIX
BF
traction
training

Sammanlagt plockades drygt 4300 lemmar bort.

Totalt kom indatalistan att bestå av 17557 grundformer. Större delen av materialet utgörs av substantiv (ca 80 %, att jämföras med adjektiv och verb, vilka båda ligger kring 8,5 %).

Ca 5000 av orden var markerade som sammansättning sedan tidigare (mer om detta under nästa punkt). Det faktiska antalet sammansättningar såg dock ut att vara större. Jag plockade ut ord som börjar på någon av fyra slumpvis utvalda

bokstäver (a, i, d, och t) och gick igenom dem manuellt med avseende på antalet sammansättningar och antalet led för att bilda mig en uppfattning om hur stor del av materialet som utgjordes av sammansättningar. Resultatet visade på 70-75 % sammansättningar, något lägre för ord som börjar på vokaler (där finns många partikelsammansatta ord, t.ex. *inskriva*, *avbilda*, *omvandla*). Ca 15 % av sammansättningarna hade fler än två led.

3.1.2 UCP-analyser

UCP är ickedeterministisk, vilket innebär att alla möjliga segmenteringar och analyser genereras för varje ord. UCP kan antingen utföra enbart vanlig böjningsanalys eller dessutom utföra sammansättningsanalys.

Sammansättningsanalysen kan ställas in så att endast två led tas ut, ett förled och ett efterled. Denna inställning användes i ett inledande skede, men tvåledsindelningen klarade inte att hantera ordfogningar som t.ex. *trevägsnippel* eller vissa längre sammansättningar med fler än två led. Inställningarna ändrades därför så att ett ord kunde delas upp i fler delar.

Den lista som skapats utifrån orden i databasen gavs som indata till UCP. Varje ord i listan tilldelades där en eller flera analyser. En analys presenteras som en attribut-värdestruktur, d.v.s. en oordnad mängd attribut-värdepar som ger information om lemma, ordklass etc.

Exempel på en analys⁷ av ett (icke sammansatt) ord:

```
garanti :
(* = (START = 1
      END = 9
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = garanti.nn
      STEM = garanti
      DIC.STEM = garanti))
```

För ett enkelt ord är värdet på attributet COMPOUND -. För ett ord som av UCP analyserats som en sammansättning blir istället värdet +. I detta fall utgörs lemmat av de ingående ledens lemman sammanbundna med plustecken. Stammen anges på samma sätt. Eftersom UCP även utför vanlig böjningsanalys får de sammansatta orden minst två analyser, dels en eller flera sammansatta analyser och dels

⁷Viss information som inte var relevant för underökningen har undertryckts.

en enkel analys, under förutsättning att sammansättningens stam finns inlagd i sin helhet i lexikonet.

Exempel på ett sammansatt ord:

```
garantiavdelning :
(* = (START = 1
      END = 18
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = garanti.nn+avdelning.nn
      DIC.STEM = garanti+avdelning
      STEM = garantiavdelning))
(* = (START = 1
      END = 18
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = garantiavdelning.nn
      STEM = garantiavdelning
      DIC.STEM = garantiavdelning))
```

En del sammansättningar markerades som sådana redan då de lades in i lexikonet, genom att man i lemmat angav den sammansatta formen med ett plustecken. UCP behandlar denna stam likadant som den sammansatta stammen *garantikontroll.nn* i föregående exempel och en ordform kan därför få två analyser med sammansatt lemma, men där endast den ena analyserats som sammansättning av UCP:

```
garantirutin :
(* = (START = 1
      END = 14
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
```

```

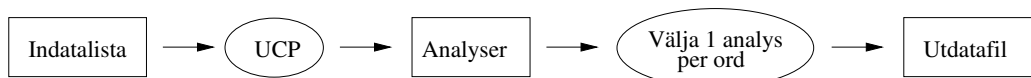
        LEM = garanti.nn+rutin.nn
        DIC.STEM = garanti+rutin
        STEM = garantirutin))
(* = (START = 1
      END = 14
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = garanti.nn+rutin.nn
      STEM = garantirutin
      DIC.STEM = garantirutin))

```

Det förekom även att en sammansättning endast fick en enkel analys. I dessa fall har UCP misslyckats med att analysera ordet som en sammansättning. Detta tas upp under utvärderingspunkten.

3.2 Algoritm för analysvalsprogrammet

Indatan i form av en lista av ord från databasen har alltså i ett första steg försetts med UCP-analyser för varje ord. I de fall där UCP tilldelar ett ord mer än en sammansatt analys, måste dock den korrekta uppdelningen av varje ord hittas.



Figur 1: Analysprocessen steg för steg

Nedan redovisas de regler som ställts upp för val av rätt analys för varje ord. I reglerna har ingen hänsyn kunnat tas till betoning eller semantisk information.

Analysvalsprogrammet utgår ifrån de UCP-analyser som finns för varje ord. Ett ord i taget med tillhörande analyser läses in i programmet. Endast de sammansatta orden är dock relevanta för undersökningen. Indatalistan innehåller i stort sett samtliga svenska grundformer ur MatsLex och en del av dem är vanliga, enkla former. Enkla analyser ignoreras vid inläsningen till analysvalsprogrammet.

När ett ord med tillhörande analyser läses in föreligger en av tre möjligheter:

1. Ingen sammansatt analys finns. Ordet har endast tilldelats en enkel analys och är därför inte relevant för programmet, som går vidare direkt till nästa ord.
2. Endast en analys finns. Ordet har entydigt kunnat delas upp i sina ingående delar. Denna analys väljs och skickas tillsammans med ordformen till utdatafilen.
3. Flera analyser finns. Programmet måste med hjälp av regler välja den bästa.

De två första punkterna medför inga problem. Den tredje kräver en noggrann utformning av regler som anger vilken uppdelning som är bäst för ett visst ord. Ett antal ord⁸ med tillhörande analyser studerades i syfte att se vilka olika typer av segmenteringar som kan föreligga. Denna undersökning användes som grund då reglerna skapades.

I vissa fall behövs endast enkla regler för att plocka ut den bästa analysen, men vissa ord kräver mer specifika regler. Reglerna formulerades därför enligt en ”trappa”, där programmet fortsätter ett steg i taget tills dess att endast en analys finns kvar. Då avbryts stegandet och analysen väljs ut som den bästa för den aktuella ordformen.

Trappmodellen innehåller följande steg.

1. Jämför antalet led i lemmat – välj minst antal. Lika många led = gå vidare till nästa steg.
2. Jämför första delen i tekniska stammen – välj längst. Lika långa = gå vidare till nästa steg.
3. Jämför första delen i lemmat – välj längst. Lika långa = gå vidare till nästa steg. Om fortfarande analyser med fler än 2 led kvarstår, hantera manuellt.
4. Är det ett ord av typen stång-tång, där skillnaden i analyserna består i att den ena analysen tolkar ett -s- som ett foga-s, medan den andra tolkar det som första bokstaven i efterföljande led (se nedan i detta avsnitt)? Välj stång-analysen förutom när -s- föregås av avledningssuffixen *-ing*, *-tion* eller *-het*. Om ordet inte är av denna typ = gå vidare till nästa steg.
5. Jämför andra delen av lemmat – välj längst. Lika långa = ge varningsmeddelande, men lägg ändå vidare en av analyserna för utskrift till utdatafilen.

⁸De ord som studerades för att formulera reglerna började på d eller t (slumpvis utvalda bokstäver), sammanlagt ca 1500 ord.

De första två stegen i modellen följer Karlssons metod som går ut på att i första hand välja den analys som brutits ner i minst antal led och att i andra hand (om samma antal led föreligger) välja den analys som har kortast förled (se avsnitt 2.2.1). Trots att metoden är enkel fungerar den mycket bra. Dessa två steg räckte dock inte till för materialet i den här undersökningen.

Det förekom nämligen att analyser hade lika långt förled. Exempel på detta är de analyser som UCP ger av ordet *droppform*:

```

droppform :
(* = (START = 1
      END = 11
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = droppe.nn+form2.nn
      DIC.STEM = dropp+form
      STEM = droppform))
(* = (START = 1
      END = 11
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = dropp.nn+form2.nn
      DIC.STEM = dropp+form
      STEM = droppform))
(* = (START = 1
      END = 11
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = droppe.nn+form2.nn
      STEM = droppform
      DIC.STEM = droppform))

```

Stammen är densamma, men lemnarna skiljer sig åt. Det är viktigt att välja

rätt analys, eftersom den semantiska innebörden kan skilja sig väsentligt mellan de olika alternativen. I exemplet ovan bör den första analysen väljas ut, då det rör sig om formen hos en droppe, inte hos ett dropp. För att komma åt ett fall som detta måste man gå in och titta på lemmats form. Man kan välja att plocka ut den analys som har längst första del i lemmat, eller kortast första del. Det första fallet ger en bra analys av exemplet ovan, men misslyckas med att ta ut den bästa analysen av ord på *tank-*, där lemman med *tanke.nn* då väljs före de korrekta med *tank.nn*. Detta problem går inte att lösa på automatisk väg, eftersom semantisk information inte finns. I materialet fanns dock flest ord där den längre varianten var den korrekta. Därför valdes denna regel och övriga ord fick korrigeras manuellt.

Efter det tredje steget i trappmodellen var det svårt att utforma några regler för analyser med fler än två led, varför eventuellt förekommande sådana skrivs ut till en separat fil för manuell bearbetning.

Det fjärde steget tar hand om en grupp av ord som kom att bli särskilt besvärliga. Det gäller de ord där ett medialt s kan vara antingen ett foge-s eller ett inledande s i efterföljande led. Ett exempel är ordet *gardinstång*, som fick följande analyser.

```
gardinstång :
(* = (START = 1
      END = 13
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = gardin.nn+stång.nn
      DIC.STEM = gardin+stång
      STEM = gardinstång))
(* = (START = 1
      END = 13
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = gardin.nn+tång.nn
      DIC.STEM = gardin+tång
      STEM = gardinstång))
(* = (START = 1
      END = 13
```

```

WORD.CAT = NOUN
GENDER = UTR
FORM = INDEF
NUMB = SING
CASE = BASIC
COMPOUND = -
LEM = gardin.nn+stång.nn
STEM = gardinstång
DIC.STEM = gardinstång))

```

I detta fall rör det sig om en stång. I de fall det skulle vara en tång föregicks foge-s påfallande ofta av en avledning, som t.ex. i *plomberingstång*. Sålunda formulerades en regel som, i de fall ett analyspar av denna typ hittats, i första hand valde den variant där s:et hör till nästa led, men inte då s föregicks av någon av ändelserna *-ing*, *-tion* eller *-het*, samtliga relativt säkra utpekare av foge-s. Om det efter dessa ändelser endast finns ett s, måste det vara ett foge-s och kan alltså inte höra till nästa led.

Denna regel klarar inte att hantera vissa ord, t.ex. *ringsko*. En felaktig uppdelning av just detta ord kan undvikas genom en regel som förbjuder avledningar där stammen endast består av konsonanter, men alla ord av denna typ bör ändå kontrolleras och vid behov korrigeras manuellt. För att lättare kunna veta vilka ord som man bör se över finns en utskriftsfunktion i programmet där samtliga ord av denna typ skrivs ut till en särskild kontrollfil.

I lexikonet fanns även ord som fortfarande hade fler än en analys kvar, trots att de genomgått utrensningen i stegen innan. Ett exempel på ett sådant ord är *arbetscykel*, där enda skillnaden mellan analyserna är att de har olika böjningstyp och att det ena efterledet är försett med lemmanummer.

```

arbetscykel :
(* = (START = 1
      END = 13
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = arbete.nn+cykel.nn
      DIC.STEM = arbet+cyk
      STEM = arbetscykel))
(* = (START = 1
      END = 13
      COMPOUND = +

```

```

WORD.CAT = NOUN
GENDER = UTR
FORM = INDEF
NUMB = SING
CASE = BASIC
LEM = arbete.nn+cykel2.nn
DIC.STEM = arbet+cyk
STEM = arbetscykel))
(* = (START = 1
      END = 13
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = arbete.nn+cykel1.nn
      STEM = arbetscykel
      DIC.STEM = arbetscyk))

```

Att endast en av analyserna är försedd med lemmanummer beror på att man vid manuell inläggning av ett ord i lexikonet har försett ordet med lemmanummer för att skilja det från ett homonymt ord med annan böjning som redan tidigare fanns i lexikonet, men glömt att lägga in lemmanummer även för det befintliga ordet. För att hitta denna skillnad jämförs i det femte steget analysernas efterled. Eftersom varianten med lemmanummer är mer specifik väljs analysen med längst efterled ut.

Det finns dock fall där båda analysernas efterled är försedda med lemmanummer, t.ex. *beläggprov* nedan.

```

beläggprov :
(* = (START = 1
      END = 12
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = belägg.nn+prov2.nn
      DIC.STEM = belägg+prov
      STEM = beläggprov))
(* = (START = 1
      END = 12

```

```

COMPOUND = +
WORD.CAT = NOUN
GENDER = NEUTR
FORM = INDEF
NUMB = NIL
CASE = BASIC
LEM = belägg.nn+prov1.nn
DIC.STEM = belägg+prov
STEM = beläggprov))
(* = (START = 1
      END = 12
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      COMPOUND = -
      LEM = belägg.nn+prov1.nn
      STEM = beläggprov
      DIC.STEM = beläggprov))

```

Samma sak gäller givetvis även ord där lemmanumren finns i det första ledet. Det finns inte något sätt att automatiskt hantera lemmanummer utan att lägga till avancerad semantisk information. Därför väljer programmet helt enkelt en av analyserna och skickar ett meddelande till kontrollfilen om att detta ord måste kontrolleras manuellt.

När programmet väl valt ut den bästa analysen är det enkelt att plocka ut huvudordet eftersom huvudordet alltid är det sista ledet. Utdata från programmet blir en fil där en analys för varje ord i databasen finns med. Analyserna är sorterade efter huvudled för att underlätta vid jämförelse mellan ord som har samma huvudled. Dessutom skapas en fil med de problematiska ord som måste kontrolleras manuellt.

En aspekt som i ett inledande skede togs med vid framtagningen av reglerna var användandet av foge-s i sammansättningar med sammansatt förled. En regel skulle kunna vara att ett ord med tre led, där ett foge-s finns mellan led två och tre, ska delas efter foge-s. Om inget foge-s finns ska ordet istället delas efter det första ledet. Denna regel skulle placeras parallellt med regel två ovan, efter en kontroll om ordet har tre led eller inte, och skulle kunna förhindra att ett ord med sammansatt efterled som t.ex. *stallvarvtal* tilldelas analysen *stallvarv.nn+tal.nn*, vilket sker med de ovan presenterade reglerna. Det finns dock flera anledningar till att reglerna inte kom att formuleras baserat på en eventuell förekomst av foge-s. En anledning är att vissa ord som i nuläget tilldelas en korrekt analys skulle bli fel.

Ett exempel på detta är ordet *bromspedalgivare*, som har ett sammansatt förled, men inget foge-s. Fel skulle det även bli vid ord med sammansatt efterled, men där det finns ett foge-s mellan led två och tre eftersom det krävs i vanlig enkel sammansättning mellan leden i efterledet. Dessa ord får en korrekt uppdelning enligt de existerande reglerna, men skulle bli fel om en regel som tittade på foge-s lades in (eftersom denna regel skulle ersätta nuvarande regel 2 för ordet i fråga). Fler ord skulle på detta sätt få en felaktig analys än med de regler som senare valdes.

En annan, mycket viktig, anledning till att inte blanda in foge-s i reglerna är att orden i databasen inte alltid följer de traditionella reglerna för sammansättning utan snarare stöder Liljestrands teori om att användningen av foge-s minskar i teknisk text (se avsnitt 2.1.3). Tex. finns i databasen ordet *kontrollamppanel*. Många ord av denna typ finns upptagna i databasen både med och utan foge-s. Med detta som grund kan inte foge-s användas som bas för regler.

Schematiskt kan de olika stegen i ”regeltrappan” ritas upp så här:

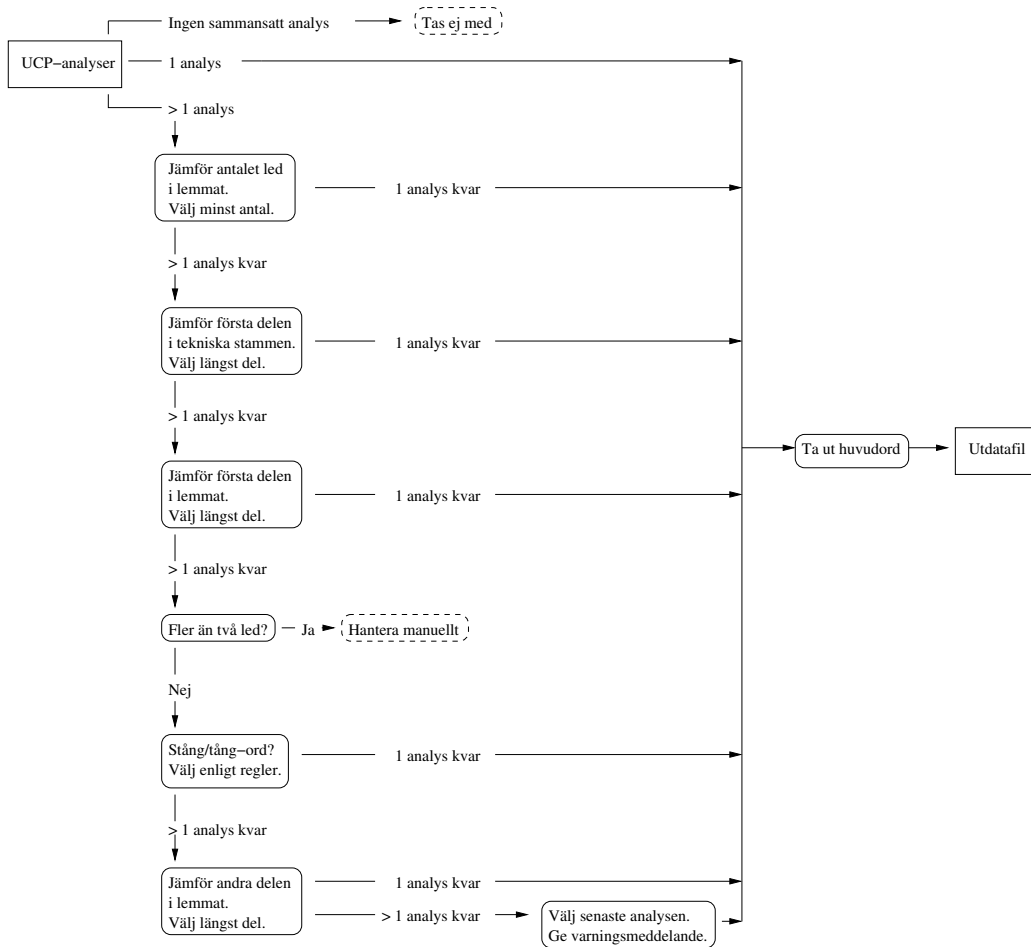
3.3 Implementation/genomförande

Programmet börjar med att läsa in en ordform i taget med tillhörande analyser. Antalet analyser varierar från ordform till ordform, men ett enkelt sätt att få med alla analyser för en viss ordform är att fortsätta att läsa in analyser tills dess att nästa ordform påträffas.

I den särdragstruktur som varje analys består av finns följande attribut.

```
START
END
COMPOUND
WORD.CAT
GENDER
FORM
NUMB
CASE
LEM
DIC.STEM
STEM
```

Av dessa är START, END, WORD.CAT, GENDER och STEM inte relevanta för analysvalsprogrammet. När programmet läser in en rad där någon av dessa förekommer görs ingenting. Vid övriga särdrag sker följande: Värdena på COMPOUND, GENDER, FORM, NUMB och CASE används för att tillåta respektive förbjuda att en viss analys läses in och knyts till en ordform. När en rad innehållande attributet COMPOUND med värdet + hittas rör det sig om en sammansatt analys, vilken ska läsas in.



Figur 2: Schema över analysvalsprogrammets olika steg

I programmet finns en kontroll som sorterar bort analyser där sista ledet står i genitivkasus, plural eller bestämd form, eftersom endast grundformerna tagits med ur den lexikala databasen. På så vis ges tyvärr ingen sammansatt analys för ord som alltid står i plural, t.ex. *skyddskläder*, men i gengäld förhindras ett flertal felaktiga konstruktioner. Utan denna regel kan t.ex. ordet *kolos* segmenteras i delarna *ko+lo*. De ord som alltid står i plural har antingen mönsterordet PRESTANDA eller KLÄDER. Detta gör det lätt att manuellt kontrollera dessa i efterhand, genom att en utskrift görs av ord som har dessa mönsterord till en separat kontrollfil.

Algoritmen implementerades i Perl. (Nedan visas pseudokod, programmet finns i sin helhet i bilaga A)

1. Öppna indatafilen (UCP:s analyser).
2. Läs in en rad i taget.
3. Arbeta med en ordform i taget och lägg in den samt dess analyser i en array.
 - 3a. Att en analys ska ”läsas in” innebär att det till ordformen ska knytas ett lemma och en stam för varje sammansatt analys som ordformen tilldelats. Villkoret för att en analys ska läsas in är att den analyserats som en sammansättning av UCP, d.v.s. att attributet COMPOUND har värdet +. En siffra läggs in efter lemmat för att kunna hålla reda på vilka delar som hör till samma analys. Samma sak gäller för stammen.
 - 3b. När ny ordform hittas, innebär det att inga fler analyser finns för föregående ordform. Bearbeta föregående ordform (välj ut den analys som är bäst) och fortsätt sedan med den nya ordformen.
4. Bearbeta analysarray:
 - 4a. Inga sammansatta analyser har lästs in för ordformen. Inget görs. Ordet tas inte med i utdatafilen.
 - 4b. En analys finns för ordformen. Denna läggs vidare för utskrift.
 - 4c. Fler än en analys finns för ordformen. Analyserna jämförs.
5. Jämföra analyser:
 - 5a. Räkna antalet led i analysen. Minst antal = välj. Lika många = gå vidare.
 - 5b. Räkna bokstäver i stammens första led. Längst första led = väljs ut. Lika långa = gå vidare.
 - 5c. Räkna bokstäver i lemmats första led. Längst första led = väljs ut. Lika långa första led = gå vidare.
(Ord vars analyser har fler än två led förs åt sidan för manuell bearbetning)
 - 5d. Kontrollera ”stång/tång”. Om fortfarande inget kunnat väljas, gå vidare.
 - 5e. Räkna bokstäver i efterledet. Längst efterled = väljs. Lika långa = välj det senaste och skriv ut meddelande om manuell kontroll.
6. Plocka ut huvudord: Huvudordet är alltid det sista ledet. Plocka ut det sista ledet i lemmat och ta bort lemmabeteckning samt eventuellt lemmanummer.

7. Skriv ut analyser: När samtliga ords analyser gått igenom och den bästa analysen för varje ord plockats ut skrivs dessa ut till en fil, sorterade efter huvudled, på formatet

```
nivåanpassning :  
LEM = nivå.nn+anpassning.nn  
Efterled = anpassning
```

3.4 Manuellt arbete

Vissa saker som UCP inte klarade av (dessa tas upp i nästa avsnitt) gjorde att många ord inte fick någon sammansatt analys och därför sorterades ut tillsammans med de icke sammansatta orden av analysvalsprogrammet. Listan över ord som programmet plockade bort måste därför gås igenom för att se vilka bland dem som var sammansättningar och egentligen skulle tas med. Dessa knappt 3000 ord delades sedan upp manuellt och lades in i utdatafilen.

Den manuella uppdelningen resulterade i ett relativt stort antal enkla ord som inte fanns med i lexikonet. Dessa lades under en projektanställning in i databasen av Kristina Ohlander. I vissa fall behövde även nya mönsterord skapas. Det gällde ord som inte kan stå för sig själv utan endast som efterled i en sammansättning, t.ex. *polig* och *stegs* i *10-polig* och *2-stegs*. Flera förekomster fanns av dessa båda typer och därför skapades mönsterorden POLIG och STEGS.

Utdatafilen som blev resultatet av analysvalsprogrammet måste också gås igenom manuellt eftersom det fanns undantag till de flesta av reglerna i programmet. De faktorer som måste kontrolleras finns beskrivna i avsnitt 6.

4 Resultat

4.1 UCP

Av de 17557 ordformer som fanns i den lexikala databasen gav UCP 8662 st minst en sammansatt analys. De flesta av ordformerna tilldelades bara en sammansatt analys, men det fanns en del ordformer som fick betydligt fler analyser. I tabell 1 visas hur orden fördelar sig på antalet analyser.

Antal analyser	Antal ord
1 analys	6712 ord
2 analyser	1339 ord
3 analyser	425 ord
4 analyser	104 ord
5 analyser	49 ord
6 analyser	17 ord
7 analyser	9 ord
8 analyser	5 ord
9 analyser	1 ord
10 analyser	1 ord

Tabell 1: Antalet ord fördelade på det antal analyser de tilldelades.

Det ord som fick 10 sammansatta analyser var *varvtalsstegringsprov*, som delades upp i olika kombinationer av orden *varv.nn*, *tal.nn*, *steg.nn*, *stege.nn*, *ring.nn*, *prov1.nn* och *prov2.nn*.

UCP klarade dock inte att hitta alla sammansättningar i materialet. 2955 sammansatta ord fick endast enkla analyser. Knappt 2500 av dessa har delats upp manuellt och lagts in i utdatafilen. Övriga 500 ord var svårupplade och i vissa fall var det svårt att motivera en uppdelning. En lista över dessa ord har levererats till MATS-projektet tillsammans med den färdiga utdatafilen.

4.2 Analysvalsprogrammet

Utdata från analysvalsprogrammet är en fil med en analys för varje sammansättning. Varje analys presenteras på följande format.

```
nivåanpassning :  
LEM = nivå.nn+anpassning.nn  
Efterled = anpassning
```

När programmet hade körts med UCP-analyserna som indata hade 8662 ordformer med tillhörande analyser lagts in i filen. Ca 400 av dessa korrigerades vid

en manuell genomgång av filen. Därefter lades de 2500 manuellt uppdelade orden som nämns i föregående avsnitt till. Den slutgiltiga utdatafilen innehåller 11109 ordformer, ca 63 % av ordformerna i databasen.

I tabell 2 visas fördelningen av korrekta respektive felaktigt utplockade analyser. De felaktiga analyserna är även uppdelade i olika feltyper. Dessa beskrivs närmare i utvärderingsavsnittet (avsnitt 6)

Korrekta analyser	8241
Felaktiga analyser, UCP-baserade fel	335
Annan ordklass än substantiv	277
Ingående led saknas i lexikon	29
Övergenerering	29
Felaktiga analyser, urvalsprogrammet	86
Sammansatt efterled	4
Ordformen är av typen stång/tång	6
Rätt analys har kortast förled	11
Ordet står i plural	15
Fel lemmanummer	50

Tabell 2: Antalet korrekta resp. felaktiga analyser fördelade på felorsak.

5 Utvärdering av UCP

En typ av utvärdering vore att jämföra UCP med ett annat program för sammansättningsanalys. I undersökningen har dock endast ett analysprogram studerats. Istället görs i detta avsnitt en kvalitativ utvärdering av UCP, d.v.s. vilka problem som finns och vad programmet klarar att hantera respektive misslyckas med.

Utvärderingen visar att två huvudsakliga typer av fel förekom:

1. Enkla ord tilldelades en sammansatt analys eftersom de består av delar som finns i lexikonet, t.ex. *koppar* som analyserades som *kopp.nn + ar.nn*.
2. Sammansatta ord tilldelades endast enkla analyser eller endast felaktiga sammansatta analyser. Flera undergrupper finns till denna punkt:
 - a) UCP:s sammansättningsanalys är än så länge endast utvecklad för substantiv. Därför får sammansättningar som innehåller led från någon annan ordklass, t.ex. *bladffjädrad*, inte någon korrekt analys. Ordet *styregenskap*, där inte analysen *styra.vb+egenskap.nn* hittas, får istället den mindre lyckade analysen *styre.nn+gen.nn+kap.nn*. Sammanlagt påverkade detta 1345 ord, mer än 10 % av alla sammansättningar. Om UCP utvecklas till att även omfatta adjektiv och verb, skulle en stor del av dessa ord kunna hanteras.
 - b) Delar av vissa ord fanns inte i lexikonet. Ett exempel på detta är ordet *bomullstrasa*, där *trasa* fanns i lexikonet, men inte *bomull*. 700 sammansättningar missades p.g.a. detta. Ett flertal av de orddelar som saknades har i efterhand lagts in i databasen för att komplettera denna.
 - c) I den lexikaliska databasen förekom en del fel, t.ex. stavfel eller felaktigt lemma. P.g.a. detta kunde inte delarna i vissa sammansättningar hittas. Ett exempel är ordet *vinkelmätinstrument*, som i databasen var felstavat *vin-keltmätinstrument*, vilket innebär att de ingående delarna inte kunde identifieras. Ett knappt hundratal sammansättningar missades på detta sätt. De felaktigheter som hittades rapporterades och har korrigerats i databasen.
 - d) 17 ord skulle behöva en ny stam. T.ex. har ordet *trumma* stammen *trumm*, vilket innebär att sammansättningar som börjar på *trum-* inte kan hanteras. Samma sak gällde för *flamma* och *stomme*, *klamma* och *timme* samt verbet *trimma*, som hade stammen *trimma*. Dessa felaktigheter har rapporterats.
 - e) 170 ord skulle behöva byta mönsterord, eftersom de bildar sammansättning annorlunda än sitt nuvarande mönsterord. Ett exempel på detta är ordet *ansvar* som har mönsterordet BRUS. Sammansättningen *ansvarsområde* kunde inte segmenteras eftersom reglen för BRUS inte tillåter foge-s.

Att det befintliga mönsterordet inte använder foge-s i sammansättning var den vanligaste feltypen, men även mer ovanliga former fanns, som t.ex. *ti-devarv*.

Ovanstående punkter beror alla på fel i den lexikala databasen eller definitionerna av mönsterorden. De grammatiska reglerna kan inte påverka dessa punkter. Nedanstående punkter har dock med UCP:s regler för sammansättningar att göra.

f) Vissa mönsterord var inte alls definierade för att kunna ingå i sammansättningar. Så var t.ex. fallet med mönsterorden PRESTANDA och HUVUD . SC. Ord med dessa mönsterord kunde enligt reglerna inte stå som efterled i en sammansättning. Reglerna för mönsterorden ändrades och det femtiotal ord som drabbades inledningsvis kunde därefter analyseras utan problem. Även definitionerna av mönsterorden BUSS, SATS och HUS måste ändras för att ord med dessa mönsterord skulle kunna ingå i sammansättningar. Dessa fel upptäcktes och korrigerades dock tidigt under undersökningen och dessa ord, sammanlagt ett par hundra ord, finns därför inte med i det antal sammansättningar som UCP inte klarade att analysera.

g) 618 ord har bindestreck, t.ex. *cd-skiva*. En sammansättningsregel som hanterar detta fenomen skulle troligen avhjälpa de flesta av dessa ord som inte tilldelades någon sammansatt analys i den här undersökningen.

h) I de sammansättningar där de två avslutande bokstäverna i förledet och den inledande bokstaven i efterledet är identiska skrivs de tre bokstäverna ihop till två, t.ex. *buss+sida* → *bussida*. Alla de 35 ord i materialet som innehöll en ihopskrivning av konsonanter skulle troligen kunna hanteras om en regel för trippelkonsonanter infördes.

Kategoriseringen av fel har gjorts så generellt som möjligt. Därför har ord som inte klaras eftersom de innehåller ett led av annan ordklass än substantiv förts till denna grupp, även om det kan hända att vissa av dessa ord också skulle passa in på någon av de mindre vanliga grupperna. Om ett av ovanstående problem åtgärdas innebär det därför inte automatiskt att samtliga ord som tidigare missades p.g.a. detta problem kommer att klaras av.

Anna Sågvall Hein och Bengt Dahlqvist var till stor hjälp vid utvärderingsarbetet och rättade till regler respektive lexikon. Under detta arbete förbättrades både lexikon och sammansättningsgrammatik (UCP).

6 Utvärdering av analysvalsprogrammet

Även om analysvalsprogrammet klarar av att hantera större delen av orden kan inte resultatet användas direkt utan manuell genomgång. Eftersom UCP inte ger ett helt tillförlitligt resultat, t.ex. inte analyserar alla sammansättningar som sådana eller analyserar enkla ord som sammansättningar, samt att det finns undantag till reglerna i analysvalsprogrammet, så finns det ord som analysvalsprogrammet missar att få fram korrekta analyser för.

Programmet förutsätter att det finns åtminstone en korrekt analys för varje ord, samt att inget enkelt ord kunnat analyseras som en sammansättning. Därför kan inte resultatet från analysvalsprogrammet läggas in direkt i databasen. Först måste hela utdatafilen gås igenom manuellt.

Den resulterande utdatafilen innehåller analyser av tre olika slag:

1. Den korrekta analysen har plockats ut. 8241 av 8662 analyser var helt korrekta.
2. UCP ger endast felaktiga analyser och programmet väljer en av dessa. Tre undergrupper finns till denna punkt:
 - a) UCP är endast utvecklad för att hantera substantiv. För sammansättningar där en av delarna har annan ordklass missas den korrekta analysen. I de fall där sammansättningen även får en sammansatt analys med enbart substantiviska delar kommer denna att väljas av analysvalsprogrammet. Så är t.ex. fallet med *tippanordning* som endast tilldelades den sammansatta analysen *tipp.nn+anordning.nn*, men där varianten med verbet *tippa.vb* är den korrekta. 277 ord fick fel analys p.g.a. detta.
 - b) Vissa ord finns inte med i lexikonet. Om en sammansättning där ett sådant ord ingår istället kan delas upp på ett annat (felaktigt) sätt kommer denna analys med till analysvalsprogrammet. Till denna punkt hör även de sammansättningar som har en felaktig form i databasen och därför får felaktiga analyser av UCP. Ett exempel på detta är *B-bilar*, som får analysen *B-bil.nn+ar.nn*. Fel av denna typ har korrigerats både i utdatafilen och i lexikonet.
 - c) UCP delar upp ett enkelt ord och denna analys väljs av analysvalsprogrammet. En kontroll har dock lagts in av huvudledet så att de huvudled som är kortare än tre bokstäver ger ett varningsmeddelande. Detta görs eftersom det oftast är de kortaste huvudleden som är fel (eftersom de är en del av ett enkelt ord som sällan är väldigt långa). På så vis plockas bl.a. analyserna *påstå* → *påse.nn+tå.nn* och *exemplar* → *exempel.nn+ar.nn* bort. Ett trettiotal enkla ord slank igenom från UCP, många av dessa ord på *ing*, t.ex. *ändring* som fick analysen *ände.nn+ring.nn*.

3. UCP ger minst en semantiskt korrekt analys, men programmet misslyckas med att plocka ut den. Flera undergrupper finns till denna punkt:
- a) Programmet väljer en analys med sammansatt förled fast det ska vara sammansatt efterled. Regeln säger att analysen med längst första led väljs i första hand, vilket innebär att *stallvarvtal* delas upp i delarna *stallvarv.nn+tal.nn* istället för den korrekta uppdelningen *stall.nn+varvtal.nn*. 5 ord påverkas av denna regel.
 - b) Programmet väljer fel analys vid ”stång/tång”-ord (se avsnitt 3.2). De ord där s är ett foga-s men inte föregås av någon av de ändelser som specificeras i regeln väljs felaktigt den analys där s:et hör till efterledet, som i *framkantslutning* → *framkant.nn+slutning.nn*. I materialet fanns även ett ord som drabbades på motsatt sätt. *ing* i ordet *ringsko* tolkades som en ändelse och ordet fick därför analysen *ring.nn+ko.nn* istället för den korrekta *ring.nn+sko.nn*. Totalt påverkades 8 ord av detta.
 - c) Programmet väljer fel analys vid jämförelse mellan förledens längd. Reglerna säger att längst förled ska väljas, men i några fall är det kortare förledet det rätta. Ett exempel är ordet *tankarmatur*, där analysen *tanke.nn+armatur.nn* väljs istället för den korrekta *tank.nn+armatur.nn*.
 - d) Den rätta analysen väljs inte eftersom ordet står i plural. Ett exempel på ett sådant ord är *skyddskläder*. En regel i analysvalsprogrammet förbjuder efterled som står i plural och detta ord tilldelas därför ingen analys alls, utan förs ut för manuell bearbetning. De ord som berörs av detta har mönsterorden KLÄDER och PRESTANDA, sammanlagt något tiotal ord.
 - e) Programmet väljer fel analys vid jämförelse mellan två analyser med lika långa led. Exempel på detta är de analyser där endast lemmanumret är olika. Programmet har inte tillgång till semantisk information och väljer därför fel analys ibland.

De hypoteser som ställdes upp vid arbetets början (se avsnitt 1.3) kan nu utvärderas:

- Antagandet att antalet analyser för varje sammansättning skulle vara relativt få stämmer bra. Endast 7 % av de sammansatta orden fick tre eller fler analyser.
- Väldigt många av orden i databasen var sammansättningar, närmare bestämt 63 %. Detta stämmer mycket väl överens med undersökningar som pekar på att sammansättningar är mycket vanliga i teknisk text.
- UCP klarade av förvånansvärt många typer av sammansättningar. En stor begränsning är dock det faktum att sammansättningsdelen endast är utvecklad för substantiv. Lexikonet innehöll även för få enkla ordformer för att alla sammansättningar skulle ha en teoretisk chans att få en sammansatt analys.

- Övergenereringen var inte omfattande och utgjorde inget problem i sig, men i och med att vissa analyser omöjliggjordes av ovanstående faktorer hände det att en sammansättning endast fick felaktiga analyser.
- Som väntat var det de långa orden som fick flest analyser.
- Många sammansättningar tilldelades endast en sammansatt analys, som direkt kunde väljas av analysvalsprogrammet. En del ord ställde dock till lite mer problem och vissa fick gå igenom hela fem olika urvalssteg i programmet.

7 Sammanfattning

Syftet med det här arbetet var att komplettera den svenska delen av MatsLex-databasen med information om vilka ord som är sammansättningar, vilka delar dessa utgörs av samt vilket huvudled sammansättningen har.

I ett första steg kördes samtliga ordformer i databasen genom Uppsala Chart Parser, UCP. Resultatet av detta var en fil med en böjningsanalys för varje ord och en eller flera sammansättningsanalyser för de flesta av sammansättningarna. UCP:s sammansättningskomponent är dock bara utvecklad för substantiv, vilket gjorde att över 10 % av sammansättningarna missades. Relativt många sammansättningar innehöll delar som inte fanns med som enkla ordformer i lexikonet och dessa kunder därför inte heller tilldelas någon korrekt sammansatt analys.

Filen med analyser användes som indata till ett program som väljer den bästa analysen för en ordform i de fall där UCP har kunnat analysera ordet på mer än ett sätt. Ett antal ordformer studerades i syfte att ta fram regler till programmet. Reglerna utgår ifrån tidigare arbeten inom detta område som visar att det ofta är den analys som har minst antal led och längst förled som är bäst. Utöver dessa regler lades även mer specifika regler till för att kunna hantera alla konstruktioner i materialet. Programmet utgår dock från att minst en analys för varje ord i indatafilen är korrekt samt att inget enkelt ord har kunnat få en sammansatt analys. Om så inte är fallet plockas felaktiga analyser ut. Reglerna i programmet bidrar också till fel eftersom det finns undantag till varje regel. 95 % av de analyser som plockades ut var dock korrekta.

Arbetet kom även att omfatta en del manuellt arbete. Dels gällde det att plocka ut de sammansättningar som UCP endast kunde tilldela en enkel analys och manuellt dela upp dessa och dels gällde det att gå igenom utdatafilen från analysvalsprogrammet för att plocka bort enkla ord som analyserats som sammansättningar och korrigera de ord där programmet valt fel analys.

Det faktum att 11617 av 17557 ord i databasen visade sig vara sammansättningar stöder påståendet att sammansättningar är särskilt vanliga i teknisk text. Dessutom visar det på hur viktigt det är att kunna hantera dessa på ett bra sätt i ett maskinöversättningssystem.

8 Utveckling/nästa steg

Många fler sammansättningar skulle kunna analyseras om sammansättningsanalysen byggdes ut så att den även kunde hantera andra ordklasser än substantiv, men i och med att fler ordklasser blandas in skulle även andelen felaktiga analyser öka. Om programmet kan hantera prepositioner kommer ordet *tidsur* att få både den korrekta analysen *tid.nn+ur.nn* och den felaktiga analysen *tid.nn+ur.pp*. Fler regler måste då läggas till i analysvalsprogrammet, så att det jämför ordklassen hos sammansättningen med ordklassen hos efterledet. Eftersom efterledet bestämmer hela sammansättningens ordklass måste dessa stämma överens. I de fall där det är förledens ordklass som skiljer sig åt mellan olika analyser kan dock problemet inte lösas med regler. Dessa ord riskerar att få en felaktig analys så länge inget bra sätt finns för att hantera semantisk information.

En annan möjlig utveckling för att få fram ännu säkrare analyser vore att utnyttja Karlssons (1992) Derivative Elimination Principle (se avsnitt 2.2.1). För att kunna använda denna princip måste dock avledningsregler läggas till.

Andra intressanta områden vore att studera andra korpusar och lexikon, från andra domäner, för att se hur väl de framtagna reglerna stämmer in på dessa eller att gå vidare med andra närliggande, sammansättningsrika språk som t.ex. tyska för att jämföra och se om det går att använda samma översättningsekvivalenter för de båda språken.

Referenser

Ahrenberg, Lars (1984). De grammatiska beskrivningarna i Sve.Ucp. I: Sågvall Hein, Anna (red.) *Föredrag vid De Nordiska Datalingvistikdagarna 1983*. Uppsala universitet. Centrum för datorlingvistik.

Brodda, B. (1979). Något om de svenska ordens fonotax och morfotax: Iakttagelser med utgångspunkt från experiment med automatisk morfologisk analys. I: *Papers from the Institute of Linguistics, University of Stockholm 38*. Stockholms universitet. Institutionen för lingvistik.

Dahllöf, Mats (1989). *Satslösning i en lexikonorienterad parser för svenska*. Göteborgs universitet.

Delsing, Lars-Olof (2001). Dagamma, dagstidning och s-genombrott. I: *Språkvård* nr 3, s 4-9.

Dura, Elzbieta (1998). *Parsing Words*. Göteborg: Novum Grafiska AB.

Hutchins, W. John & Somers, Harold L. (1992). *An Introduction to Machine Translation*. London: Academic Press Ltd

Karlsson, Fred (1992). A comprehensive Morphological Analyser for Swedish. I: *Nordic Journal of linguistics* nr 15, s. 1-45.

Liljestrand, Birger (1993). *Så bildas orden. Handbok i ordbildning* Lund: Studentlitteratur

Svenska akademiens ordlista över svenska språket.(!!)

Sågvall Hein, Anna (1983). *A Parser for Swedish. Status Report for Sve.Ucp. February 1983*. Rapport nr UC DL-R-83-2. Uppsala universitet. Centrum för datorlingvistik.

Teleman, Ulf, Staffan Hellberg, Erik Andersson mfl (1999). *Svenska Akademiens grammatik del 2 Ord*. Uddevalla: Media Print

Thorell, Olof (1981). *Svensk ordbildningslära*. Stockholm: Norstedts Tryckeri

Wennerberg, John (1962). De dubbelt sammansatta orden i svenskan. I: *Nysvenska studier* nr 41. Lund: Carl Bloms Boktryckeri AB

Övrig läsning om sammansättningar och maskinöversättning

Hui, Bowen (1998). *The role of morphology in Machine Translation*. Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.

Starbäck, Per (2001). UCP2. <http://stp.ling.uu.se/starback/ucp/>

Sågvall Hein, Anna & Dahllöf, Mats (1989). 2 Procedural Frameworks. I: Sågvall Hein, Anna, Dahllöf, Mats & Hörmander, Sofia, *Studies of grammars, formalisms, and parsing. A Report from the Project Grammar Models for Natural Language Processing. Rapporter från Språkdata 25*. Göteborgs universitet. Institutionen för språkvetenskaplig databehandling.

Sågvall Hein, Anna & Starbäck, Per (1999). A Test Version of a Grammar Checker for Swedish. In: Sågvall Hein, Anna (ed.) *Chart-based Grammar Checking. Papers by Anna Sågvall Hein, Per Starbäck, Per Weijnitz. Working Papers in Computational Linguistics & Language Engineering 12*. Uppsala universitet. Institutionen för lingvistik.

ten Hacken, Pius (1999). Motivated Tests for Compounding.I: *Acta Linguistica Hafniensia* nr 31, s 27-58. Köpenhamn: C.A. Reitzels forlag.

A Programkod

```
#!/usr/bin/perl

#Programmets huvuddel läser in rader från UCP:s utdatafil. För varje
#ny ordform som analyserats som sammansättning läggs analyserna in i en
#array. När antalet analyser är slut processas arrayen av sub analysval,
#som väljer ut den bästa analysen och lagrar den för utskrift.
#Sedan töms arrayen och nästa ordform med tillhörande analyser läggs in.
#Slutligen skrivs alla analyser ut till en fil sorterade efter huvudord.

#*****
#Här börjar huvudprogrammet
#*****

#Läs in fil: Fil med UCP:ns utdata, "indatalista.parses", som jag döper
#till indatafil. Öppna filen för läsning.
#-----

unless (open(indatafil, "indatalista.parses"))
{
    die ("cannot open input file");
}

#-----
#Öppna fil för utskrift:"klarfil" som jag döper till utdatafil.
#Till den ska den bästa analysen av varje ordform skrivas ut.
#Öppna filen för appending. Öppna även fil för utskrift av felmeddelanden
#samt fil där ord som inte fått någon sammansatt analys hamnar för manuell
#genomgång så att inget sammansatt ord smiter med där.
#-----

unless (open(utdatafil, ">>klarfil"))
{
    die ("cannot open output file");
}

unless (open(felfil, ">>felfil"))
```

```

{
    die ("cannot open output file felfil");
}

unless (open(enkelfil, ">>enkelfil"))
{
    die ("cannot open output file enkelfil");
}

#-----
#Läs in den första raden och lägg den i variabeln $word{"wordform"}.
#Detta behövs eftersom just denna rad inte passar in på mönstret
#för de övriga. För alla andra ordformer gäller att om programmet
#hittar en sådan så ska den array som redan finns processas och tömmas
#innan den nya ordformen läggs in. Men från början så finns ju ingen
#tidigare array.
#Counter är en räknare som ska göra så att de olika analyserna numreras
#och elementen får namn som lemma1, lemma2 o.s.v.
#processedwords räknar hur många ordformer som processas, d.v.s. har
#fler än en (godkänd) analys.
#laeggin kollar så att inte några särdrag i analysen visar att det inte är en
#grundform samt att den analyserats som en sammansättning och därför ska
#läggas in i arrayen.
#-----

$line = <indatafil>;
chop ($line);
$word{"wordform"} = $line;

$counter = 1;
$processedwords = 0;
$laeggin = 0;

#-----
#Läs information från filen och sortera in relevant information i en array.
#Loopa så länge det ej är en tom rad, d.v.s. end of file.
#Läs in en rad och gör saker beroende på vad det är för en rad.
#Detta är programmets "skelett".
#-----

```

```

while($line ne "")
{
    $line = <indatafil>;

    #-----
    #Om raden börjar med bokstav (alt siffra) är det en ordform. Om
    #counter=1 så har ingen analys godkänts och lagts in i förra
    #ordformens array. Om counter=2 så finns det bara en analys och
    #då läggs den genast vidare för utskrift. Annars processas
    #föregående ordforms array. Sedan läggs den nya ordformen in i arrayen.
    #Counter sätts till 1 för att den första analysen alltid ska heta 1.
    #-----

    if($line =~ /^[0-9a-zAÄÖ]/)
    {
        if($counter == 1)
        {
            #Ingen smsanalys finns för föregående ordform.
            print enkelfil (" $word{\\"wordform\\"}\n");
        }

        elsif($counter == 2)
        {
            #Endast en analys finns så den väljs.
            #Ta fram huvudordet: det sista ledet.

            @huvudordssplit = split(/\+/, $word{"lem1"});
            $plus = @huvudordssplit;
            $huvudord = @huvudordssplit[$plus-1];

            chop ($huvudord);
            chop ($huvudord);
            chop ($huvudord);

            if($huvudord =~ /[0-9]$/)
            {
                chop($huvudord);
            }
        }
    }
}

```

```

    if($huvudord =~ /^[^aouåeyäö]+$/)
    {
#t.ex. s.nn, kan vara fel
print felfil ("$word{\\"wordform\\"} $huvudord är kort\n");
    }

#lagra analysen för utskrift senare.

$tempvar = $word{"wordform"}.\t".$word{"lem1"};
$store{$tempvar} = $huvudord;
$efterindex{"$huvudord"} = $huvudord;

    }

    else
    {
        $printcounter = $counter-1;
        #Flera analyser finns, gå till subrutin för att
        #välja ut den bästa.
        &analysval(%word);
    }

#Lägg in den nya ordformen i arrayen, så att dess analyser kan
#börja lagras.
    chop ($line);
    $word{"wordform"} = $line;

$counter = 1;

} #(Slut if-sats som hittar ordformer)

#-----
#Om raden innehåller "lem=" är det ett lemma med formen
#lem = AFFÄR.NN+FLICKA.NN
#Andra delen av raden (delen efter "=") ska läggas in i arrayen
#på plats lem, men bara om ordet analyserats som sammansättning av UCP,
#(d.v.s. om COMPOUND har värdet +) samt om ordet står i grundform.
#laeggin sätts alltså till 1 om COMPOUND är + men till 0 om någon
#otillåten form hittas.
#-----

```

```

        elsif ($line =~ /COMPOUND \= \+\/)
        {
$laeggin = 1;
        }

        elsif ($line =~ /CASE \= GEN/ || $line =~ /FORM \= DEF/
|| $line =~ /NUMB \= PLUR/)
        {
$laeggin = 0;
        }

        elsif ($line =~ /LEM \=\/)
        {
if($laeggin)
{
        chop ($line);

        @lemmasplit = split(/ = /, $line);
        $word{"lem$counter"} = $lemmasplit[1];

        #Om det är en sammansatt analys och lemmat lagts in
        #läggs även nästa attributvärde, stammen, in.
        $stemline = <indatafil>;
        chop ($stemline);

        @stemsplit = split(/ = /, $stemline);
        $word{"stem$counter"} = $stemsplit[1];

        $counter++;
        $laeggin = 0;
}
}
} #(Slut while-sats som läser in rader från fil och lägger in viss
    #info i array.)

#-----
#För att även den sista ordformen ska komma med måste samma sak som vid
#funnen ny ordform utföras.
#-----

if($counter == 1)

```

```

{
    #(Ingen smsanalys finns för ordformen.)
}

elsif($counter == 2)
{
    #En analys finns. Ta fram huvudordet: det sista ledet.
    @huvudordssplit = split(/\+/, $word{"lem1"});
    $antalhuvudordsdelar = @huvudordssplit;
    $huvudord = @huvudordssplit[$antalhuvudordsdelar-1];

    chop ($huvudord);
    chop ($huvudord);
    chop ($huvudord);

    if($huvudord =~ /[0-9]$/)
    {
        chop($huvudord);
    }

    if($huvudord =~ /^[^aouåeiyäö]+$/)
    {
        #t.ex. s.nn
        print felfil ("$word{"wordform"}: $huvudord för kort\n");
    }

    if(length($huvudord) < 3)
    {
        #huvudordet är högst två bokstäver långt.
        print felfil ("$huvudord i $word{"wordform"} är kort. Kolla upp!!\n");
    }

    #lagra analysen.
    $tempvar = $word{"wordform"}."\t".$word{"lem1"};
    $store{$tempvar} = $huvudord;
    $efterindex{"$huvudord"} = $huvudord;
}

else

```

```

{
    $sprintcounter = $counter-1;
    &analysval(%word);
}

#-----
#När samtliga ordformer gåttts igenom och den bästa analysen för varje
#sammansättning valts ut sorteras dessa efter huvudord och skrivs ut till
#en fil. Sorteringen går till så att programmet tar ett efterled i taget
#i alfabetisk ordning och skriver ut de ordformer som har detta huvudord.
#OBS!! Tar tid! Borde finnas ett bättre sätt!!?
#-----

@keylista = sort keys(%efterindex);

for($a=0; $a<@keylista; $a++)
{
    foreach $grej (%store)
    {
        if($keylista[$a] eq $store{$grej})
        {
            @utskrift = split(/\t/, $grej);
            print utdatafil (" $utskrift[0]\nLEM = $utskrift[1]\n
Efterled = $store{$grej}\n\n");
        }
    }
}

*****
#Här slutar huvuddelen av programmet.
*****

*****
#Här börjar sub analysval, som tar den array som hör till en viss
#ordform och plockar ut den bästa analysen.
*****

```

```

sub analysval
{
    $processedwords++;

    $high = 0; #Här ska numret på den hittills bästa analysen lagras.
    $lemnumberhigh = 100; #Antalet delar i lemmat, med flit satt högt
    #för att första analysen alltid ska vara lägre.

    #-----
    #Loopa igenom arrayen och jämför de olika analyserna med varandra.
    #Först jämförs antalet led i lemmat. Den analys som har minst antal
    #led väljs. Om båda analyserna har lika många led går man vidare
    #till nästa steg.
    #-----

    for ($i=1; $i<$counter; $i++)
    {
@lempartsthis = split(/\+/, $word{"stem$i"});
$lemnumberthis = @lempartsthis;

if($lemnumberthis > $lemnumberhigh)
{
    #lemmat har fler led än föregående lemma, tas bort
    delete($word{"stem$i"});
    delete($word{"lem$i"});
}

elseif($lemnumberthis < $lemnumberhigh)
{
    #lemmat har färre led och väljs. Blir nytt bästalemma.
    #Om en analys lagts in tidigare, tas den bort.
    if($high)
    {
delete($word{"stem$high"});
delete($word{"lem$high"});
    }

    $lemnumberhigh = $lemnumberthis;
    $high = $i;
}
}

```

```

else
{

#-----
#Båda analyserna har lika många led i lemmat. Gå vidare och jämför
#vilken analys som har längst första led i stammen.
#Den med längst förled väljs enligt Karlssons princip.
#-----
@stempartsthis = split(/\+/, $word{"stem$i"});
@stempartshigh = split(/\+/, $word{"stem$high"});
$stemlengthhigh = length($stempartshigh[0]);
$stemlengththis = length($stempartsthis[0]);

if($stemlengththis > $stemlengthhigh)
{
#Längre första led i stammen; väljs.
delete($word{"stem$high"});
delete($word{"lem$high"});

$high = $i;
}

elseif($stemlengththis < $stemlengthhigh)
{
#Kortare första led i stammen; tas bort.
delete($word{"stem$i"});
delete($word{"lem$i"});
}

else
{

#-----
#Lika långa första led i stammen.
#Gå vidare och jämför längd på första del i lemmat.
#(Detta är inte alltid samma som i stammen).
#Längst del väljs, eftersom det oftare är t.ex.
#backe.nn /backa.vb än back.nn.
#-----

@lemsplit = split(/\+/, $word{"lem$i"});
$lengthlem1this = length($lemsplit[0]);

```

```

@lemhighsplit = split(/\+/, $word{"lem$high"});
$lengthlemlhigh = length($lemhighsplit[0]);

if ($lengthlemlthis > $lengthlemlhigh)
{
    delete($word{"stem$high"});
    delete($word{"lem$high"});
    $high = $i;
}

elsif ($lengthlemlthis < $lengthlemlhigh)
{
    delete($word{"stem$i"});
    delete($word{"lem$i"});
}

else
{

#-----
#Lika långa första delar i lemmat.
#En möjlig anledning är att det är ett par analyser
#av typen "stång/tång", där ett s kan tolkas som
#antingen foge-s eller början på nästa led.
#Om ett sådant par föreligger, välj den utan foge-s
#i samtliga fall utom de då första ledet slutar på en
#avledningsändelse.
#Analyser med fler än två led tar jag tills vidare
#hand om manuellt.
#-----

if($lemnumberthis > 2)
{
    print felfil ("{$word{"wordform"}} Varning!! Lika långa!!!
    Detta ska kollas upp manuellt!\n");
    delete($word{"stem$i"});
    delete($word{"lem$i"});
}

else
{
    if($stempartsthis[1] eq "s".$stempartshigh[1])

```

```

        {
            if($stempartshigh[0] =~ /ing$/ || $stempartshigh[0] =~ /tion/
|| $stempartshigh[0] =~ /het/)
        {
            delete($word{"stem$i"});
            delete($word{"lem$i"});
        }

else
{
    delete($word{"stem$high"});
    delete($word{"lem$high"});
    $high = $i;
}

        }

        elsif($stempartshigh[1] eq "s".$stempartsthis[1])
        {
            if($stempartsthis[0] =~ /ing$/ || $stempartsthis[0] =~ /tion/
|| $stempartsthis[0] =~ /het/)
            {
delete($word{"stem$high"});
                delete($word{"lem$high"});
                $high = $i;
            }

else
{
    delete($word{"stem$i"});
    delete($word{"lem$i"});
}

        }

else
{
#-----
#Om det inte är ett stångord jämförs andra
#ledens längd. Analysen med längst efterled
#väljs i första hand, eftersom man t.ex.
#hellre vill ha cykell.nn än cykel.nn
#-----

```



```

$antalhuvudordsdelar = @huvudordssplit;
$huvudord= @huvudordssplit[$antalhuvudordsdelar-1];

chop ($huvudord);
chop ($huvudord);
chop ($huvudord);

if($huvudord =~ /[0-9]$/)
{
    chop($huvudord);
}

if($huvudord =~ /^[^aouåeiyäö]+$/)
{
#t.ex. s.nn
print felfil ("$word{\\"wordform\\"}: $huvudord för kort\n");
}

if(length($huvudord) < 3)
{
#huvudordet är högst två bokstäver långt.
print felfil ("$huvudord i $word{\\"wordform\\"} är kort. Kolla upp!!\n");
}

#-----
#Lagra den bästa analysen i arrayen %store, som sedan ska skrivas ut
#sorterad efter huvudord.
#-----

$tempvar = $word{"wordform"}."\t".$word{"lem$high"};
$store{$tempvar} = $huvudord;
$efterindex{"$huvudord"} = $ord{"$huvudord"};

} #(Slut sub analysval.)

```

B Testkörning

1. Indata: lista över ord som hämtats ur databasen.

```
angrepp
korrosionshastighet
kraftuttagsvarvtal
låsringstång
tippanordning
```

2. Analyser från UCP. Olika typer:

a) Endast en enkel analys ges.

```
angrepp :
(* = (START = 1
      END = 9
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      COMPOUND = -
      LEM = angrepp.nn
      STEM = angrepp
      DIC.STEM = angrepp))
```

b) En enkel och en sammansatt analys ges.

```
korrosionshastighet :
(* = (START = 1
      END = 21
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = korrosion.nn+hastighet.nn
      DIC.STEM = korrosion+hastighet
```

```

        STEM = korrosionshastighet))
(* = (START = 1
      END = 21
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      COMPOUND = -
      LEM = korrosionshastighet.nn
      STEM = korrosionshastighet
      DIC.STEM = korrosionshastighet))

```

- c) En enkel och flera sammansatta analyser ges.
 (Specialfall: Foge-s kan tolkas fel.)

kraftuttagsvarvtal :

```

(* = (START = 1
      END = 20
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = kraft.nn+uttag.nn+svarv.nn+tal.nn
      DIC.STEM = kraft+uttag+svarv+tal
      STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = kraft.nn+uttag.nn+varv.nn+tal.nn
      DIC.STEM = kraft+uttag+varv+tal
      STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      COMPOUND = +

```

```

WORD.CAT = NOUN
GENDER = NEUTR
FORM = INDEF
NUMB = NIL
CASE = BASIC
LEM = kraft.nn+utttag.nn+varvtal.nn
DIC.STEM = kraft+utttag+varvtal
STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = kraftutttag.nn+svarv.nn+tal.nn
      DIC.STEM = kraftutttag+svarv+tal
      STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = kraftutttag.nn+varv.nn+tal.nn
      DIC.STEM = kraftutttag+varv+tal
      STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      LEM = kraftutttag.nn+varvtal.nn
      DIC.STEM = kraftutttag+varvtal
      STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20

```

```

COMPOUND = +
WORD.CAT = NOUN
GENDER = NEUTR
FORM = INDEF
NUMB = NIL
CASE = BASIC
LEM = kraft.nn+uttag.nn+varv.nn+tal.nn
DIC.STEM = kraftuttagsvarv+tal
STEM = kraftuttagsvarvtal))
(* = (START = 1
      END = 20
      WORD.CAT = NOUN
      GENDER = NEUTR
      FORM = INDEF
      NUMB = NIL
      CASE = BASIC
      COMPOUND = -
      LEM = kraftuttagsvarvtal.nn
      STEM = kraftuttagsvarvtal
      DIC.STEM = kraftuttagsvarvtal))

```

- d) En enkel och flera sammansatta analyser ges.
 Specialfall: Foge-s kan tolkas fel.

kombinationstång :

```

(* = (START = 1
      END = 18
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF
      NUMB = SING
      CASE = BASIC
      LEM = kombination.nn+stång.nn
      DIC.STEM = kombination+stång
      STEM = kombinationstång))
(* = (START = 1
      END = 18
      COMPOUND = +
      WORD.CAT = NOUN
      GENDER = UTR
      FORM = INDEF

```

```

    NUMB = SING
    CASE = BASIC
    LEM = kombination.nn+tång.nn
    DIC.STEM = kombination+tång
    STEM = kombinationstång))
(* = (START = 1
    END = 18
    WORD.CAT = NOUN
    GENDER = UTR
    FORM = INDEF
    NUMB = SING
    CASE = BASIC
    COMPOUND = -
    LEM = kombinationstång.nn
    STEM = kombinationstång
    DIC.STEM = kombinationstång))

```

- e) En enkel och en sammansatt analys ges.
 Specialfall: Endast felaktiga sammansatta analyser ges.

tippanordning :

```

(* = (START = 1
    END = 15
    COMPOUND = +
    WORD.CAT = NOUN
    GENDER = UTR
    FORM = INDEF
    NUMB = SING
    CASE = BASIC
    LEM = tipp.nn+anordning.nn
    DIC.STEM = tipp+anordning
    STEM = tippanordning))
(* = (START = 1
    END = 15
    WORD.CAT = NOUN
    GENDER = UTR
    FORM = INDEF
    NUMB = SING
    CASE = BASIC
    COMPOUND = -
    LEM = tippanordning.nn
    STEM = tippanordning

```

DIC.STEM = tippanordning))

3. Utdata från analysvalsprogrammet.

a) Enkla analyser ignoreras.

b) När endast en sammansatt analys finns väljs denna.

korrosionshastighet :
LEM = korrosion.nn+hastighet.nn
Efterled = hastighet

c) Programmet väljer den analys som har minst antal led.

kraftuttagsvarvtal :
LEM = kraftuttag.nn+varvtal.nn
Efterled = varvtal

d) Efter avledningsändelsen ''-ing'' tolkas s som ett foga-s.

kombinationstång :
LEM = kombination.nn+tång.nn
Efterled = tång

e) Även om endast felaktiga sammansatta analyser ges från UCP väljs en av dessa. Informationen måste korrigeras manuellt.

tippanordning :
LEM = tipp.nn+anordning.nn
Efterled = anordning

tippanordning :
LEM = tippa.vb+anordning.nn
Efterled = anordning