

Syntaktisk informert frasesamanstilling

Kevin Brubeck Unhammer

07/11, 2010

Innhold

1	Innleiing	1
1.1	Vegkart	4
2	Bakgrunn og omgrepsavklaring	5
2.1	Metodar for frasesamanstilling	5
2.2	Eit kort oversyn over leksikalsk-funksjonell grammatikk og terminologi	7
3	Krav til frasesamanstilling	10
3.1	Innleiing	10
3.2	Formål med frasesamanstilling	10
3.3	Frasesamanstilling i ein LFG-trebank	12
3.4	Kva kan lenkjast?	14
3.5	Krav på ordnivå	16
3.5.1	Ordklasse	17
3.6	Krav på f-strukturnivå	18
3.6.1	Krav om lik argumentstruktur	18
3.6.2	Ulik følge i argumentstruktur	20
3.6.3	Krav om argumentlenkjer	21
3.6.4	Adposisjonsobjekt	23
3.6.5	Kausativar og inkorporering	24
3.7	Krav på c-strukturnivå	26
3.7.1	Lenkja f-strukturar utan c-strukturnodar	29
3.7.2	Eit strengare lenkjingskriterium	32
3.7.3	Funksjonelle c-strukturnodar	34
3.8	Rangering	36
3.8.1	Rangering ved følge	36
3.8.2	Rangering ved djupn	37
3.8.3	Rangering for heile samanstillinga	37
3.9	Oppsummering	38

4	Implementasjonen av lfgalign	39
4.1	Lenkjer mellom f-strukturar	40
4.1.1	Overflødige adverbial	44
4.1.2	Når f-strukturlenkjene ikkje er ein-til-ein	44
4.1.3	Kan me gjere f-struktursamanstillinga bottom-up?	45
4.2	Rangering	45
4.3	Lenkjing av c-strukturnodar	47
5	Diskusjon	50
5.1	Materiale	50
5.2	N-grambaserte metodar	51
5.3	Kor avhengig er lfgalign av bottom-up-informasjon?	54
5.3.1	Kommentarar og feilanalyse	55
5.3.2	Overlapp med RIA	58
5.4	Opne problem	61
5.4.1	Fragmentariske analysar	61
5.4.2	«Mjuk» LPT-korrespondanse, og rangering som avskjering	62
5.4.3	Top-down-lenkjing av f-strukturar, og problemet med re-entrancy	62
5.5	Bruksområde	64
5.5.1	Oppdage argumentstrukturalternasjon	64
5.6	Oppsummering	64
6	Avslutning	67
A	Kode for å køyre RIA-evaluering	72

Samandrag

This thesis describes the development of an automatic phrase alignment method using as input parallel sentences parsed in Lexical-Functional Grammar, where similarity in analyses is used as evidence that constituents (syntactic phrases) or functional elements (predicates, arguments, adjuncts) may be linked. A set of principles for phrase alignment are formulated, with the goal of annotating a parallel treebank for linguistic research. An implementation is given and evaluated and compared with methods based on N-gram tables.

fylle ut, og så
skrive på
norsk..

Kapittel 1

Innleiing

Denne masteroppgåva utforskar kva det vil seie at to uttrykk er omsetjingar av kvarandre, og korleis me automatisk kan generere samanstillingar (*alignments*) av uttrykk som står i eit slikt omsetjingsforhold.

Omsetjingsforhold finn me mellom ord og setningar i kontekst på ulike språk, men me kan au finne ulike typar ekvivalensforhold (samanstillingar) mellom frasar innanfor setningane, og mellom lingvistiske skildringar av setningane. I samanheng med Xpar-prosjektet (XPar, 2008; Dyvik et al., 2009) har eg sett på metodar for automatisk frasesamanstilling – å finne omsetjingsforhold mellom grupper av fleire ord. Resultatet blir ein *annotasjon*, endå ei lingvistisk skildring av tekstene.

Det at me kan finne korrespondansar mellom lingvistiske skildringar (t.d. frasestrukturannotasjonar, eller trekkstrukturane til dei grammatiske rammeverka HPSG eller LFG) gjer det tydeleg at me arbeider med ein *modell* av språket; ulike skildringar kan vere sanne innanfor modellen, utan at modellen er lik språket. Korrespondansane er au teoretiske storleikar, og me kan leggje ulike kriterium til grunn for å seie at to representasjonar korresponderer; på same måte som me kan leggje ulike kriterium til grunn for å seie at to ulike uttrykk er omsetjingar av kvarandre.

Kriteria avheng av formålet. Samanstillingsannotasjon kan t.d. nyttast som grunnlag for statistisk eller eksempelbasert maskinomsetjing, i tillegg til oppbygging av parallelle korpora for språkstudium. For statistisk maskinomsetjing vil alle uttrykk vere omsetjingar av kvarandre med eit visst sannsyn (kanskje null), ein har vanlegvis ikkje kriterium som krev lingvistisk analyse. Når samanstillinga skal nyttast i parallelle korpora for lingvistiske undersøkingar vil ein kanskje ha krav om at uttrykk som skal lenkjast er «like» på eit eller anna mål, utover at dei har opptredt saman ofte.

I den manuelle samanstillinga i Samuelsson & Volk (2006) har dei t.d. ein del reint semantiske kriterium for å opprette fraselenkjer i ein parallell trebank, men dei har ikkje krav om syntaktisk likskap. Xpar-prosjektet, som denne masteroppgåva er ein del av, har mellom anna som mål å oppdage forhold mellom grammatiske funksjonar, tematiske roller og kasusmarkering, ved hjelp av parallelle trebankar annotert med djupe grammatiske analysar. Trebankane vil innehalde parallell tekst

på norsk, georgisk, tigrinya og nederlandsk. For å lenkje to frasar i dette prosjektet krev me ein viss syntaktisk likskap i omgivnadene til frasane, sjølv om frasane internt kanskje er syntaktisk ulike.

Grammatikkane som gir analysane er utvikla med tanke på at *like syntaktiske fenomen på ulike språk skal få like analysar*, frasesamanstillinga bør då kunne tene på at omsetjingar som har ein syntaktisk likskap vil få liknande analysar.

Dei grammatiske analysane er gjort i leksikalsk-funksjonell grammatikk, LFG (Bresnan, 2001). Ei grammatisk analyse i LFG involverer både konstituentstruktur (c-struktur) og funksjonell struktur (f-struktur). Konstituentstrukturen liknar på frasestrukturen frå andre grammatiske tradisjonar. Dei funksjonelle strukturane er trekkstrukturar, som mellom anna representerer avhengnadsforhold mellom syntaktiske funksjonar som predikat, subjekt og objekt, i tillegg til å halde informasjon om grammatiske trekk som genus, tal eller kasus. Nodar i c-strukturen kan spesifisere informasjon på ulike stader i f-strukturen¹.

I Xpar-prosjektet vil ein finne ut om metodar for frasesamanstilling kan tene på det at LFG-grammatikkane for dei ulike språka er skrivne med same prinsipp lagt til grunn; to parallellstilte setningar bør ha f-strukturar som er like nok til at me kan samanstille frasar ved hjelp av likskapen mellom f-strukturane. I Dyvik et al. (2009, s. 72) finn me følgjande hypotese:

On the basis of monolingual treebanks constructed from a parallel corpus by means of parallel grammars it will be possible to achieve automatic word and phrase alignment with significantly higher precision and recall than hitherto achieved through other means.

kor «parallel grammars» her tydar at grammatikkane har ein viss parallellisme i både f-struktur og c-struktur.

Men i tillegg til at ein kanskje kan få betre skåre på desse kvantitative måla, vil lenkjer mellom f-strukturar gi informasjon som er kvalitativt forskjellig frå det ein kan få med å berre sjå på lenkjer mellom ord, N-gram eller konstituentar.

Og sidan f-strukturane er meint å skildre informasjon på eit meir «språkuavhengig» nivå enn t.d. konstituentstruktur, bør f-strukturane vere gode kandidatar for å finne informasjon som korresponderer på dei to språka. Tanken er at me frå to f-strukturar som skildrar omsette setningar, kan

1. lage ei samanstilling mellom relevante delar av f-strukturane,
2. nytte denne funksjonelle samanstillinga til å finne ei frasesamanstilling, ved å følge avbildinga frå f-struktur til c-struktur.

Eitt problem som byr seg er hypotesemangfaldet: kva for «delar av f-strukturane»? Korleis kan me avgrense søkjerommet? I det minste må me kunne kople dei lenkja f-strukturane opp mot c-strukturknodar; her er PRED-elementet ein god kandidat.

¹Ved c-struktur-f-strukturavbildinga ø, ein funksjon som tek ein c-strukturnode og returnerer ein (delvis) f-struktur.

Vidare må me vite *korleis* me samanstillar desse delene. Me kan t.d. byrje med å lenkje dei yttarste f-strukturane frå kvart språk, og så rekursivt lenkje visse relevante substrukturar². Eit naivt førsteutkast til ei *f-samanstilling*, samanstilling på f-strukturnivå, kan då sjå slik ut:

$$falign(f_1, f_2) = \{(f_1(\text{PRED}), f_2(\text{PRED}))\} \cup \bigcup_{g_1, g_2 \in fpairs(f_1, f_2)} falign(g_1, g_2)$$

Funksjonen *falign* vil gi ei mengd av par av f-strukturar, kor kvart par altså er samanstillt. Problemet er då å finne ut kva for par av substrukturar som er «relevante», her representert ved *fpairs*(f_1, f_2).

Sjølvs om f-strukturar abstraherer frå skilnadene i korleis ulike språk nyttar ordgruppering og ordform til å kode syntaktiske forhold (Bresnan, 2001, s. 14), vil det oppstå forskjellar i f-strukturane til to parallelstilte setningar i eit korpus; både pga. «omsetjarfridom», ulikskap i argumentstrukturar og det at ulike språk nyttar ulike syntaktiske funksjonar til å uttrykkje det same konseptet. Det kan t.d. godt hende at me bør lenkje eit objekt på eitt språk til eit setningskomplement på eit anna språk. Skal ein algoritme gå frå f-strukturar til frasesamanstilling må han i det minste vere robust nok til å takle slik mangel på samsvar. Til å byrje med kan me tenkje oss at *fpairs* gir alle par av grammatiske funksjonar som har same plass i argumentstrukturen til predikatet. Så viss predikatet ‘*sein*’ i f_1 har eit subjekt på førsteplass i strukturen, og så eit setningskomplement, medan ‘*have*’ i f_2 har eit subjekt og så eit objekt, vil *fpairs* i det minste returnere $\{(f_1(\text{SUBJ}), f_2(\text{SUBJ})), (f_1(\text{XCOMP}), f_2(\text{OBJ}))\}$. Ved å nytte argumentplass kan me då enkelt få til lenkjer mellom grammatiske funksjonar med ulike namn. Men der me ikkje har slikt samsvar i argumentstrukturar, vil *fpairs* ha ein vanskelegare jobb.

Og sidan me gjerne au vil lenkje adverbial, som er representert som uordna mengder i f-strukturane, er det klart at *fpairs* ikkje er triviell å definere. Ein del av denne oppgåva vil altså vere å komme med forslag til funksjonen *fpairs*.

Om to f-strukturar er lenkja, har me grunn til å lenkje c-strukturnodane som projiserer dei. Men her er det ikkje sikkert me vil lenkje *alle* nodane; intuitivt vil me berre at nodar som dominerer korresponderande innhald skal lenkjast. Ein formalisering dette steget, med diskusjon rundt problema, inngår au i denne oppgåva.

I første omgang spesifiserer eg kva for lenkjer mellom f-strukturar og c-strukturnodar me *ønskjer*. Eg implementerer så eit program `lfgalign` som automatisk finn samanstillingar med slike lenkjer. Dette programmet opprettar frasesamanstillingar med hjelp av f-strukturinformasjonen gitt av grammatikkar som er skrivne på felles prinsipp, i tillegg til å kunne avgrense lenkjingar med hjelp av bottom-up-informasjon om kva for ordlenkjer som er moglege. F-strukturane avgrensar igjen kva for ordsamanstillingar som er moglege, og kva for c-strukturnodar (syntaktiske

²Dette krev sjølvstendig at dei yttarste f-strukturane faktisk korresponderer i lenkja setningar, noko me ikkje alltid kan ta for gitt.

frasar) som kan lenkjast. Til sist evaluerer eg resultatet av å køyre programmet mitt, og samanliknar dette med kva for samanstillingar me kan få frå andre metodar.

1.1 Vegkart

I neste kapittel gir eg eit oversyn over feltet *frasesamanstilling*, i tillegg til ein kort introduksjon til terminologi og konseptar frå LFG som blir nytta i resten av teksta.

I kapittel 3 går eg gjennom kva me ønskjer av ei frasesamanstilling når formålet m.a. er å oppdage relasjonane mellom syntaktiske funksjonar, kasusmarkering og tematiske roller med hjelp av ein parallell trebank. Dette ender opp i ei mengd med «krav» som samanstillingane må fylle for å vere lovlege, og som implementasjonen av den automatiske frasesamanstillinga må følge. Eg gir i tillegg nokre heuristiske rangeringskriterium for dei tilfella der me har ulike konkurrerande f-struktursamanstillingar. Eit oversyn over implementasjonen kjem i kapittel 4.

Eg evaluerer samanstillingane som kjem ut av denne metoden i kapittel 5. Her samanliknar eg desse samanstillingane med det som er mogleg der me berre har konstituentstruktur (syntaktiske tre) i tillegg til N-grambasert ordsamanstilling. Eg nyttar dei typologisk svært ulike språka georgisk og norsk i eit lite testsett kor eg går gjennom lenkjingane manuelt. I tillegg ser eg på forskjellane mellom f-strukturlenkjingane frå min implementasjonen og dei som kjem frå ein N-grambasert metode for lenkjing av f-strukturar, på eit større, tysk-engelsk testsett.

Kapittel 2

Bakgrunn og omgrepsavklaring

I dette kapitlet gir eg eit oversyn over dei vanlegaste metodane innanfor feltet frasesamanstilling, og då spesielt dei som nyttar djup syntaktisk analyse eller rettar seg mot trebankar. Eg gir au ein kort gjennomgang av nokre teoretiske omgrep og konsept som eg kjem til å nytte i resten av oppgåva.

2.1 Metodar for frasesamanstilling

Frasesamanstilling vil seie lenkjing av (representasjonar av) delar av setningar som (representerer ord som) har ein omsetjingsmessig korrespondanse. Merk at ordet «frase» ofte blir nytta i litteraturen om kontinuerlege strenger av ord (N-gram) som ikkje treng vere syntaktiske konstituentar. I vid forstand kan me au inkludere lenkjing av delar av dependensstrukturar, eller av syntaktiske funksjonar, som begge representerer mengder med ord.

Automatisk frasesamanstilling er eit nytt felt. Det finst allereie veldig gode system for automatisk lenkjing av setningar; her har ein fått svært gode resultat ved å nytte ein statistisk omsetjingsmodell (Chen, 1993); andre metodar har nytta avstand eller delstrengoverlapp (Manning & Schütze, 1999, s. 467–484 gir eit oversyn). Automatisk samanstilling av ord har au komme langt (sjå Brown et al. (1993) for dei klassiske «IBM-modellane»; Och & Ney (2003) gir eit godt oversyn over ytinga til leiande metodar). Men på nivåa mellom ord og setning er det vanskelegare å vurdere feltet. Det finst fleire moglege einingar å lenkje – kontinuerlege N-gram, kontinuerlege eller diskontinuerlege konstituentar, dependensstrukturar, syntaktiske funksjonar – og i motsetning til einingar som *ord* eller *setning*, er einingane i ein frasesamanstilling sjeldan teoretisk ukontroversielle¹. Dei ulike tilnærmingane som finst, og einingane dei lenkjer, er prega av formåla til utviklarane.

Eit av dei tidlegaste forsøka på å lenkje frasar var Kupiec (1993), her berre nominalfrasar. Metoden besto i å først køyre ein statistisk ordklassetaggjar, så finne sannsynlege nominalfrasar på kvart språk (dvs. «chunking») med reine regulære

¹Manning & Schütze (1999, s. 470) skil i tillegg mellom *samanstillingsproblem* og *korrespondanseproblem*, der berre det siste kan involvere kryssande avhengnader.

uttrykk, og så lenkjing av slike kontinuerlege ordstrengar basert på sannsynsmaksimering². Resultata var relativt gode for enkle frasar (nitti av dei hundre høgast rangerte korrespondansane var gode), men modellen var svært enkel og involverte ikkje nokon kontekst rundt frasane.

Innanfor korpuslingvistikken har Piao & McEnery (2001) nytta enkel kollokasjonsinformasjon og ordklasseheuristikkar for å først finne sannsynlege nominale frasar på engelsk og kinesisk, og så lenkje desse ved hjelp av sannsynsheuristikkar som t-skåre og *Mutual Information*. Dei køyrer fleire runder med lenkjing av lengre og lengre N-gram. Her, som i Kupiec (1993), er evalueringsgrunnlaget rett og slett ein manuell gjennomgang av dei mest sannsynlege omsetjingane dei får.

Den manuelle frasesamanstillinga i Samuelsson & Volk (2006), nemnt i introduksjonen, blei nytta som evalueringsstandard for den automatiske metoden i Samuelsson & Volk (2007). Her finn dei ei konstituentsamanstilling frå ei ordsamanstilling, der berre N-gram som svarer til ein syntaktisk node blir lenkja som frasar. Formålet er å lage ein parallell trebank, kor det altså er unyttig å lenkje «frasar» som *ikkje* er konstituentar. Eg kjem tilbake til denne metoden i kapittel 5.

Sjølv om fraselenkjer kan vere nyttige i korpuslingvistikken er det hovudsakleg innanfor statistisk maskinomsetjing at ein har forska på samanstilling av frasar. Koehn et al. (2003) gir ei grundig evaluering av ulike statistiske metodar for frasesamanstilling til bruk i stokastisk maskinomsetjing. Dei nyttar BLEU-skåren til å rangere resultata (Papineni et al., 2001, i Koehn et al., 2003, s. 51), som gir ei rangering ved (N-grambasert) samanlikning med ferdig omsett tekst.

Den første metoden, *AP*, er reint N-grambasert. Dei nyttar verktøyet Giza++ (Och og Ney, 2000, i Koehn et al., 2003, s. 50) til å indusere ordsamanstilling frå eit setningssamanstilt korpus (vha. «modell 4» for ordsamanstilling, utvikla ved IBM av Brown et al. (1993)). Denne samanstillinga er 1-til-n (t.d. eitt engelsk ord til to franske), så dei finn ordsamanstilling for begge retningar og tek så snittet av alle moglege N-gramsamanstillingar som ikkje er i konflikt med ordsamanstillingane. Dei føyer så på ord frå unionen av desse vha. nokre enkle heuristikkar.

Den andre metoden, *Syn*, tek berre med dei frasane som står under syntaktiske nodar i eit parsar korpus; frasesamanstillinga til *Syn* er ein delmengd av den i *AP*. Denne syntaktisk informerte modellen gav ein mykje dårlegare BLEU-skåre enn den reint N-grambaserte modellen (faktisk dårlegare enn omsetjingane frå den opphavlege modell 4 for ordsamanstilling, utan frasesamanstilling). Dei forklarar dette med den store mengda uttrykk som ikkje utgjer syntaktiske konstituentar i følge parsaren deira, men likevel konsekvent blir omsett til visse uttrykk på det andre språket (t.d. «es gibt» på tysk til «there is» på engelsk).

Seinare resultat har vist at ein *kombinasjon* av syntaktisk informerte metodar med reint N-grambaserte modellar (dvs. i motsetning til å berre fjerne samanstillingar mellom ikkje-konstituentar) kan auke skåren i ein maskinomsetjingsevaluering, både om ein som i *Syn*-modellen nyttar frasestrukturinformasjon, men i endå stør-

²Dette er ein av dei vanlegaste metodane for iterativ reestimering av parametre i ein statistisk modell; eg kjem tilbake til sannsynsmaksimering i kapittel 5.

re grad om ein nyttar dependendsinformasjon (Tinsley et al., 2007; Hearne et al., 2008). Dette er interessant med tanke på at LFG-analysane gir begge typar informasjon.

Riezler & Maxwell (2006) utvikla ein metode for å kombinere frasebasert statistisk maskinomsetjing med LFG-basert setningsgenerering. Dei finn ei n-til-m-ordsamanstilling med Giza++ som i metodane over, men parsar i tillegg setningane i LFG. Dei to moglege f-strukturane som liknar mest blir valt ut, og frå ordsamanstillinga finn dei mange-til-mange-korrespondansar mellom substrukturane i f-strukturane. Ved å leggje til LFG-basert generering fekk det kombinerte systemet betre resultat på langdistanseavhengnader og generalisering til nye uttrykk med strukturell likskap til tidlegare observerte uttrykk. Dei går altså frå ordlenkjer til f-strukturlenkjer, motsett retning frå metoden i denne oppgåva. Ordlenkjer har au gitt f-strukturlenkjer i overføringsbasert (transferbasert) statistisk maskinomsetjing (Graham & Genabith, 2010, 2009; Graham et al., 2009), som eg kjem tilbake til i kapittel 5.

Så langt har eg ikkje komme over metodar som prøver å finne eller betre på frase- og ordsamanstilling direkte frå ein LFG-parse – det er dette som er strategien til programmet `lfgalign` i kapittel 4 – men det er stor overlapp mellom krava som kjem i kapittel 3 og dei gitt i den første publiseringa i Xpar-prosjektet, Dyvik et al. (2009).

2.2 Eit kort oversyn over leksikalsk-funksjonell grammatikk og terminologi

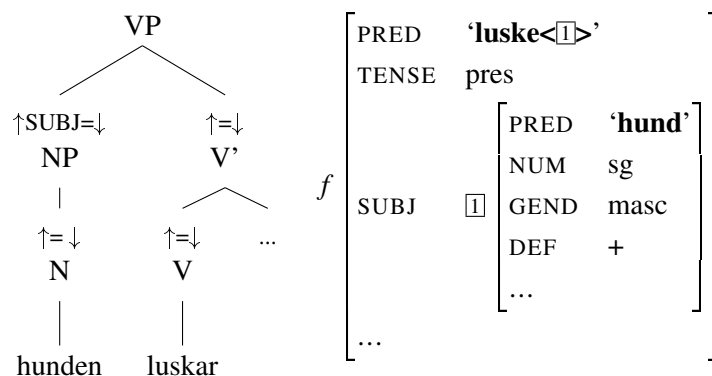
I dei følgjande kapitla nyttar eg ein del terminologi frå LFG, Leksikalsk-Funksjonell Grammatikk. Difor gir eg her eit kort oversyn over det som kan vere nytt for dei som er meir vand med andre grammatiske rammeverk, i tillegg til å avklare eit par eigne termar eg nyttar i teksta.

LFG er eit **modellteoretisk**, ikkje-derivasjonelt, rammeverk for grammatikk. Pullum & Scholz (2001) gir ein god gjennomgang av forskjellen mellom dei meir tradisjonelle derivasjonelle (au kalla enumerative) grammatikkane og modellteoretiske grammatikkar. Derivasjonelle grammatikkar, som transformasjonsgrammatikkane til Chomsky, definerer eit språk som *ei mengd av uttrykk* ved avleiing frå eit startsymbol. Ein modellteoretisk grammatikk, derimot, gir skildringar av *enkeltuttrykk*, kor eitt uttrykk kan ha fleire moglege skildringar (språket er ikkje definert som ei mengd).

Ein modellteoretisk grammatikk kan i tillegg skildre strukturen (eller dei moglege strukturane) til *fragment* av setningar, og denne strukturen er lik det bidraget som fragmentet tilfører analysen av heile setninga. Det tilsvarande er ikkje mogleg å gjere derivasjonelt. Pullum & Scholz (2001, s. 32–33) gir t.d. eit fragment som kjem midt i eit høgreforgreina tre; ei derivasjonell skildring ville måtte skildre treet over eller under, men utan informasjon om kva som kjem til høgre eller venstre kan me ikkje (på ein ikkje-vilkårleg måte) skildre subtreet utanfor fragmentet heilt

fram til terminal- eller startsymbol.

I LFG har analysane ulike *nivå*, eller *strukturar* (dette er ein av hovudforskjellane frå rammeverket HPSG (Sag et al., 2003), som LFG elles kan likne på). Konstituentforhold er skildra i **c-strukturen** («constituent structure»), medan forhold mellom syntaktiske funksjonar og grammatiske trekk kjem til syne i **f-strukturen** («functional structure»), ein trekkstruktur. Ein trekkstruktur er ei mengd attributt og verdiar, kor ein verdi kan vere atomær eller peike på ein ny trekkstruktur. Figur 2.1 illustrerer eit enkelt døme for eit fragment.



Figur 2.1: Konstituentstruktur og funksjonell struktur

Konstituentstrukturen liknar på tradisjonelle frasetre, kor dominans mellom nodane viser frasehierarkiet i analysen av setninga. Nodekategoriane er vanlegvis basert på \bar{X} -prinsipp. Hovudet i ein frase er XP, ein XP kan bestå av ein *spesifikatorfrase* (valfritt) og ein X' (\bar{X}). Ein X' kan bestå av ein X' og eit *adjunkt*, eller ein X og eit *komplement*³. I figur 2.1 har me t.d. ein VP (her er X=V), med ein spesifikator til venstre (ein ny frase, NP), og V' (dvs. \bar{V}) til høgre. V' består av V og kanskje eit komplement til høgre. I dette tilfellet er spesifikator subjekt (kanskje har me eit refleksiv pronomen som komplement).

I tillegg har kvar node i LFG ei kopling til f-strukturen, via **c-struktur-f-strukturavbildinga** ϕ . Nodar i c-strukturen kan spesifisere informasjon på ulike stader i f-strukturen (me seier at nodane **projiserer** f-strukturar, eller delar av dei). I dette tilfellet går ϕ av VP her til f-strukturen f , VP projiserer f . NP-noden er annotert med $\uparrow\text{SUBJ}=\downarrow$, dette les me som at «denne noden projiserer subjektet til ϕ av mornoden», altså projiserer NP-en SUBJ av f . NP er ikkje åleine om å gjere dette, N-noden har $\uparrow=\downarrow$ som vil seie at N projiserer same f-struktur som NP. Dette subjektet har fleire trekk i f-strukturen, t.d. NUM og GEND som har atomære verdiar og seier at dette er i eintal og maskulinum. Viss eit anna ord i setninga må samsvare med dette for å vere grammatisk, kan me krevje i grammatikken at me kan **unifisere** visse trekk; for atomære trekk som dette kan me alltid unifisere dei

³Det kan sjølvsagt vere gode grunnar (frå evidens på kvart av dei to språka) til at argumentstrukturane er ulike.

viss atomet er formmessig likt. Me kan au unifikere heile trekkstrukturar så lenge dei ikkje har trekk som ikkje kan unifikere; dei unifikerte strukturane er då blitt *ein* struktur, og alle referansar til dei to peiker no på same struktur. Slik er det mogleg å få *sykliske* strukturar – ein f-struktur er ein *graf*. Det er altså ikkje mogleg å gjere om ein f-struktur til ein trestruktur utan å miste informasjon.

I figur 2.1 er verdien av PRED-trekket til subjektet '**hund**'. PRED er eit spesielt trekk, verdien her er ein *semantisk form*. Desse er alltid *unike*, og kan ikkje unifikereast sjølv om dei har lik form. I tillegg viser dei *argumentstrukturen* til predikatet. I figur 2.1 har predikatet '**luske**' eitt argument, subjektet (det at argumentet er unifikert med SUBJ-trekket er vist ved at dei begge har indeksen $\bar{1}$).

Visse *endosentrisitetsprinsipp* avgrensar avbildinga mellom c-struktur og f-struktur, ved å vise til \bar{X} -kategoriane; til dømes har me alltid $\uparrow=\downarrow$ på ein X' som står under XP.

Avbildinga frå c-struktur til f-struktur er mange-til-ein. Som nemnt projiserte både NP og N same f-struktur. Desse nodane dominerer same ord i c-strukturen, men det går fint an at to nodar som dominerer ulike mengder med ord kan projisere same f-struktur; då har me ein **diskontinuerleg konstituent**.

Viss me følgjer avbildinga frå c-struktur til f-struktur tilbake til c-strukturen igjen, finn me det **funksjonelle domenet** til ein f-struktur. Me skriv $\phi^{-1}(f)$ (altså inversen av ϕ) for det funksjonelle domenet til f-strukturen f . Dette tilsvarer dei nodane i c-strukturen som saman projiserer denne f-strukturen (Bresnan, 2001, s. 126). Sidan dette er inversen av ein funksjon, kan me altså ha diskontinuerlege konstituentar i same funksjonelle domene, på same måte som ulike argument til ein funksjon kan gi same verdi.

I denne oppgåva nyttar eg, i tillegg til LFG-terminologien, orda *lenkjing* og *samanstilling* i omtrent same tyding som dei engelske termene *link* og *alignment*. Ei samanstilling er ei mengd lenkjer. Merk at ei enkeltlenkje treng ikkje å vere ein-til-ein. Lenkjer og samanstillingar er ekvivalensforhold som me kan finne mellom lingvistiske *representasjonar* (f-struktur, c-struktur) eller *uttrykk* (ord, setningar). Lenkjing mellom dei siste er meir ateoretisk/datanært – grunnlaget for å opprette ei lenkje mellom to c-strukturknodar (representasjonar) er at uttrykka i kontekst som dei representerer er omsetjingar (og har lik nok syntaks i følgje dei to grammatiske analysane til at me kan lenkje nodane). Neste kapittel prøver å avgrense *når* me ønskjer å lenkje to representasjonar.

Kapittel 3

Krav til frasesamanstilling

3.1 Innleiing

I denne delen prøver eg å finne fram til kva som er den best moglege frasesamanstillinga. Eg argumenterer for at «best» her må tolkast i forhold til eit formål; her er formålet å annotere ein trebank for lingvistiske studium, m.a. for å undersøkje ulike samsvar mellom kasusmarkering og semantisk rolletildeling. Som utgangspunkt har eg visse krav for ordsamanstilling gitt i Thunes (2003), saman med krava for frasesamanstilling i Dyvik et al. (2009). Eg viser kvifor ein, for våre formål, må revidere kravet til Thunes om likskap i argumentstruktur. Eg gir nokre døme for å grunngje krava i Dyvik et al. (2009), i tillegg til å utdjupe dei for å gjere dei enklare å implementere i kapittel 4. Dette involverer au å omformulere krava for c-struktursamanstilling slik at dei ikkje refererer til ordlenkjer, berre f-strukturlenkjer. Sidan eit av måla med Xpar-prosjektet er å finne ut kor mykje frasesamanstillingsinformasjon me kan få ut av parallellismen i f-strukturane (eller, sett frå den andre sida, kor uavhengig ein kan gjere seg av den bottom-up-informasjonen ei ordlenkje gir), blir det eit avleidd mål å formulere frasesamanstillingskrava med referanse til f-strukturane der det går an.

3.2 Formål med frasesamanstilling

Ei frasesamanstilling er ein slag annotasjon av eit korpus. På same måte som oppbygginga av eit korpus avheng av formålet til korpuset, kan ein ikkje definere den ideelle annotasjonen av eit korpus utan å ta høgd for kva ein skal nytte annotasjonen til.

Me kan illustrere dette med eit enkelt, praktisk døme: ved automatisk ordklassetagging må ein gjerne avvege mellom dekning (å finne flest moglege analysar for flest mogleg ord) og presisjon (å berre ende opp med korrekte analysar). Viss formålet er å annotere ein leksikografisk ressurs, vil det vere viktigare med høg dekning på bekostning av presisjon, sidan leksikografen gjerne leiter etter

nye/kreative bruksområde av ord. Skal taggaren nyttast til maskinomsetjing i staden, kan ein ikkje nytte meir enn éin analyse til slutt, så her er presisjon viktigast.

Sjølvsagt kan ein her seie at den *ideelle* annotasjonen vil vere å berre ha korrekte analysar, men sjølv ved ideelle krav er formålet viktig: er ein ute etter å finne N-gram som ofte blir omsett med kvarande, men som *ikkje* er syntaktiske konstituentar, er det klart at retningslinjene nedanfor ikkje er så nyttige¹. I tillegg kan ein sjå på kva slags setningspar som er relevante å kunne handtere – skal me annotere setningar som er klart ugrammatiske, som inneheld openberre skrivefeil eller liknande? Rosén & De Smedt (2007, s. 158) skriv i denne samanhengen at «building a treebank is not just a matter of assigning some analysis to everything, but also of making grammaticality judgments»; og å analysere alt berre for å analysere det kan gå mot sitt formål – dette gjeld særleg når ein er ute etter å undersøkje eit *språk*, og ikkje berre eit *korpus*. Analogen til frasesamanstilling blir då at ufullstendige analysar er mindre viktige å handtere, når me er ute etter å byggje ein parallell trebank for språkstudium.

Sidan utviklinga av automatisk frasesamanstilling hovudsakleg har skjedd innanfor frasebasert statistisk maskinomsetjing (PBSMT), kjem me ikkje utanom ei samanlikning her. I PBSMT er formålet med ei fraselenkje å betre maskinomsetjing på eitt eller anna mål, t.d. BLEU-skåren. BLEU-skåren samanliknar ferdig omsett tekst (ein gullstandard) med det automatisk omsette, ved å sjekke kor mykje N-gram-overlapp det er mellom tekstene. Ei lenkje mellom N-grammet *es gibt* og *there is* (dvs. eit auka sannsyn for å nytte slike par i omsetjinga) kan gi ein høgare (betre) endeleg skåre i BLEU. Som vist i Koehn et al. (2003) fekk dei ein *lågare* BLEU-skåre når dei fjerna lenkjer mellom N-gram som, i følgje ein robust statistisk frasestrukturparsar, ikkje var syntaktiske frasar (konstituentar). Dvs. at i figur 3.1 vil lenkja vist ved den prikkete linja bli fjerna frå mengda over moglege lenkjer om ein berre held seg til syntaktiske konstituentar, og *p(es gibt, there is)* vil ikkje bli tilsvarande auka i den statistiske omsetjingsmodellen. Sidan PBSMT, som skildra i Koehn et al. (2003), er agnostisk til syntaktiske høve i omsetjingssteget² er det for dei ingen grunn til å berre halde seg til samanstilling mellom syntaktiske konstituentar; dei har i utgangspunktet meir nytte av kollokasjonsinformasjon.

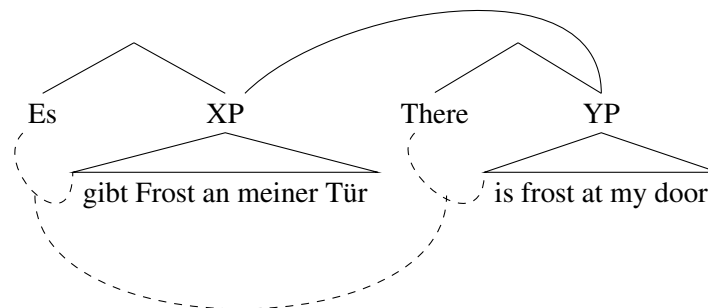
Men sett no at me ikkje har som formål å nytte frasesamanstillinga til reint N-grambasert omsetjing. Kva for *lingvistiske* krav kan me stille til å kalle to frasar samanstilte? Me må i alle fall tillate ein del skilnad. I alle større parallelltekster vil parallellstilte setningar ha visse syntaktiske og semantiske³ omsetjingsskifte, t.d.

¹Eit anna perspektiv ein kan ta er kva for samanstillingar eit utval av tospråklege annotørar meiner er passande – gitt visse overordna retningslinjer for annotasjonen. Volk et al. (2008) operasjonaliserer konseptet *samanstilling* på denne måten, og testar slik kor enkelt det er å følgje dei overordna retningslinjene deira.

²Både omsetjingsmodellen og språkmodellane er reint N-grambaserte her, og har difor ikkje nytte av syntaktisk informasjon (i motsetning til syntaktisk informert generering slik Riezler & Maxwell (2006) implementerer).

³Sidan eg går ut frå at data er setningssamanstilt, kjem eg ikkje inn på diskurs-/pragmatiske verknader, med mindre dette fører til forskjellar innanfor setningane (sjå t.d. del 3.5 om lenkjer mellom koreferente substantiv og pronomen).

todo: referere
til den
faktiske
parsaren? det
var Bikel
kanskje?



Figur 3.1: N-gram-samanstilling versus syntaktiske frasar

leksikalisering av syntaktiske konstruksjonar eller omvendt, endring av ordklasse, presisering/depresisering, endringar i leksikalske trekk (t.d. telleleg/utelleleg), osv. (Munday, 2001, s. 56–62), slik at den einaste fullstendige, «perfekte» samanstillinga vil vere identitetsfunksjonen. Kor mykje mangel på samsvar me godtek blir då avgjort av formålet med samanstillinga.

Eitt av formåla med samanstillinga i denne oppgåva er å kunne oppdage korleis ulike språk realiserer semantiske roller syntaktisk; då spesielt i forhold til hypotesane gitt i XPar (2008, s. 7), t.d. at «case marking might be useful to further determine a given argument’s semantic role». Skal me finne det siste, må me altså kunne lenkje frasar med ulik kasusmarkering, men ha krav om lik tildeling av semantiske roller; samtidig skal me sjå at me ikkje kan ha krav om lik syntaktisk funksjon. I tillegg vil me sjølvstøtt ikkje lenkje på tvers av konstituentgrenser, sidan det er fullstendige konstituentar⁴ som fyller dei semantiske rollene.

Eit anna mogleg formål er å nytte desse frasesamanstillingane til maskinomsetjing. Riezler & Maxwell (2006) nyttar ein frasesamanstilling (fått ved hjelp av N-gramtabellar) til å oppdage overføringsreglar for bruk i LFG-basert generering i maskinomsetjing. Dette er reglar som omsett fragment av ein f-struktur på kjeldespråket til f-strukturfragment på målspråket. (Eit krav på utforminga av moglege overføringsreglar hindrar at ein får reglar som lenkjar ikkje-konstituentar, eg kjem tilbake til dette nedanfor.) Ein liknande metode er Graham & Genabith (2010); Graham et al. (2009). Samanstillinga utvikla her burde au kunne nyttast til å finne slike overføringsreglar, men dette er ikkje noko eg har lagt vekt på.

Nedanfor gir eg eit forslag til krav for frasesamanstilling, med formåla nemnt her i tankane. Om alle krava er moglege å implementere, er eit separat problem.

⁴LFG tillèt som nemnt diskontinuerlege konstituentar, men dette er ikkje det same som ikkje-konstituentar av typen «es gibt» / «there is».

3.3 Frasesamanstilling i ein LFG-trebank

Samanstilte frasar bør ha nok semantisk likskap til å kunne opptre som omsetjingar i liknande omgivelnader (Dyvik et al., 2009, s. 74). Thunes (2003) gir nokre prinsipp – som er passande å ha som utgangspunkt – for å fastslå det som kan kallast *omsetjingsmessig korrespondanse* (her for ordsamanstilling). Dette er prinsipp som skal gjelde for eit litt forskjellig formål, men som au «ligger nær opp til det vi intuitivt mener er riktig» (Thunes, 2003, s. 2). Prinsippa blir nytta til å lage ein gullstandard for ordsamanstilling⁵, hovudsakleg for dei opne klassene, og er definert ved å vise til kva for rolle eit argumentord spelar, eller kva for rolletildeling eit predikat eller modifierande ord gir. Så for å t.d. samanstill to verb må dei ha like mange semantiske argument (men argumenta treng ikkje alle realiserast syntaktisk) og dei må *tildel same roller*; medan argumenta må *spele same rolle*, og både argument og adjunkt må vere *koreferente*. Lenkja ord må vere del av frasar som spelar same rolle i «det som er felles i interpretasjonene av [dei to setningane]» (Thunes, 2003, s. 3).

Viss me tek utgangspunkt i det siste, vil det vere naturleg å i tillegg lenkje desse frasane som spelar same rolle i «det som er felles i interpretasjonene».

Krava for ordsamanstillinga må au vere fylt for at desse frasane kan samanstillast. Ei ordsamanstilling er altså naudsynt for ein frasesamanstilling, og omvendt. Dette er berre problematisk om me føreset at det eine er derivert av det andre; men dette har me ingen *a priori* grunn til å gjere. Krava eg her utviklar bør i staden sjåast på som *skrankar* på moglege samanstillingar i modellen (jamfør 2.2 om modellteoretiske grammatikkar), heller enn derivasjonelle forhold. Samtidig er det som nemnt eit mål å finne ut kor uavhengig me kan gjere oss av ordlenkjingsinformasjonen (dette er au nyttig for implementasjonen), utan at det treng å gi krava ei *retning*.

Ei frasesamanstilling er ei skildring av forhold mellom *fragment* av setningar, dette er endå ein grunn til at det er naturleg å skildre dei ønskelege forholda som skrankar på moglege samanstillingar. Me kan setje skrankar på f-struktur-, konstituent- og ordsamanstilling samtidig, utan å måtte ha krav om at den eine samanstillinga er fullstendig (eller delvis) avleia av den andre, før me veit om eit slikt avleiingsforhold er empirisk fundert. Me kan i tillegg ha ufullstendige samanstillingar i dei tilfella der det er ufullstendig samsvar mellom setningane (der ei fullstendig samanstilling ville brutt visse krav).

Sidan metoden er mynta på bruk i ein LFG-parsa trebank, og delvis vil nytte denne annotasjonen som datagrunnlag, er det naturleg å nytte same konsept som

⁵(Thunes, 2003, s. 2): «Våre prinsipper er satt opp for å tjene et bestemt formål, nemlig å samle inn data som metoden i Semantic Mirrors skal anvendes på», ein metode for å automatisk finne WordNet-liknande relasjonar frå parallelltekst. I denne metoden vil det vere naturleg med høge krav til presisjon, men kanskje lågare krav til dekning: speilmetoden skal finne leksikalske semantiske forhold som held på *typenivå*, medan for trebanken er det viktigare korleis me kan annotere eit *token* av t.d. eit verb i ein viss VP i ei gitt korpussetning.

blir nytta i LFG⁶ (f-struktur, c-struktur, endosentrisitetsprinsipp, \bar{X} -tre, osv.) au i desse krava til den «beste» frasesamanstillinga; i den grad LFG gir ein generaliserbar skildring av syntaks, bør desse krava vere generaliserbare til andre teoriar, men ein del forhold som er avleidd av LFG-prinsipp må sjølvstendig modifiserast om krava skal generaliserast til andre teoriar.

Utan skrankar i det heile vil alt kunne lenkjast til alt (noko som er like unyttig som å ikkje lenkje noko); i del 3.4 ser eg på kva for typar element i dei lingvistiske analysane (ord, grammatiske trekk, konstituentar, ...) det er fornuftig å tillate lenkjer mellom. I avsnitta nedanfor spesifiserer eg kva som må til for at me skal lenkje element av desse typane.

3.4 Kva kan lenkjast?

Viss to uttrykk er samanstilt på setningsnivå (slik at me dimed kan gå ut frå at dei er omsetjingar av kvarandre), og begge har ein LFG-analyse, så har me iallfall tri ulike nivå kor me kan finne ekvivalensforhold under setningsnivå:

1. mellom ord i setningane,
2. mellom f-strukturar,
3. mellom c-strukturnodar.

På begge språk har me alle nivå – det er ingen grunn til å lenkje på tvers av nivå sidan forhold mellom desse nivåa er implisitt i LFG-analysen.

Alle ord i setninga er *kandidatar* for samanstilling med ord i omsetjinga, men det kan godt hende at eit ord *ikkje* har ei lenkje, og me kan heller ikkje utelukke at det finst mange-til-mange-lenkjer som ikkje kan «delast opp». Dette gjeld au nodane i c-strukturen.

Me utelukker lenkjing av ikkje-konstituentar som *there is* på c-strukturnivå sidan ei lenkje mellom to c-strukturnodar impliserer at heile frasen under er lenkja. Det finst ingen c-strukturnodar som dominerer berre *there, is* og ingen andre ord (heller ikkje *es, gibr*), så dette er ikkje lenjekandidatar. *There is* og *Es gibr* i figur 3.1 kan då ikkje samanstillast åleine, men berre som del av ei ytre frasesamanstilling⁷.

Når det gjeld f-strukturane er det ganske mange element me teoretisk sett kunne ha lenkja, t.d. enkelttrekk som kasus eller dei uordna mengdene med adjunkt, men det som er mest *nyttig* og *meningsfullt* er nok å berre lenkje der det er ei nær kopling til orda i setninga. Sidan alle PRED-element i ein f-struktur unikt står for

⁶I tillegg finst andre positive biverknader av ein LFG-basert frasesamanstilling for bruk i denne samanhengen, som at ein kan studere kor parallelle dei parallelle grammatikkane i ParGram-prosjektet (Butt et al., 2002) faktisk er, på ulike nivå (leksikon og argumentstruktur, c-struktur, f-struktur).

⁷Slike forhold kan me sjølvstendig finne igjen etter lenkjinga, men då vil me au kunne generalisere til andre ordformer. Eg kjem tilbake til dette i kapittel 5.

predikerande ord, kan me – gitt to samanstilte setningar – la *kandidatane for samanstilling på f-strukturnivå* inkludere alle desse PRED-elementa i f-strukturane til setningane⁸. PRED-element representerer semantiske bidrag som oftast er påkrevde på begge språk i omsetjingar, medan andre f-strukturtrekk gjerne er valfrie på det eine av språka; det er ikkje alle språk som har t.d. obligatorisk kasusmarkering, og ein vil kanskje nytte trebanken til å oppdage nettopp slik variasjon. PRED-elementa er i tillegg gjerne enklare å knyte direkte opp mot den konkrete, observerte tekststrengen (eventuelt testast mot korpora, eller talarintuisjonar), medan t.d. eit trekk som aspekt kanskje er umogleg å skilje frå tempus i affikset (det vil vere vanskelegare å teste om ei lenkje mellom aspekt-trekk er empirisk motivert utan å dra inn ein heil del teori).

Samtidig er det au eit omsetjingsforhold mellom trekka i same f-struktur som dei lenkja PRED-elementa, og me ville kanskje ikkje ha omsett dei to PRED-elementa i andre f-strukturkontekstar. Difor bør me au sjå på ei PRED-lenkje som ei lenkje mellom *f-strukturane til desse PRED-elementa*⁹. Med dette i tankane, kombinert med c-struktur-f-strukturavbildinga ϕ (sjå del 2.2), får me følgjande samanheng, illustrert i figur 3.2:

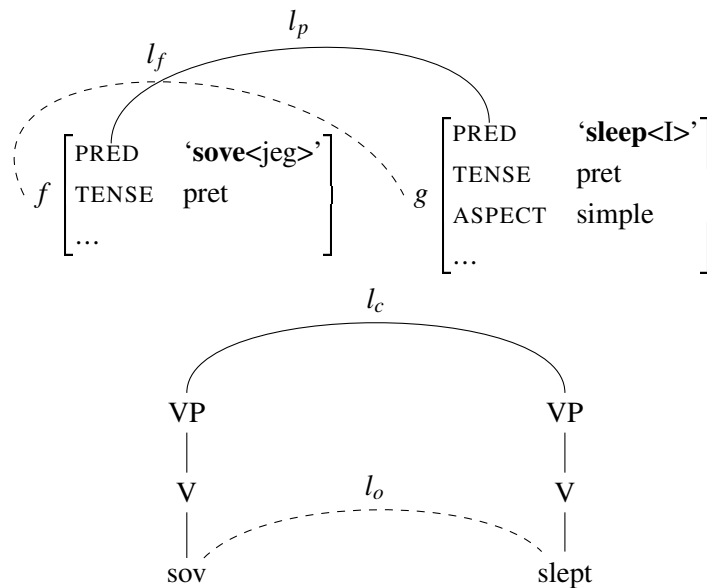
- (1) Ei lenkje mellom to PRED-element p og q , kor p er medlem av f-strukturen f , og q er medlem av f-strukturen g , tilseier at:
 - a. me tolkar f-strukturane f og g som lenkja,
 - b. orda i setningane som projiserer PRED-elementa tek del i ei lenkje (kor andre ord kan vere involvert), og at
 - c. nodar innanfor $\phi^{-1}(f)$ og $\phi^{-1}(g)$, dei funksjonelle domena til f-strukturane f og g , kan lenkjast

Punkt (1-a) og (1-c) over seier at viss PRED-elementa projisert av t.d. to verb i verbfrasar er lenkja, kan VP-ane som heilskap lenkjast, i tilfellet i figur 3.2 kan iallfall dei øvste nodane i VP-ane lenkjast, i tillegg til f-strukturane frå ytre PRED til verba. Det er dette at heile VP-ane (kanskje inkludert objekt) er lenkja som gjer det til ei fraselenkje og ikkje berre ei ordlenkje. Punkt (1-a) er forsvart over, medan punkt (1-c) kjem som ein konsekvens av at det er det funksjonelle domenet som spesifiserer informasjonen i f-strukturane, nodane her bør difor lenkjast berre viss f-strukturane er lenkja. Men som punkt (1-c) indikerer finst det au situasjonar der nodar innanfor domena skal stå ulenkja.

Alle nodar i c-strukturen (alle syntaktiske *frasar/konstituentar* i setninga) som kan koplast til PRED-haldande f-strukturar, vil vere kandidatar for samanstilling på c-strukturnivå (dette inkluderer diskontinuerlege konstituentar), men ikkje alle vil bli lenkja. I del 3.7 ser eg på kva som må til for å lenkje nodar i det funksjonelle domenet. I tillegg finst det nodar over ord som ikkje projiserer PRED-element, desse

⁸I del 3.7.3 kjem eg tilbake til spørsmålet om me vil inkludere visse f-strukturar utan PRED-element i kandidatane for samanstilling.

⁹Eventuelt kunne me ha definert lenkjingskandidatane på f-strukturnivå som alle PRED-haldande f-strukturar, resultatet blir det same.



TODO: teikne inn f-domene

Figur 3.2: Ei PRED-lenkje l_p kan tolkast som ei f-strukturlenkje l_f , og impliserer ei c-strukturlenkje l_c mellom toppnodane i dei funksjonelle domena. Orda som projiserer PRED-elementa er med i ei lenkje l_o (som kan inkludere fleire ord).

kjem eg tilbake til i del 3.7.3.

I følge punkt (1-b) vil fraselenkja leie til at sjølve verba i to lenkja VP-ar au er lenkja, som tilseier at *ei PRED-lenkje impliserer ei ordlenkje*. I visse tilfelle er dette heilt uproblematisk, t.d. viss *I slept down by the river* skal lenkjast med *Eg sov nede med elva* vil me uansett lenkje *slept* og *sov*; dette kan gjelde transitive verb au:

- (2) a. The locusts have no king, just noise and hard language
 \leftrightarrow
 b. Grashoppene har ingen konge, berre støy og krasse ord

have/har tek del i VP-samanstillinga *have no king.../har ingen konge...*, her au skal det vere uproblematisk å lenkje enkeltorda *have* og *har*.

Men som nemnd treng ikkje ordsamanstillinga vere ein-til-ein, det punkt (1-b) seier er at desse orda iallfall er ein del av ein samanstilling med kvarandre (i døme (2) altså VP-samanstillinga). Kanskje er dette ei mange-til-mange-lenkje som ikkje *kan* reduserast til ein-til-ein-lenkjer; eller kanskje er det som i (2) mogleg å skilje ut delsamanstillingar, som *have/har*. Eg kjem tilbake til dette **TODO: når?** seinare.

Sidan PRED-lenkjing impliserer ordlenkjing, må me sjekke om krava på ord-nivå (del 3.5) er oppfylte for å lenkje to PRED-element. **TODO: litt brå avslutning**

3.5 Krav på ordnivå

Ord som skal lenkjast må i Thunes (2003) vere del av frasar som spelar same rolle i det som er felles i interpretasjonane, her kan me omskrive det til at dei må vere del av *frasar som er lenkja på c-strukturnivå*; forholda i (1) gir då koplinga til krav på andre nivå (t.d. vil krav om tildeling av like mange roller vere meir passende å spesifisere på f-strukturnivå).

Det er visse ting me ikkje kan spesifisere ut frå rein c- og f-strukturinformasjon. Den norske setninga *eg vil ete* kan fint samanstillast med *I want to eat*, med ei lenkje mellom *ete* og *eat*. Men kva står i vegen for å lenkje *ete* til hovud verbet i *I want to drink*? Forskjellen på f-strukturnivå er berre at PRED-verdien er ulik (**eat** mot **drink**). Me må altså ha eit krav om at tydinga til lenkja ord (og deira predikat) er «lik nok» til at me kan sjå på dei som omsetjingar¹⁰. Dyvik et al. (2009, s. 74) krev at orda generelt, utan kontekst, må vere semantisk plausible omsetjingar, dvs. at målordet er eit medlem av mengda av *linguistically predictable translations* av kjeldeordet. Målordet har då *LPT-korrespondanse* med kjeldeordet. Nedanfor reknar eg LPT-kravet som eit krav på ordnivå, og eg føreset at LPT-informasjonen er ein type bottom-up-informasjon, som viser om to ord generelt (i ulike kontekstar) blir nytta som omsetjingar av kvarandre. Denne informasjonen kan reint praktisk komme frå automatisk ordsamanstilling, eller ei god tospråkleg ordbok, det bør ikkje spele nokon rolle for resten av krava¹¹.

Ein type presisering/depresisering (del 3.2) me ofte ser i omsetjingar er at eit pronomen på kjeldespråket blir nytta der målspråket har eit koreferent substantiv, eller omvendt. Dyvik et al. (2009) opnar for at desse au har LPT-korrespondanse (som nemnt i Thunes (2003) må lenkja ord uansett vere koreferente); om formålet vårt var maskinomsetjing heller enn å byggje ein trebank for lingvistiske studie, ville det nok vore betre å unngå slike lenkjer (Volk et al., 2008, s. 53).

TODO: Er det mogleg å presisere LPT-kravet meir? Skal det berre vere eit range-ringskrav??

Men kva då med lenkjing av pronomen til verb bøygd for person og tal i pro-drop-språk?

- (3) a. iqePa (georgisk)
 \leftrightarrow
 b. han bjeffa

Viss setningane i dømme (3) er lenkja, der *iqePa* har eit pro-argument koreferent med *han* som subjekt, bør dei to subjekta iallfall kunne lenkjast på f-strukturnivå; dei har same referent og spelar same rolle i argumentstrukturen til verba (som me går ut frå er lenkja). På ordnivå, derimot, kan me ikkje lenkje *han* til *iqePa* åleine – her må me ha ei mange-til-ein-lenkje mellom {*han*, *bjeffa*} og {*iqePa*}. Generelt må

¹⁰Eigentleg burde slike setningar ikkje vere lenkja på setningsnivå ein gong, men som me skal sjå i del 3.6.1 treng me kravet om lik tyding sjølv innanfor setninga.

¹¹Ein kan au tenkje seg at ei djup semantisk dekomponering av kvart ord sto som grunnlag for LPT-informasjon – men då vil LPT-korrespondanse mellom to ord implisere at orda er synonyme, heller enn generelt plausible omsetjingar.

me ha slike lenkjer der eitt ord projiserer fleire PRED-element¹².

3.5.1 Ordklasse

Ulike språk leksikaliserer same konsept på ulike måtar. Cheung et al. (2002, s. 3) nemnar vanskane med å ha eit krav om lik ordklasse i utviklinga av ein kinesisk-engelsk termbank, kor t.d. det engelske ordet *fulfilment* meir naturleg blir omsett til eit verb på kinesisk. På same måte vil eit georgisk verbalsubstantiv (*masdar*) gjerne bli omsett til eit verb i infinitiv på norsk. Slike skifte mellom ordklasser er svært vanlege i omsetjing¹³.

Me kan opne for ordklasseoverskridande lenkjer der det er samsvar på andre nivå, me bør iallfall krevje ein likskap i argumentstruktur; så om LPT-kravet og krava på c- og f-strukturnivå er fylt, bør det ikkje vere noko i vegen for å lenkje ord (eventuelt mengder av ord) av ulik ordklasse.

3.6 Krav på f-strukturnivå

På f-strukturnivå har me direkte tilgang til informasjon om argumentstrukturen til eit predikat, og mengda av adjunkt som modifierer predikatet. Når Thunes (2003, s. 3) skriv at to lenkja ord *a* og *b* må opptre i frasar som har «tilstrekkelig like argumentstrukturer til at uttrykkene i *as* omgivelser står i de samme semantiske relasjonene til hverandre og til *a* som de korresponderende uttrykkene i *bs* omgivelser gjør til hverandre og til *b*» er det difor passende å prøve å gjere dette til eit krav på f-strukturnivå.

Den enklaste lenkjingssituasjonen, f-strukturmessig, er der rotpredikata kan lenkjast, og første argument av predikatet på kjeldespråket kan lenkjast til første argument på målspråket, andre argument til andre argument, osv., og lenkjinga kan fortsetje slik rekursivt inn i f-strukturane¹⁴. I ein slik situasjon er det fullstendig samsvar mellom kor mange argument det finst på kvar side, og fullstendig samsvar i det tematiske rollehierarkiet (dvs. kva for posisjon kvar rolle har i argumentstrukturen), i heile strukturen.

Som me skal sjå er det ikkje vanskeleg å komme over situasjonar der dette ikkje held, og me blir nøydte til å tillate lenkjer mellom argument og adjunkt, og lenkjer som går på tvers av følgja i argumentstrukturane. I tillegg kan me ikkje klare oss utan LPT-informasjon for å avgjere *når* me har å gjere med slike meir komplekse situasjonar.

¹²Me ville au fått ei mange-til-ein-lenkje om me tillot *komplekse predikat* i analysane, t.d. slik Butt (1998) foreslår ved å la kombinasjonen av to ord endre argumentstrukturen til eitt PRED-element.

¹³Munday (Catford (1965), i 2001, s. 61) gir ein gjennomgang av slike *klasseskifte*, og andre typar omsetjingsskifte.

¹⁴Når eg her skriv *argument*, meiner eg eigentleg alle subkategoriserte ledd, både innanfor og utanfor sjølve argumentlista. I analysen av «eska opna seg» vil *eske* stå innanfor argumentlista, medan refleksiven står utanfor: '**opne**<eske>seg'. Begge er likevel subkategorisert for; begge er krevd av (den tydinga av) predikatet.

Thunes (2003) gir som nemnd eit krav om at *predikat må ha tilsvarende semantiske argument* for å lenkjast.

Sett at ei setning på språk 1 har ei *at*-setning som adjunkt, medan denne setninga på språk 2 er eit argument, og at desse setningane ville vore lenkja om dei opptredde åleine. Om dei uttrykkjer same proposisjon og *speler same rolle i verbsituasjonen*, synest det naturleg å lenkje desse.

For å gjere dette konkret kan me sjå på følgjande setning frå test-suiten til Xpar-prosjektet:

- I følge LFG-parsen til desse setningane har hovudpredikata svært ulike argumentstruktur¹⁵. Det norske *vedde* har fire argument, medan *da-najlebeba* har to (*Abrams* og *Browne*), kor at-setninga på norsk og *rom cvimda* uttrykkjer same proposisjon og spelar same rolle i verbsituasjonen. Den engelske LFG-parsen av den tilsvarende setninga (mine omsetjingar) gir tri argument, *with* blir her adjunkt, medan den tyske grammatikken, som au har tri argument, gjer *at*-setninga til adjunkt. I (5) nedanfor har eg representert dei omsetjingsmessig korresponderande frasane i f-strukturane med dei norske omsetjingane for å illustrere dette:

- [
 PRED ‘**vedde**<Abrams, sigarett, Browne, regne>’
 ADJUNCT {}
]

- ¹⁵Analysane er henta 18. mai, 2009, frå <http://decentius.aksis.uib.no/logon/xle.xml>, som implementerer LFG-grammatikkane frå ParGram-prosjektet (Butt et al., 2002).

$$\left[\begin{array}{ll} \text{PRED} & \text{'da-najleveba<Abrams, Browne, regne>'} \\ \text{ADJUNCT} & \{ \text{sigarett} \} \end{array} \right]$$

- c. Abrams hat mit Browne um eine Zigarette gewettet, (tysk)
daß es regnet.

$$\left[\begin{array}{ll} \text{PRED} & \text{'wetten<Abrams, regne>'} \\ \text{ADJUNCT} & \{ \text{Browne, sigarett} \} \end{array} \right]$$

- d. Abrams bet a cigarette with Brown that it was raining. (engelsk)

$$\left[\begin{array}{ll} \text{PRED} & \text{'bet<Abrams, sigarett, regne>'} \\ \text{ADJUNCT} & \{ \text{Browne} \} \end{array} \right]$$

Om ein skal ha grammatikkane som datagrunnlag er det altså eit reellt problem kva ein skal gjere med mangel på samsvar i argumentstruktur. Om det alltid var fullstendig samsvar i argumentstruktur, ville det vore trivielt å lenkje argument: viss to korresponderande verb hadde tri argument, ville me lenkja det første med det første, det andre med det andre og det tredje med det tredje. Men om me har analysar som dei over, ser det ut til at me er avhengig av LPT-kravet frå del 3.5 for å avgjere kva for adjunkt og argument som samsvarer.

LPT-kravet blir forresten endå viktigare når det gjeld lenkjing av adjunkt til adjunkt. Adjunkt plukker ut si eiga rolle (argument får rolla tildelt frå verbet) og f-strukturane ordnar ikkje adjunkt etter nokon rekkjefølgje, dei er representert som uordna mengder, medan følgja mellom argument iallfall potensielt kan nyttast til å indikere semantisk likskap.

Ein kan argumentere for at grammatikkane her *burde* hatt like (eller likare) analysar, dette ville letta lenkjingsarbeidet, men sidan stoda no er slik, må krava ta høgd for lenkjer mellom argument og adjunkt. Om seinare utgåver av grammatikkane gir likare analysar, vil det iallfall ikkje gi verre lenkjingsresultat.

Og ei enkel korpusundersøking tyder på at det er relativt sjeldan at ein får slike situasjonar som (5) illustrerer. I Unhammer (2009) analyserte eg setningane frå den manuelt frasesamanstilte trebanken SMULTRON (Samuelsson & Volk, 2006) med LFG-grammatikkane for engelsk og tysk i ParGram-prosjektet (Butt et al., 2002), for å undersøkje følgjande hypotese:

participants in a verbal situation are expressed as arguments (rather than adjuncts) in the source language of a translation if and only if they are expressed as arguments (rather than adjuncts) in the target language.

Mellom anna fann eg at 2 av 15 korresponderande verbtoken hadde LFG-analysar kor argument korrespondererte med adjunkt¹⁶. Her utgjorde altså dei gram-

¹⁶25 om ein inkluderer analysar kor minst eitt av argumenta ikkje hadde korrekt analyse (t.d. eit

matriske analysane (ein del av) data, og undersøkinga seier nok meir om analysane enn om språklege forhold. På et så tynt datagrunnlag kan me vel berre konstatere at me må kunne handtere argument-adjunkt-lenkjer når me prøver å lenkje, men argument-argument-lenkjer bør prioriterast viss alt anna er likt.

3.6.2 Ulik følgje i argumentstruktur

I tillegg til at argument kan lenkjast til adjunkt, kan koreferente argument ha ulik følgje i argumentstrukturen. Det er klart at me vil lenkje objektet til *gefallen* (eller bokmål: *behage*) med subjektet til *like*, og omvendt. Men rekkjefølgje i argumentstrukturane i ParGram-prosjektet er ofte basert på syntaktisk funksjon heller enn rolle, slik at eit verb som har tema som subjekt og opplevar som objekt vil ha tema før opplevar i argumentstrukturen, medan ei omsetjing av dette verbet kan ha opplevar før tema:

- (6) a. der Tonfall gefällt mir nicht
 $\left[\text{PRED} \quad \text{'gefallen'} \langle \text{Tonfall}, \text{ich}_i \rangle \dots \right]$
- \leftrightarrow
- b. jeg liker ikke tonen
 $\left[\text{PRED} \quad \text{'like'} \langle \text{jeg}_i, \text{tonen} \rangle \dots \right]$

Argumentstrukturane i (6) har omvendt intern følgje. Igjen må me ha LPT-informasjon for å avgjere kva for lenkjing som er korrekt. Men i visse tilfelle vil ikkje ein gong LPT-informasjon vere nok:

- (7) a. sie_j gefallen ihnen_i
 $\left[\text{PRED} \quad \text{'gefallen'} \langle \text{de}_j, \text{de}_i \rangle \dots \right]$
- \leftrightarrow
- b. de_i liker dem_j
 $\left[\text{PRED} \quad \text{'like'} \langle \text{de}_i, \text{de}_j \rangle \dots \right]$

Det finst ingen f-strukturinformasjon eller LPT-informasjon me kunne nytta til å sikre den korrekte lenkjinga *sie/dem* og *ihnen/de*; og viss me rangerer lik argumentstruktur over ulik, vil me her få feil resultat. Det me *kan* gjere (utanom å endre grammatikkane slik at argumentstruktur korresponderer med eit universelt tematisk rollehierarki) er å sjå på mange lenkjingar av same verbpar, og på den måten oppdage moglege feil. For enkelttilfelle, derimot, vil krava i denne oppgåva ikkje vere nok til å gi korrekt lenkjing.

PRO der grammatikken burde funne eit substantiv).

3.6.3 Krav om argumentlenkjer

Sjølv om me ikkje krev lik følgje i argumentlenkjer, og tillèt argument-adjunktlenkjer, er det eit minstekrav for å lenkje to PRED-element at alle argumenta til det eine PRED-elementet kan korrespondere med argument eller adjunkt av det andre PRED-elementet. Dette følgjer av formålet med å finne ut korleis ulike språk realiserer ulike semantiske roller syntaktisk; om eit verbargument ikkje kan lenkjast til noko i omsetjinga (ikkje ein gong eit pro-element), er det usannsynleg at verba uttrykker same situasjon, og tildeler same roller. På same måte må sjølvstgått lenkja predikat ha LPT-korrespondanse. Dyvik et al. (2009, s. 75) gir følgjande krav på f-strukturnivå¹⁷:

- (8) Krav for lenkjing av to PRED-element p og q :
- ordformene til p og q har LPT-korrespondanse
 - alle argument av p har LPT-korrespondanse med eit argument eller adjunkt av q
 - alle argument av q har LPT-korrespondanse med eit argument eller adjunkt av p
 - LPT-korrespondansane kan lenkjast ein-til-ein
 - ingen adjunkt til p er lenkja til f-strukturar utanfor q , og omvendt

Det (8-d) seier er at me ikkje lenkjer t.d. to instansar av «hest» på det eine språket til éin instans av «horse» på det andre. Krav (8-e) kjem eg tilbake til nedanfor.

Det går an å gjere (8) strengare, og krevje at argumenta – i tillegg til å ha LPT-korrespondanse – sjølv er PRED-lenkja. Dette har eg ikkje gjort i implementasjonen min, men det er mogleg å ha det som eit rangeringskriterium, noko eg kjem tilbake til i del 3.8. Ved å *ikkje* krevje at lenkjinga går heilt til botn i f-strukturen blir det mogleg å seie at *setningane* er syntaktisk like, og at kanskje visse overordna frasar er syntaktisk like, men visse *delfrasar* kan likevel vere ulike og dimed ikkje vere lenkja.

Koordineringar har ikkje eit PRED-trekk, men me handsamar dei som om dei hadde det. Alle dei koordinerte elementa er i ei *mengd* i f-strukturen til koordineringa, og det er sjølvstgått ønskeleg å lenkje desse elementa om dei korresponderer:

- (9) Ved lenkjing av f-strukturane til to koordineringar p og q , sjå på dei som om elementa i mengdene var argument til eit PRED-element, kor «argumentfølgja» er basert på setningsposisjon; p og q kan då lenkjast om dei oppfyller krav (8).

Argumentfølgje spelar berre ei rolle i rangering, som eg kjem tilbake til i del 3.8.

Kva med f-strukturomgivingane til p og q , skal me krevje at dei er like? I (9-e) har me eit krav om at adjunkt til p ikkje er lenkja til f-strukturar utanfor q , og omvendt. Men viss a_p er eit adjunkt til p , kan det lenkjast til ein *dotternode* av argument eller adjunkt til q ? La a_q vere eit argument eller adjunkt til q , viss a_q

¹⁷Med eit unntak for adposisjonsobjekt som eg kjem tilbake til i del 3.6.4.

er eit argument må det ved (9) ha LPT-korrespondanse med argument/adjunkt i p , men det treng ikkje vere lenkja – viss det er ulenkja gjeld ikkje krav (9) for a_q , så (9) hindrar ikkje ei lenkje mellom a_p og døtre av a_q .

I tillegg vil ikkje (9) hindre at t.d. den yttarste f-strukturen i kjeldespråket er lenkja til eit XCOMP-argument på målspråket; men i dette tilfellet bør kanskje ikkje *setningane* vere lenkja i utgangspunktet.

Sjølv om det er logisk mogleg å gjere slike lenkjingar, er det vanskeleg å finne ikkje-vilkårlege avgrensingar for når ein skal kunne lenkje f-strukturar som står i ulike omgivnader. I implementasjonen min har eg difor følgd eit strengare krav enn (9-e):

- (10) PRED-elementa p og q kan berre lenkjast om dei er yttarste f-strukturar i lenkja setningar, eller er argument/adjunkt til lenkja f-strukturar.

Dette er ei tentativ formulering. Til no har eg ikkje sett døme kor (10) ikkje bør gjelde, men om det finst slike døme bør sjølv sagt kravet modifierast. Sidan LFG tillèt fragmentariske analysar kan det vere *fleire* yttarste f-strukturar, alle desse kan då potensielt lenkjast med kvarandre, eller stå ulenkja (som om dei var adjunkt av eit predikat som sto utanfor dei).

Krav (8) og (10) bør i enkle situasjonar vere tilstrekkelege for lenkjing på f-strukturnivå, men det finst au meir komplekse korrespondansar mellom PRED-element. Desse ser eg på del 3.6.5.

3.6.4 Adposisjonsobjekt

I setningsparet i (4) har me eit objekt *sigarett* som svarer til PP-en *sigaretze (sigareti + ze)*, som i (11) nedanfor:

- (11) a. $\left[\text{PRED} \quad \text{'sigarett'} \right]$
- \leftrightarrow
- b. $\left[\begin{array}{cc} \text{PRED} & \text{'ze} \langle \boxed{1} \rangle \\ \text{OBJ} & \boxed{1} \left[\text{PRED} \quad \text{'sigareti'} \right] \end{array} \right]$

Medan **'sigarett'** er argument til **'vedde'**, står det ein adposisjon mellom **'sigareti'** og **'da-najleveba'**. I følgje krav (8) må me ha LPT-korrespondanse mellom **'sigarett'** og eit argument/adjunkt av **'da-najleveba'** for å lenkje **'vedde'** og **'da-najleveba'**, det har me ikkje – det står ein adposisjon i vegen.

Éi løysing ville vore å mange-mange-lenkje **'sigarett'** med **'sigareti'** og **'ze'** – men dette gir ei misvisande lenkje, sidan **'sigarett'** ikkje bidreg med noko som tilsvare den (syntaktiske) informasjonen som er gitt av **'ze'**.

Løysinga valt i Dyvik et al. (2009, s. 75, fotnote 3), som eg følgjer i implementasjonen, er å berre hoppe over slike adposisjonar. Ved lenkjing av **'vedde'**

og ‘**da-najleveba**’ ser me då på f-strukturane i (11) som om dei var som i (12) nedanfor.

(12) a. $\left[\text{PRED} \quad \text{‘sigarett’} \right]$

\leftrightarrow

b. $\left[\text{PRED} \quad \text{‘sigareti’} \right]$

Dette må ein altså ha i mente når ein følgjer krav (8). I neste del diskuterer eg kva me kan gjere i dei situasjonane der det ikkje er mogleg å berre hoppe over mellomliggande element.

3.6.5 Kausativar og inkorporering

Til no har me føresett at eit PRED-element anten er ulenkja, eller er lenkja til eitt og berre eitt anna PRED-element. Men i visse tilfelle kan det vere ønskeleg å lenkje til fleire PRED-element.

TODO: dette
gjeld mange
andre
fenomen au,
fryse ned,

I ein norsk *la*-konstruksjon, t.d. den me har i «å la noko fryse» (i tydinga å forårsake at noko frys til) har me semantiske bidrag frå både *la* og hovudverbet *fryse*, og begge har PRED-element (sjølv om bidraget frå *la* nok er meir «grammatisk»). Men slike perifrastiske konstruksjonar kan gjerne omsetjast til leksikaliserte kausativar som berre har eitt PRED-element, men likevel med tydinga «å la fryse». Påfunnet i (13) illustrerer denne situasjonen:

(13) a. ho lar-fryse huset
 $\left[\text{PRED} \quad \text{‘la-fryse<ho, hus>’} \right]$

\leftrightarrow

b. ho lar huset fryse
 $\left[\begin{array}{l} \text{PRED} \quad \text{‘la<ho, hus, [1]>’} \\ \text{XCOMP} \quad [1] \left[\text{PRED} \quad \text{‘fryse<hus>’} \right] \end{array} \right]$

Her er altså den kausative tydinga leksikalisert, og verbet har berre eitt PRED-element (på same måte som det norske verbet *kjøre* berre har eitt PRED-element, ikkje *la + bli kald*).¹⁸

Den same situasjonen får me der eit argument eller adjunkt er inkorporert i verbet på det eine språket, men uttrykt som eit separat predikat på det andre språket, t.d. samisk *fierpmástallat* som på norsk blir *å fiske med garn* – to predikat på norsk tilsvarer eitt på samisk.

¹⁸Det går sjølvsagt an å analysere sjølv leksikaliserte kausativar som om dei har fleire PRED-element, men det bør i såfall skje på uavhengig grunnlag, ikkje for å gjere lenkjinga enklare.

I (13) har *la-fryse* to argument, som ved krav (8) begge må finne korresponderande argument eller adjunkt for å lenkje *la-fryse*. Då går det ikkje an å lenkje *la-fryse* til berre *fryse*, som har eitt argument; me får eit XCOMP til overs som manglar lenkje. Me kan heller ikkje lenkje berre *la* til *la-fryse*, sidan det då får ein XCOMP til overs.

Det er mogleg å løyse dette formelt ved ei mange-mange-lenkje, kor ein tenkjer seg *la* og *fryse* som samanføyde og at dei deler argumentlister. Sidan begge verba tilfører viktig semantisk informasjon, som er reflektert i den leksikaliserte kausativen, ville det ikkje vore ønskeleg med ei ein-til-ein-lenkje sjølv om ein såg vekk frå problemet med å lenkje argumenta.

Ved å ha ei ein-mange-lenkje, frå *la-fryse* til både *la* og *fryse*, kan me oppfylle krav (8). Då treng ikkje XCOMP-argumentet lenkjast til eit argument av *la-fryse*, det er allereie lenkja til PRED-elementet; det som står igjen er unionen av argumenta til *la* og *fryse*, desse må alle ha LPT-korrespondanse med argument eller adjunkt av *la-fryse*, og omvendt må alle argument av *la-fryse* ha LPT-korrespondanse med argument eller adjunkt av *la* eller *fryse* (utanom XCOMP-argumentet til *la*, som allereie har ei lenkje). Ein kan tolke dette som om *la* og *fryse* var samanføyde til eitt predikat som krevde to argument (her: *ho* og *huset*).

Den einaste formelle forskjellen mellom dette og substantivinkorporering blir då at substantivet ikkje krev eigne argument. Det er au mogleg å tenkje seg ein kausativ med eit inkorporert objekt, omsett til *la* + *hovudverb* + *objekt*, altså ei lenkje frå eitt PRED til tri PRED. Igjen vil me då sjå på dei resterande ulenkja argumenta på kvar side; kvar av desse må lenkjast med eit unikt argument eller adjunkt.

Men det bør kanskje vere grenser for kor langt slik samanføyning kan gå, om ikkje anna fordi problemet fort blir komputasjonelt vanskeleg. Å opne for ein-mange-lenkjer mellom PRED-element (eller til og med mange-mange-lenkjer) gir ei mykje større mengd moglege løysingar på lenkjingsproblemet; i alle situasjonar der me krev LPT-korrespondanse mellom eit argument a_p av p og eit adjunkt a_q av q for å lenkje p og q , vil me no au ha ei mogleg løysing der a_q er ulenkja, medan a_p er samanføyde med p og difor ikkje treng LPT-korrespondanse med argument/adjunkt av q . Så kan det au hende at a_p sjølv kan samanføyast med eit av sine argument/adjunkt. Skal me sjå etter slike løysingar samtidig som me ser etter løysingar med ein-ein-lenkjer, vil me måtte leite gjennom mange ufruktbare stigar. Ein måte å unngå dette på er å nedprioritere samanføyning, og berre prøve dette der det ikkje finst andre alternativ.

Men det er ikkje berre av omsyn til implementasjonen ein bør nedprioritere desse. Ei ein-mange-lenkje tyder på ein type omsetjingsskifte, og det er ønskeleg å først sjå etter samanstillingar som føreset syntaktisk likskap, før ein ser etter omsetjingsskifte. Den viktigaste informasjonen me har å gå på er at setningane er omsetjingar og difor har ein viss likskap – Ockhams barberkniv gir oss då grunn til å velje ei løysing som føreset lik syntaks over ei løysing som føreset ulik syntaks. Viss det er mogleg å opprette ei samanstilling på bakgrunn av lik syntaks, vil me prioritere denne.

I implementasjonen blir difor alle ein-til-ein-lenkjer prøvd først. Sidan kan ein prøve å føye saman eit ulenkja PRED-element p med eit ulenkja PRED-element a_p kor a_p er argument eller adjunkt av p , og der p og a_p vil kunne lenkjast med eit ulenkja PRED-element q ved føringane gitt over, og alle dei andre lenkjingskrava er dekkja. Me får då ei modifisert utgåve av krav (8):

- (14) Krav for samanføyd lenkjing frå PRED-elementa p og a_p , kor a_p er eit argument eller adjunkt av p , til PRED-elementet q :
- a. ordformene til p og a_p har saman LPT-korrespondanse med ordformen til q
 - b. la A vere unionen av argument til p og argument til a_p , utanom a_p sjølv; alle element av A har LPT-korrespondanse med eit argument eller adjunkt av q
 - c. la D vere unionen av argument eller adjunkt til p og argument eller adjunkt til a_p , utanom a_p sjølv; alle argument av q har LPT-korrespondanse med eit element av D
 - d. LPT-korrespondansane er ein-til-ein
 - e. ingen adjunkt til p eller a_p er lenkja til f-strukturar utanfor q , og ingen adjunkt til q er lenkja til f-strukturar utanfor p

Det er trivielt å utvide dette kravet til å fungere for mange-mange-lenkjer au; men til no har eg ikkje komme over situasjonar som krev meir enn ein-mange/mange-ein-lenkjer, og implementasjonen min held seg til desse for no.

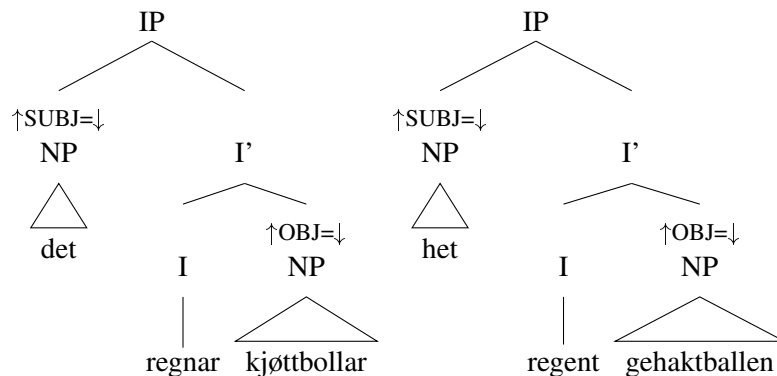
3.7 Krav på c-strukturnivå

Ein f-struktur er projisert av ei mengd c-strukturnodar, det vil seie at det er desse nodane – det funksjonelle domenet til f-strukturen – som spesifiserer informasjonen som står i f-strukturen. Viss me har grunnlag for å lenkje to f-strukturar, vil me au ha grunnlag for å lenkje nodane som projiserte desse f-strukturane. Og omvendt vil det aldri vere grunnlag for å ha ei c-strukturlenkje som står i konflikt med f-strukturlenkjer, dvs. kor ϕ av kjeldenoden er lenkja til noko anna enn ϕ av målnoden (då burde kjeldenoden vore lenkja til dette andre). Det at to nodar er lenkja på c-strukturnivå må i det minste implisere at informasjonen dei projiserer korresponderer. I utgangspunktet bør krevje følgjande:

- (15) to c-strukturnodar n_s og n_t kan berre lenkjast om $\phi(n_s)$ og $\phi(n_t)$ er lenkja på f-strukturnivå

Det enklaste ville vere å berre seie at alle nodane i dei to funksjonelle domena er mange-mange-lenkja med kvarandre, men denne lenkja vil ikkje gi oss meir informasjon enn at sjølv f-strukturane er lenkja; ei lenkje på c-strukturnivå bør kunne gi meir nyansert informasjon.

Det viktige forholdet på c-strukturnivå er *dominans*; hovedgrunnen til at me snakkar om c-struktur er at me vil skildre den hierarkiske inndeling av frasestrukturen i setninga, der ein node på høgare nivå *dominerer* mengder av nodar på lågare nivå. Ei lenkje mellom to c-strukturnodar må altså implisere at det dominerte materialet korresponderer.



Figur 3.3: Enkel lenkjing av c-strukturnodar mellom norsk og nederlandsk; IP til IP, I' til I' og I til I.

TODO: lenkjene!

I figur 3.3 er dei funksjonelle domena til *regnar/regent* lenkja¹⁹, og det same med *det/het* og *kjøttbollar/gehaktballen*. Viss me føreset at subjekt-NP-ane er lenkja med kvarandre, og at objekt-NP-ane er lenkja med kvarandre, på c-strukturnivå, vil det vere ønskeleg å ein-ein-lenkje IP-nodane, I'-nodane og I-nodane. Me skal sjå kvifor.

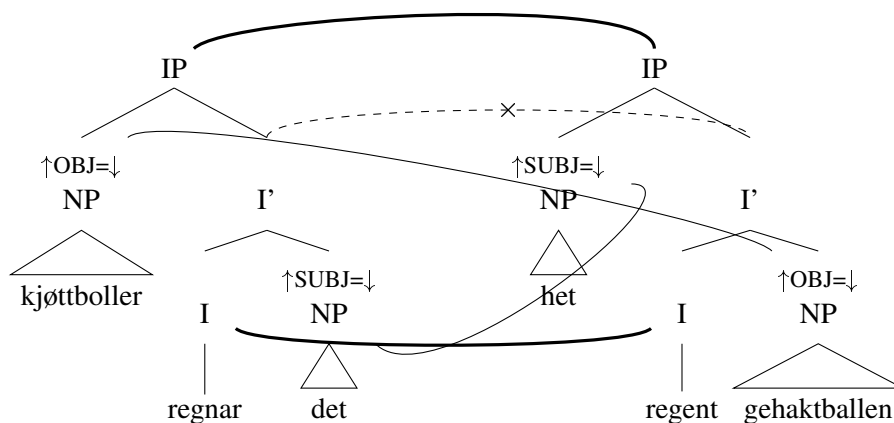
IP-nodane bør lenkjast sidan dei dominerer alt innanfor dei lenkja funksjonelle domena; det finst ikkje ein gong nodar som står utanfor det dei dominerer. Dei nodane som står nedanfor det funksjonelle domenet til IP-ane er i tillegg lenkja med kvarandre. Det vil seie at det ikkje finst informasjon på kjeldespråket som ikkje er uttrykt på målspråket (eller omvendt) innanfor det IP-ane dominerer.

I'-nodane dominerer ikkje subjektet i figur 3.3. Ei lenkjing av I'-nodane impliserer at det som står under desse korresponderer, men au at nodane står i liknande omgivnader. Det er lett å sjå føre seg eit døme der det ikkje ville vore ønskeleg med ei lenkje mellom I'-nodane. I figur 3.4 vil me t.d. ikkje lenkje desse nodane, på norsk dominerer I' subjektet, som er lenkja til subjektet på nederlandsk, men på nederlandsk står ikkje subjektet under I', og omvendt for objektet. Ei lenkje mellom I'-nodane ville sagt at nodane dei dominerte projiserte korresponderande informasjon, det gjer dei ikkje i figur 3.4. (I 3.3, derimot, står dei lenkja objekta under I', medan dei lenkja subjektet er utanfor.) Men merk at IP-nodane likevel

¹⁹I desse trea har eg annotert f-strukturforhold på visse nodar; der eg ikkje har teikna inn dette gjeld $\uparrow=\downarrow$, altså at noden er i same funksjonelle domene som mornoden. Eg har i tillegg forenkla kategorinamna ein del frå dei som kjem direkte frå ParGram/Xpar-analysane, det bør ikkje ha noko å seie for framstillinga her.

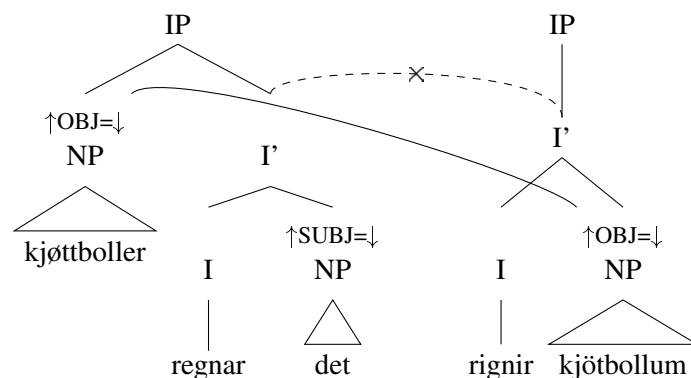
kan lenkjast, dei dominerer begge både subjekt og objekt, sjølv om dei kjem i ulik følge under. I-nodane dominerer berre verba, og kan au lenkjast.

nederlandsk
kunne hatt
inversion au!



Figur 3.4: C-strukturlenkjer kan ikkje gå på tvers av dominerte lenkjer (nynorsk og nederlandsk)

Sjølv om subjektet sto ulenkja, t.d. ved lenkjing inn i eit pro-drop-språk eller liknande, ville me fått same situasjon; I'-nodane i figur 3.5 kan ikkje lenkjast sidan I' på islandsk dominerer objektet, medan I' på norsk ikkje gjer dette, og objekta er lenkja med kvarandre (her både på c- og f-strukturnivå). Ei lenkje mellom desse I'-nodane ville sagt at dei dominerer korresponderande materiale, men det gjer dei ikkje.



Figur 3.5: C-strukturlenkjer kan ikkje gå på tvers av dominerte lenkjer (nynorsk og islandsk)

Når treet deler seg i to som i desse figurane, får me ei mogleg oppdeling av kjeldene til f-strukturinformasjonen. Me vil ikkje lenkje nodar som ikkje gir same tilskot til f-strukturen, på same måte som me ikkje vil lenkje på tvers av f-strukturlenkjer.

I både figur 3.4 og figur 3.5 er det slik at det I'-nodane dominerer gir ulike tilskot til f-strukturen, dei kan difor ikkje lenkjast. Likevel må me tillate litt slingeringsmonn her, nodane skal ikkje trenge projisere heilt like f-strukturar. Det som er relevant er det som blir lenkja i f-strukturen.

Som desse døma viser må me nyansere prinsippet om å ikkje lenkje c-strukturknodar på tvers av f-strukturlenkjer, til å ta innover seg dominans: me vil ikkje lenkje c-strukturknodar viss *det dei dominerer* kjem i konflikt med f-strukturlenkjer.

I visse tilfelle kan det hende at sjølv toppnodane i det funksjonelle domenet ikkje bør lenkjast. I døma over dominerer toppnoden i det funksjonelle domenet, IP, alt som står under $\phi(IP)$ i f-strukturen. I figur 3.6, derimot, er objektet til *regna* ikkje dominert av toppnoden i det funksjonelle domenet til *regna*, VP-en; men det er lenkja til objektet i funksjonelle domenet til *rained*. F-strukturane til dei to VP-ane er lenkja, men toppnodane i dei funksjonelle domena kan ikkje lenkjast sidan dei to toppnodane dominerer materiale som inneheld ulike lenkjer på f-strukturnivå – ei slik c-strukturlenkje ville stått i konflikt med f-strukturlenkjene. Intuitivt synest det au feil med ei lenkje mellom konstituentane *det regner* og *it rained meatballs*. Dei kan iallfall ikkje reknast som omsetjingar av kvarandre åleine; i ein større kontekst kan dei inngå i ein korrespondanse, men denne større konteksten har me jo lenkja allereie ved IP-nodane.

I det minste bør me difor krevje følgjande av lenkjer på c-strukturnivå:

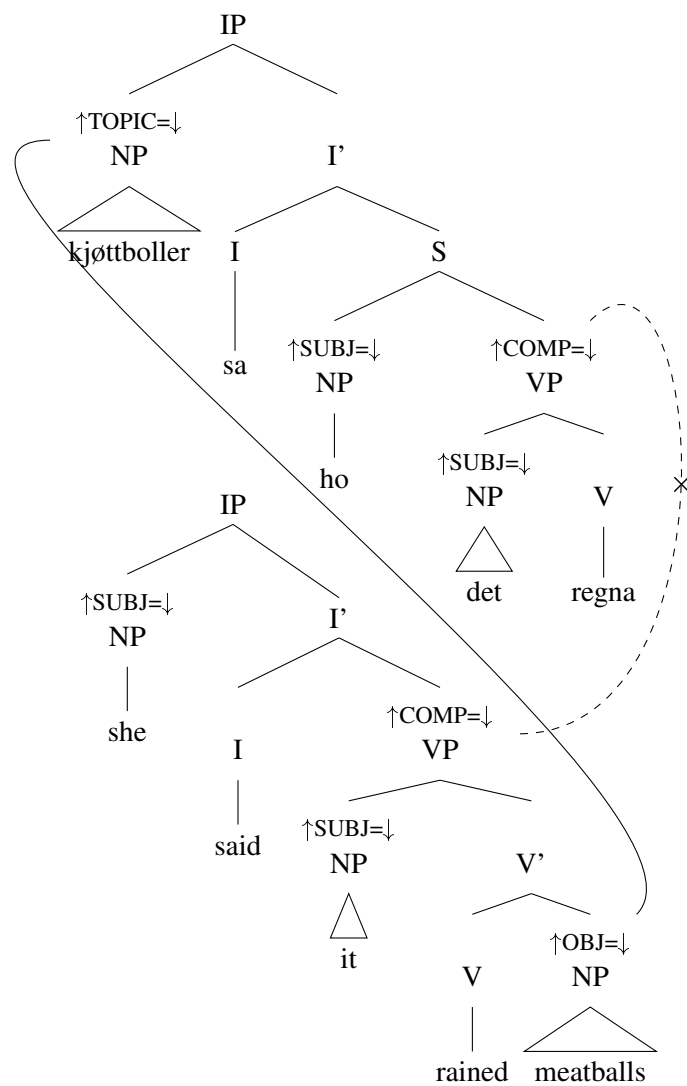
- (16) Ein node n_s kan lenkjast med ein node n_t berre viss:
- $\phi(n_s)$ er lenkja på f-strukturnivå med $\phi(n_t)$, og
 - det ikkje finst nodar under n_s som er lenkja med nodar utanfor det funksjonelle domenet til n_t , og
 - det ikkje finst nodar under n_t som er lenkja med nodar utanfor det funksjonelle domenet til n_t .

Men, kva om det finst nodar under n_s som ikkje er lenkja på c-strukturnivå (kanskje fordi det ikkje finst tilsvarende nodar på målspråket, t.d. ved lenkjing inn i pro-drop-språk), men som har ei lenkje på f-strukturnivå? Her finst det fleire alternative løysingar, som eg ser på nedanfor.

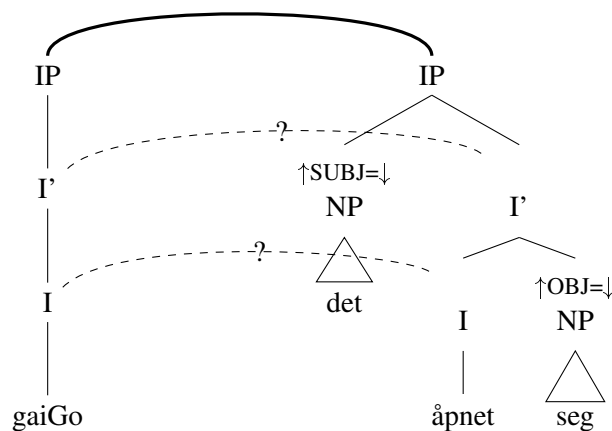
3.7.1 Lenkja f-strukturar utan c-strukturknodar

I figur 3.7 kan iallfall IP-nodane lenkjast, dei dominerer alle orda på begge setningane, og f-strukturane er lenkja. Men NP-subjektet på den norske sida, er ikkje lenkja med noko i det georgiske treet; dette subjektet er lenkja med eit pro-element på f-strukturnivå. Den informasjonen (her reint syntaktisk) som ordet *det* tilfører IP, ligg under I' på georgisk. Ved I-nodane manglar det norske treet i tillegg den informasjonen som *seg* tilfører.

Hadde det georgiske treet hatt spesifikator og komplement som kunne lenkjast til spesifikator og komplement på norsk, ville det ha vore uproblematisk å lenkje I' og I. Men om me berre har krav (16) å halde oss til, er det uspesifisert kva me



Figur 3.6: Sjølv toppnodane i eit funksjonelt domene kan stå ulenkja; her kan ikkje VP-nodane lenkjast sidan det norske TOPIC er objektet til *regna*, lenkja til objektet under VP på engelsk



Figur 3.7: Skal ulenkja søsternodar hindre lenkjing? (Georgisk og bokmål)

skal gjere i ein situasjon kor nodar lenkja på f-strukturnivå ikkje er lenkja på c-strukturnivå.

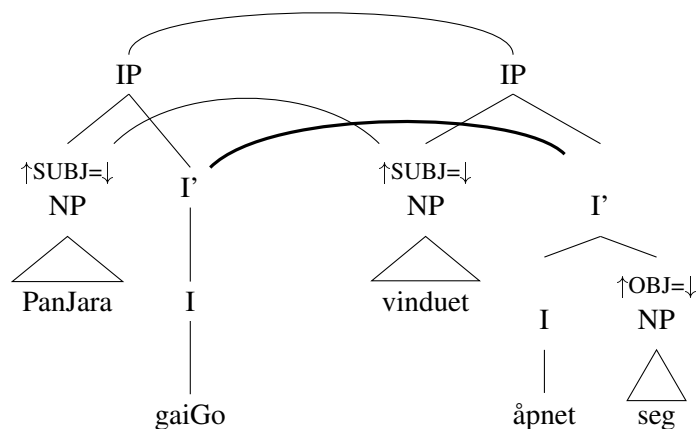
Det finst (iallfall) to alternativ.

Det eine alternativet er å seie seie at I- og I'-nodane ikkje skal lenkjast, sidan *det* og *seg* er lenkja på f-strukturnivå (til subjekt og objekt av *gaiGo*), då tolker me det slik at I' og IP dominerer ulikt lenkja materiale. Det at det *ikkje* finst ei lenkje mellom I'-nodane, men mellom IP-nodane, vil då opplyse oss om at I'-nodane dominerer ulike f-strukturlenkja informasjonstilskot på dei ulike språka; likeins for I-nodane. Eg kjem tilbake til korleis ein kan formalisere dette kravet i del 3.7.2.

Det andre alternativet er å ikkje gjere forskjell på IP, I' og I når det gjeld c-strukturlenkjinga. Grunnen til å gjere dette er at *gaiGo* både korresponderer med heile frasen *det åpnet seg*, men au med berre *åpnet seg*. I figur 3.8 ser me t.d. at I'-nodane kan lenkjast (utan å sjå på anna enn krav (16)), det vil altså vere mogleg å lenkje I'-nodane i andre omgivnader. Det finst ein slags dobbeltheit mellom korrespondansen *gaiGo-det åpnet seg* og korrespondansen *gaiGo-åpnet seg* og me kan uttrykkje dette ved å ikkje gjere forskjell på IP og I' i figur 3.7 (Dyvik 2010 (?)). *har eg forstått dette rett? (korleis er dette ein korrespondanse på tokennivå?)*

Dyvik et al. (2009, s. 77) definerer i denne samanhengen omgrepet *lenkja leksikalske nodar*, *LL*, kor *LL(n)* er mengda av nodar dominert av *n* som har ei ordlenkje. For å lenkje c-strukturnodane *n_s* og *n_t*, som er i lenkja funksjonelle domene, må alle nodane i mengda *LL(n_s)* vere lenkja til nodar i *LL(n_t)*. Ulenkja nodar under *n_s* og *n_t* står ikkje i vegen for lenkjing av *n_s* og *n_t*, men dei to mengdene kan ikkje vere tomme.

Dette kravet gjer at ein ikkje treng krav (16), og vil gi ei mange-mange-lenkje mellom alle nodane i dei to funksjonelle domena til *gaiGo-åpnet* i figur 3.7. Viss me skriv ei f-strukturlenkje som eit ordna par mellom PRED-verdien på kjeldesida (georgisk, med subskript *s*) og PRED-verdien på målsida (norsk, med subskript *t*) får me $LL(IP_s) = LL(I'_s) = LL(I_s) = \{(\mathbf{ga-Geba}, \mathbf{åpne})\} = LL(IP_t) = LL(I'_t) =$



Figur 3.8: Delvis mogleg lenkjing av underordna c-strukturnodar mellom georgisk og bokmål

$LL(I_t)$ kor *det* og *seg* er ulenkja på både c-strukturnivå og ordnivå²⁰.

3.7.2 Eit strengare lenkjingskriterium

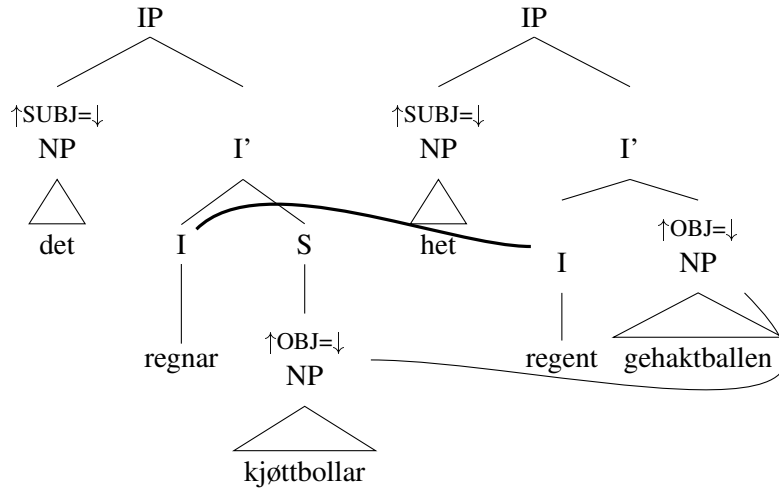
Sidan det er mogleg å ønskje seg å ikkje lenkje I' - og I-nodane i 3.7, gir eg her ein måte å formalisere dette på.

For å tillate lenkjene i figur 3.3, men ikkje dei stipla lenkjene i figur 3.7, ville det vore nok å krevje at søsternodane var lenkja. I figur 3.3 kan I-nodane lenkjast fordi objekta er lenkja, I'-nodane fordi subjekta er lenkja. I figur 3.7 kan dei norske I' - og I-nodane ikkje lenkjast med noko fordi søstrene deira ikkje er lenkja. Men dette blir for strengt. Det kan t.d. vere gode uavhengige grunnar til å ha ein mellomliggende S-node før objektet på norsk, kor S er i same funksjonelle domene som IP, medan det kanskje finst uavhengige grunnar for å *ikkje* gjere dette på andre språk. Figur 3.9 demonstrerer denne situasjonen. Her kan ikkje S lenkjast til objektet sidan dei ikkje er i same funksjonelle domene, men me vil jo likevel lenkje I-nodane; så eit krav om lenkja søsternodar blir for strengt.

Me treng altså eit litt meir nyansert krav. Som nemnt i fotnote 20 går det an å få til dette ved ein kombinasjon av konseptet om lenkja leksikalske nodar og å krevje at orda *det*, *åpnet* og *seg* i figur 3.7 er mange-mange-lenkja på ordnivå til *gaiGo*, sidan dei er lenkja til subjekt, predikat og objekt av *gaiGo* på f-strukturnivå.

²⁰ Merk at orda *det* og *seg* måtte definerast som ulenkja for at denne definisjonen skulle fungere, noko som krev at ein nyanserer krav (1-b) litt. Viss me hadde definert *det* og *seg* som mange-ein-lenkja (med *åpnet*) inn i *gaiGo*, ville me fått same resultat som det krav (17) i del 3.7.2 gir. Viss georgisk er kjeldespråket (n_s , norsk: n_t) blir $LL(IP_s) = LL(I'_s) = LL(I_s) = \{(gaiGo, det), (gaiGo, åpnet), (gaiGo, seg)\} = LL(IP_t)$. Mengdene $LL(I'_t) = \{(gaiGo, åpnet), (gaiGo, det)\}$ og $LL(I_t) = \{(gaiGo, åpnet)\}$ på den norske sida har då ikkje korresponderande mengder på georgisk og blir ikkje lenkja.

TODO:
diskutere litt
meir
forskjellane
på desse
alternativa



Figur 3.9: I-nodane bør lenkjast sjølv om søsternodane ikkje er lenkja (norsk og nederlandsk)

Men viss me vil unngå å referere til ordlenkjer, går det au an å definere kravet i form av f-strukturlenkjer på preterminale nodar²¹:

- (17) For å lenkje c-strukturnodane n_s og n_t :
 La $l_c(f)$ vere mengda som inneheld f-strukturlenkja til f , og f-strukturlenkjene til alle argument a av f som ikkje har c-strukturnodar, dvs. kor $\phi^{-1}(a) = \emptyset$.
 La $L_c(n)$ vere mengda av $l_c(\phi(n'))$ for alle f-strukturlenkja preterminale n' som er dominert av n . n_s og n_t kan lenkjast om $L_c(n_s) = L_c(n_t)$.

I figur 3.8 har me då følgjande situasjon:

$$\begin{aligned} L_c(IP_s) &= \{(\mathbf{PanJara}, \mathbf{vindu}), (\mathbf{ga-Geba}, \mathbf{\acute{a}pne}), (\mathbf{pro}, \mathbf{seg})\} = L_c(IP_t) \\ L_c(I'_s) &= \{(\mathbf{ga-Geba}, \mathbf{\acute{a}pne}), (\mathbf{pro}, \mathbf{seg})\} = L_c(I'_t) \\ L_c(I_s) &= \{(\mathbf{ga-Geba}, \mathbf{\acute{a}pne}), (\mathbf{pro}, \mathbf{seg})\} \neq \{(\mathbf{ga-Geba}, \mathbf{\acute{a}pne})\} = L_c(I_t) \end{aligned}$$

Dette vil seie at krav (17) gir lenkjer mellom IP-nodane og I'-nodane, men ikkje mellom I-nodane. I figur 3.7 vil ikkje ein gong I'-nodane få ei lenkje, sidan den norske I'-node dominerer $\{(\mathbf{ga-Geba}, \mathbf{\acute{a}pne}), (\mathbf{pro}, \mathbf{seg})\}$ medan den georgiske I'-node dominerer $\{(\mathbf{pro}, \mathbf{det}), (\mathbf{ga-Geba}, \mathbf{\acute{a}pne}), (\mathbf{pro}, \mathbf{seg})\}$, det same som IP-nodane.

Merk at om me omdefinierer $l_c(f)$ til å ikkje innehalde f-strukturlenkjer til argument av a , vil krav (17) gi same c-strukturlenkjer som kravet frå Dyvik et al. (2009), men definert i form av f-strukturlenkjer på preterminale nodar.

²¹Då kan me au representere mange-til-mange-ordlenkjer som «udelelege», $(\{\mathbf{gaiGo}\}, \{\mathbf{det}, \mathbf{\acute{a}pnet}, \mathbf{seg}\})$ blir den einaste ordlenkja i dømet over, sidan me ikkje må samanlikne ordlenkjene frå IP_t , I'_t og I_t .

3.7.3 Funksjonelle c-strukturnodar

Ikkje alle ord tilsvare PRED-element i f-strukturen, dette gjeld typisk funksjonsord (t.d. *som*, *at*). ...og desse vil me lenkje kvifor? i kva situasjonar? TODO diskuter. Ved endosentrisitetsprinsippa til Bresnan (2001) er komplementet til funksjonelle kategoriar (C, I, P) ein funksjonell ko-kjerne, det er altså komplementet som gir PRED-elementet i dette funksjonelle domenet.

Problemet med å nytte krava nemnt over i dette tilfellet er at nodar over funksjonsord er i det same funksjonelle domenet som komplementet, og nodane over funksjonsorda tilføyer ikkje ei ny PRED-lenkje som kan dele opp treet slik me gjorde tidlegare. Så me må utvide prinsippa for å dele opp c-strukturreet i buntar som dominerer same mengd med lenkjer.

Ord som ikkje projiserer PRED-lenkjer kan likevel ha LPT-korrespondanse og bestå krava på ordnivå, men når me skal lenkje desse på c-strukturnivå må me sjekke ordkrava direkte (me kan ikkje gå via nokon f-strukturlenkjing). LPT-kravet gir oss eit utgangspunkt for lenkjing.

Viss begge språk har funksjonsord, men funksjonsord som ikkje kan sjåast på som moglege omsetjingar (t.d. *fordi* og *whether*), bør me nok ikkje ein gong lenkje komplementa, sidan funksjonsorda då gjer at komplementa spelar ulike roller i omgivnadene²². Samtidig vil me ikkje at eit manglande funksjonsord på det eine språket skal hindre lenkjing av komplementa, sidan det kan hende at funksjonsordet ikkje er krevd på det språket (eventuelt kjem dette fram som korrespondansar i f-strukturtrekk, eg har ikkje teke høgd for korrespondansar mellom andre f-strukturelement enn PRED i denne oppgåva).

Me kan krevje at komplementa er lenkja for å sikre at me ikkje lenkjer nodar som står i ulike kontekstar (me vil ikkje lenkje *at* i «han såg at det gjekk bra» med *that* i «he saw that she drew a picture»), jamfør kravet om lenkja argument for lenkja predikat i del 3.6.1.

Desse ønskene kan me formalisere slik:

(18) Krav for lenkjing av funksjonelle kategoriar i c-strukturen:

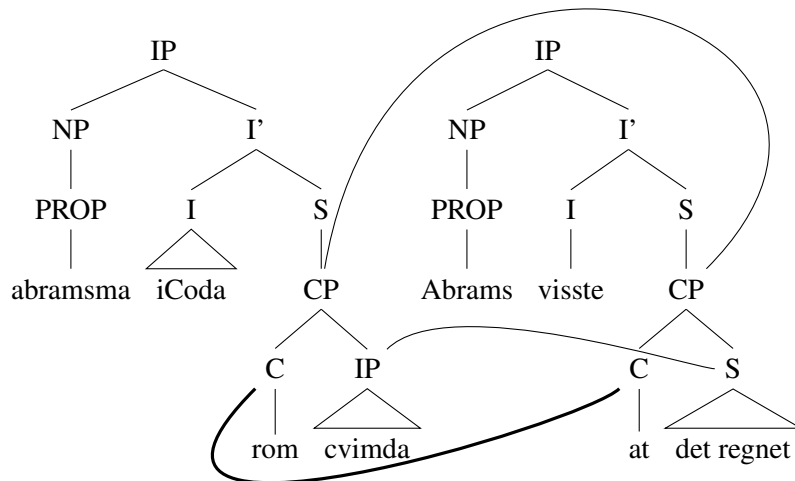
- a. Gitt ei mogleg lenkjing av FP og GP, kor F og G er funksjonelle kategoriar der komplementa elles kan lenkjast, tolk LPT-korrespondansen mellom orda under F' og G' som eit medlem av lenkjemengda L_c (evt. LL), kor denne må vere lik for at FP og GP skal kunne lenkjast, då kan me au lenkje F' og G'.
- b. Gitt ei mogleg lenkjing av FP og XP, der F er ein funksjonell kategori, medan X er ein ikkje-funksjonell kategori, ignorerer me den funksjonelle kategorien i c-strukturlenkjinga. Sidan det ikkje er nokon forskjell i L_c (evt. LL) mellom FP og F', er F' medlem av nodemengden

²²Skal ein lenkje ordet *som* (utan PRED) med ordet *which* (med PRED)? Viss krava elles er oppfylt, kan det kanskje vere informativt med ein type «defekt» lenkje, sjølv om berre det eine ordet blir rekna for å vere eit innhaldsord. Frasane til deira funksjonelle domene vil uansett kunne lenkjast viss dei andre krava er oppfylte.

som blir lenkja til XP.

Om (18-a) er oppfylt, kan me få samanstillinga vist i figur 3.10. Her vil dei funksjonelle domena til CP og CP kvar kunne delast opp i to delar, kor den funksjonelle delen har LPT-korrespondanse medan komplementa er lenkja på f-strukturnivå. Lenkjemengdene under CP-nodane er like, og dei under C-nodane er like.

(Alle nodane under S vist i dei to trea er i same funksjonelle domene, så om dei funksjonelle domena er lenkja, vil krav (15) vere oppfylt kva gjeld CP-komplementa – lenkjinga går ikkje ut over dei funksjonelle domena, medan krav (17) er dekkja for S-nodane med unntaket over.)



Figur 3.10: Mogleg samanstilling av funksjonelle c-strukturnodar mellom georgisk og norsk (bokmål)

Der det eine språket har eit funksjonsord og det andre språket ikkje krever det, bryr me oss ikkje om funksjonsordet. For å sjekke noko slikt må me som nemnt sjå på andre trekk enn PRED i f-strukturane, noko som blir utanfor denne oppgåva; men om me hadde sjekka slike f-strukturkorrespondansar kunne me unngått kravet om LPT-korrespondanse og i staden nytta informasjon frå f-strukturane til lenkjing av funksjonelle kategoriar. Utan å ha slike mekanismar på plass blir f-strukturtenkjinga avhengig av c-strukturforhold, og i implementasjonen min har eg difor lagt mindre vekt lenkjing av funksjonelle kategoriar.

3.8 Rangering

Gitt ei viss f-struktursamanstilling, vil det berre vere éin mogleg måte å lenkje på c-strukturnivå. Men slik f-strukturkrava er stilt, kan me få mange ulike moglege samanstillingar på f-strukturnivå. Ideelt sett burde krava vere nyanserte nok til å plukke ut berre dei samanstillingane som er ønskelege, men som vist over er dette ikkje alltid like lett, spesielt om me ikkje har fullstendig informasjon om LPT-korrespondansar.

Difor er det nyttig å ha nokre kriterium, eller i det minste heuristikkar, for å rangere ulike f-struktursamanstillingar. Det er sjølvsagt svært mange måtar ein kan rangere to strukturar på; kriteria nedanfor er tenkt å gi dei samanstillingane som føreset at argumentstrukturane er så like som mogleg. Implementasjonen av kriteria kjem i del 4.2.

Merk at om me har «fullstendig» LPT-informasjon (t.d. ei perfekt omsetjings-ordbok), treng me aldri rangere. Desse kriteria er difor kanskje mindre teoretisk interessante, sidan dei alltid kan overstyrast ved å gi systemet meir kunnskap. Men det kan au vere interessant å teste kor godt ulike rangeringskriterium fungerer der me ikkje har LPT-informasjon i det heile – då vil dei i prinsippet fungere som «mjuke» skrankar på f-struktursamanstillinga.

Nedanfor følgjer dei kriteria eg har basert implementasjonen på. Som nemnt er det berre ein av mange moglege måtar å gjere det på, men kriteria tek innover seg dei trekka som er relevante for argumentstrukturen: argumentfølgje og rekursiv f-strukturlenkjing.

3.8.1 Rangering ved følgje

I Dyvik et al. (2009, s. 75–76) blir det formulert eit spesialtilfelle av krava for lenkjing på f-strukturnivå, der det er like mange argument på kvart predikat, og førsteargument er lenkja til førsteargument, andre til andre, osv. I slike situasjonar er ingen adjunkt lenkja til argument; og om argumentstrukturane i analysane reflekterer det semantiske rollehierarkiet, vil me aldri lenkje t.d. agens til patiens og patiens til agens. Der grammatikkane er skrivne etter parallelle prinsipp, bør ei slik samanstilling – om alt anna er likt – gi den ønskelege lenkjinga.

Difor har me dette som eit rangeringskriterium. For å nyansere det litt, kan me sjå på kor mange av lenkjene frå argumenta til eit predikat som ikkje er argument-adjunkt-lenkjer, og der dei ikkje har ulik posisjon i argumentstrukturen. Meir formelt:

- (19) La m vere mengda av ein-til-ein LPT-korresponderande argument/adjunkt av to lenkja f-strukturar F_s og F_t . La n vere dei elementa av m som anten begge er adjunkt, eller begge er argument og har same posisjon i argumentstrukturane sine. *Følgjeskåren* til F_s, F_t er då $\frac{n}{m}$.

I spesialtilfellet nemnt over, vil skåren altså vere 1. Når det står «ein-til-ein LPT-korresponderande» over, er det for å plukke ut ein viss måte å samanstillе ar-

gumenta og adjunkta til F_s og F_t – men utan krav om at desse skal vere rekursivt lenkja. Kriteriet nedanfor rangerer rekursive lenkjer høgare enn enkle LPT-korrespondansar.

3.8.2 Rangering ved djupn

Krav (8) krev ein-til-ein LPT-korrespondanse mellom argument/adjunkt av kjelde- og målpredikatet, men stiller ikkje krav om at dei LPT-korresponderande elementa sjølv må vere lenkja på f-strukturnivå. Så viss «She tried to ride a bike» er lenkja med «Ho prøvde å sykle», kan me lenkje *try* med *prøve* sjølv om *ride* og *sykle* ikkje kan lenkjast – *bike* er eit påkravd argument²³. I situasjonar der me har eit val mellom berre LPT-korrespondanse, og full rekursiv lenkjing, vil ei rekursiv lenkje skildre meir strukturell likskap enn berre LPT-korrespondansen.

Difor har me dette som eit rangeringskriterium. Me kan formalisere det slik:

- (20) La m vere mengda av ein-til-ein LPT-korresponderande argument/adjunkt av to lenkja f-strukturar F_s og F_t . La n vere dei elementa av m som anten ikkje har argument sjølv, eller som er lenkja på f-strukturnivå. *Djupnskåren* til F_s, F_t er då $\frac{n}{m}$.

Der alle LPT-korresponderande argument/adjunkt av F_s og F_t au er lenkja, vil djupnskåren vere 1.

3.8.3 Rangering for heile samanstillinga

Kriteria over gir to skårer for eit visst par av PRED-element. Utan å ha testa desse kriteria empirisk er det naturleg å vekte dei likt; altså bør skåren for eitt par av PRED-element innehalde produktet av desse skårane.

Men argumenta og adjunkta av desse to elementa kan sjølv ha ulike moglege delsamanningar, som kan gi ulike skårer. La a_s og a_t , argument eller adjunkt av F_s og F_t , vere lenkja i ei mogleg samanstilling av F_s og F_t . Innanfor a_s, a_t finn me kanskje ulike moglege samanstillingar, her vel me den som gir best skåre. Men det kan hende at det finst ei alternativ måte å lenkje på, kor F_s er lenkja til G_t , a_s til d_t (argument/adjunkt av G_t), og a_s, b_t har høgare skåre enn a_s, a_t . Viss alt anna er likt, bør me då heller velje F_s, G_t enn F_s, F_t .

Den endelege skåren for F_s, F_t er då produktet av følgjeskåren, djupnskåren og den vekta summen av dei endelege skårane for lenkja argument/adjunkt av F_s og F_t . Denne summen er vekta på lengden av lista med lenkja argument/adjunkt, for å ikkje gi unaturleg høg skåre til f-strukturar med mange argument/adjunkt; men viss me til slutt har mange greiner med same skåre, vel me den lengste sidan den då har flest lenkja adjunkt²⁴.

²³Merk: utan meir informasjon her vil krava i tillegg opne for ei mange-mange-lenkje mellom *ride* + *bike* og *sykle*; for skuld argumentet ser me berre på ein-til-ein-lenkjer her.

²⁴Dette er ganske *ad hoc* skåringsfunksjon, og det er ikkje sikkert at alle kriteria bør ha lik vektning. Sidan det ikkje er meininga å nytte denne skåringa i maskinlæring har eg ikkje undersøkt om dette

3.9 Oppsummering

Krava formulert i denne delen definerer kva for lenkjer som er ønskelege mellom konstituentar og mellom f-strukturar, når formålet er å annotere ein parallell tre-bank for lingvistiske studium. Det finst visse tilfelle der krava *ikkje* kan avgjere kva som er den ideelle lenkja utan meir informasjon enn det dei grammatiske analysane gir; eg diskuterer slike døme i kapittel 5, i tillegg til døme som er problematiske for krava sjølv der me har fullstendig informasjon, og der meir arbeid trengst. Kapittelet som kjem no gir detaljane for programmet `lfgalign`, som implementerer krava frå dette kapittelet sånn at det er lettare å teste kor dei slår feil.

kan gjerast til ein stringent sannsynsmodell (der summen av skårar for alle moglege samanstillingar aldri går over 1); men det bør vere mogleg å gjere noko slikt ut av det.

Kapittel 4

Implementasjonen av `lfgalign`

Eit formelt krav kan se bra ut på papiret, men ha skjulte manglar som ikkje kjem fram før ein har testa det. Ei implementering gjer det med ein gong synleg om det finst manglar i det formelle kravet, eller om noko ikkje er presist nok spesifisert.

For å finne ut av kor godt krava i forrige kapittel fungerer til å avgrense kva for lenkjer som er moglege, har eg implementert dei etter beste evne i eit Common Lisp¹-program. Dette kapittelet gir eit overblikk over implementasjonen, medan neste kapittel går gjennom resultat av køyring².

Programmet `lfgalign` tek inn LFG-analysane av to setningar som me av uavhengige grunnar trur er omsetjingar av kvarandre. LFG-analysane må vere disambiguerte og i Prolog-formatet frå XLE³. Programmet les inn dei to filene og opprettar ein intern representasjon av LFG-analysen.

Me kan i tillegg gi programmet informasjon om kva for ord-omsetjingar me ser på som lingvistisk prediktable. Intensjonen er at dette kan vere informert av omsetjingstabellen frå eit automatisk ordsamanstillingsprogram, eller av handskrivne omsetjingsordbøker.

Programmet byrjar lenkjinga med f-strukturane. Ei f-struktursamanstilling er ei mengd med *lenkjer* mellom individuelle f-strukturar. Resultatet av lenkjinga på dette nivået kan vere tvitydig: sidan det ofte finst fleire måtar å lenkje argument og adjunkt på, får me i første omgang mange samanstillingar mellom kjelde- og

¹Dette språkvalet kan gjere eventuell integrering med andre LFG-system lettare (Common Lisp er m.a. nytta i LFG Parsebanker (Rosén et al., 2009)).

²«An article about [...] science in a scientific publication is **not** the scholarship itself, it is merely **advertising** of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.» (Jon Claerbout, i Stodden, 2009, s. 7–8). Kjeldekoden til implementasjonen er tilgjengeleg frå <http://github.com/unhammer/lfgalign> (som fri og open programvare under GNU General Public License versjon 3 eller seinare), saman med testmaterialet nytta i neste kapittel.

³Formatet er dokumentert på <http://www2.parc.com/isl/groups/nlt/xle/doc/xle.html>. Importeringa til Lisp-strukturar handterer «pakka representasjonar» og kjenner igjen ekvivalensforhold (t.d. der fleire ϕ -variablar refererer til same f-struktur, eller fleire Prolog-variablar refererer til same analyseval); men filene eg har testa utnyttar ikkje det fulle spennet til formatet, så det finst ganske sikkert feil.

mål-f-strukturar.

Difor rangerer me f-struktursamanstillingane, og den beste sender me vidare til c-struktursamanstillinga. Denne delen av programmet gir ut éi, utvitydig mengd med mange-til-mange-lenkjer mellom c-strukturane (her treng me ingen range-ring). Nodane i kvar av desse mange-til-mange-lenkjene definerer no den endelege frasesamanstillinga.

Nedanfor går eg gjennom detaljane rundt dei relevante delene av programmet.

4.1 Lenkjer mellom f-strukturar

Hovudalgoritmen for lenkjing mellom f-strukturar er vist i kodefigur 1. Funksjonen `f-align` returnerer ei mengd med moglege samanstillingar. Kvar samanstilling er ei mengd med par av f-strukturar⁴. Eit par (F_s, F_t) representerer ei lenkje frå ein f-struktur på kjeldespråket, til ein f-struktur på målspråket. Me går ut frå at dette paret har LPT-korrespondanse⁵, dette blir sjekka før alle kall på `f-align`. Der me ikkje har informasjon om LPT-korrespondanse mellom to ord (orda er ukjende), er lenkjing lov. Pro-element og substantiv kan alltid lenkjast med kvarandre.

Funksjonen `f-align` prøver først å lenkje F_s og F_t ein-til-ein. Viss ikkje dette går, prøver me å føye saman den eine av desse f-strukturane med eitt av argumenta til den andre.

Hjelpfunksjonen `argalign` (som igjen kallar `argalign-p`, vist i kodefigur 2) gir alle moglege «argumentpermutasjonar», dvs. moglege kombinasjonar av lenkjer mellom argumenta til F_s og F_t som tilfredsstiller kravet om LPT-korrespondanse, men utan å sjekke at desse argumenta igjen kan samanstillast. Funksjonen prøver å lenkje kvart argument til eit argument eller eit adjunkt, men gir ingen lenkjer mellom to adjunkt (sjå del 4.1.1 nedanfor om dette). Funksjonen gir heller ikkje kombinasjonar der minst eitt argument ikkje er lenkja – alle kombinasjonane må inkludere alle argument frå F_s og F_t , jf. krav (8) (ev. krav (14), for `margalign`, kalt der me har ei ein-mange/mange-ein-lenkje).

Funksjonen `argloop` i kodefigur 3 går gjennom éin av desse argumentpermutasjonane, og prøver å leggje til overflødige adjunkt, i tillegg til å kalle `sub-f` (kodefigur 4), som prøver å rekursivt lenkje kvart par av argument/adjunkt i permutasjonen.

Eit døme: viss F_s har argumenta SUBJ og OBJ og ingen adjunkt, og F_t har argumentet SUBJ og eitt adjunkt ADJ, der alle ord-omsetjingar er moglege, vil `argalign` gi dei to samanstillingane $\{(SUBJ, SUBJ), (OBJ, ADJ)\}$ og $\{(SUBJ, ADJ), (OBJ, SUBJ)\}$.

⁴Eigentleg eit slag avgjerdstre; kvart element er eit par, kor første element er lenkja mellom dei yttarste f-strukturane, og andre element er dei moglege samanstillingane for dei indre strukturane. Denne strukturen er nyttig for å rangere samanstillingar, og `f-align` blir mykje meir oversiktleg av å jobbe med eit slikt tre. Ein funksjon `flatten` kan nyttast til å omforme det ferdige treet til ei enkel liste med samanstillingar, kor kvar samanstilling er ei flat liste med lenkjer mellom f-strukturar.

⁵Når eg her skriv at to f-strukturar har LPT-korrespondanse, meiner eg sjølvsagt at ordformene til PRED-verdien til kvar f-struktur har LPT-korrespondanse.

```

alignments  $\leftarrow \emptyset$  ;
argperms  $\leftarrow$  argalign( $F_s, F_t$ ) ;
if argperms then
    add argloop(argperms,  $\emptyset$ ) to alignments ;
else
    forall the  $A_s$  in arguments( $F_s$ ) where LPT( $A_s, F_t$ ) do
        argperms  $\leftarrow$  margalign( $F_s, F_t, A_s, F_t$ ) ;
        add argloop( $F_s, F_t, (A_s, F_t)$ ) to alignments ;
    end
    forall the  $A_t$  in arguments( $F_t$ ) where LPT( $F_s, A_t$ ) do
        argperms  $\leftarrow$  margalign( $F_s, F_t, F_s, A_t$ ) ;
        add argloop( $F_s, F_t, (F_s, A_t)$ ) to alignments ;
    end
if alignments =  $\emptyset$  then return  $\emptyset$  ; // Fail
else return (( $F_s, F_t$ ), alignments) ;
Funksjon 1: f-align( $F_s, F_t$ )

```

Viss adjunktet til F_t ikkje fantest, eller ikkje hadde LPT-korrespondanse med nokon av argumenta til F_s , ville me ikkje fått nokon samanstillingar; medan viss paret (SUBJ, SUBJ) ikkje hadde LPT-korrespondanse og alt anna var likt, ville me berre fått den siste samanstillinga.

Funksjonen argloop går så gjennom kvar argumentpermutasjon og kallar sub-f på permutasjonen; sub-f prøver å kalle f-align på alle lenkjene. Sidan lenkjene som argalign gir har LPT-korrespondanse, vil alle f-strukturane i dei rekursive kalla i f-align ha LPT-korrespondanse. Eit rekursivt kall kan gi nye samanstillingar i dei indre f-strukturane, viss dei relevante krava er oppfylte.

Det er mogleg at ei lenkje frå éi samanstilling kan finnast i andre samanstillingar, sub-f unngår dobbeltarbeid ved å lagre alle delvise samanstillingar i alignable-tabellen⁶. Dette føreset at f-align(s, t) er uavhengig av konteksten rundt⁷; t.d. må mengda av samanstillingar som kjem ved å lenkje subjektet til F_s mot subjektet til F_t vere uavhengig av om objektet til F_s er lenkja mot eit objekt eller eit adjunkt osb. av F_t .

Sjølv om det er krav om LPT-korrespondanse mellom kvart argument og eit argument/adjunkt for å lenkje F_s og F_t , er det ikkje noko krav om at alle para i ein argumentpermutasjon tilfredsstiller alle lenkjingskrava. Viss f-align(OBJ, ADJ) frå dømet over gir null, og ikkje kan lenkjast (t.d. fordi ADJ hadde eitt argument, og OBJ ingen argument/adjunkt), medan f-align(SUBJ, SUBJ) kan lenkjast, vil f-align likevel returnere samanstillinga som inneheld (OBJ, ADJ) og (SUBJ, SUBJ). Me kan sjå i alignable for å finne ut av om kvar av f-strukturane kunne lenkjast; i

⁶Dette ser ut til å auke den totale farta med ca. 35 %.

⁷I programmeringsterminologi: at problemet har *optimal substruktur* – den optimale løysinga må vere mogleg å finne frå optimale delløysingar for at me skal kunne nytte slik dynamisk programmering.

usage: Kalt av argalign slik:

argalign-p(arguments(F_s), adjuncts(F_s), arguments(F_t), adjuncts(F_t))
ved samanføynd lenkje frå F_s og a_s til F_t slik:
argalign-p(arguments(a_s) \cup arguments(F_s) $- a_s$, adjuncts(a_s) \cup
adjuncts(F_s), arguments(F_t), adjuncts(F_t))

$a \leftarrow \emptyset$;

if $args_s$ **then**

| $s \in args_s$;

| **forall the** $t \in args_t$ **where** $LPT(s,t)$ **do**

| | **forall the** $p \in argalign-p(args_s - \{s\}, adj_s, args_t - \{t\}, adj_t)$ **do**
| | add $\{(s,t)\} \cup p$ to a ;

| **end**

| **forall the** $t \in adj_s$ **where** $LPT(s,t)$ **do**

| | **forall the** $p \in argalign-p(args_s - \{s\}, adj_s, args_t, adj_t - \{t\})$ **do**
| | add $\{(s,t)\} \cup p$ to a ;

| **end**

| **return** a ;

else if $args_t$ **then**

| **if** adj_s **then**

| | $s \in adj_s$;

| | **forall the** $t \in args_t$ **where** $LPT(s,t)$ **do**

| | | **forall the** $p \in argalign-p(args_s, adj_s - \{s\}, args_t - \{t\}, adj_t)$
| | | **do** add $\{(s,t)\} \cup p$ to a ;

| | **end**

| | **return** a ;

| **else**

| | **return** \emptyset ; // Fail

else

| **return** $\{\emptyset\}$; // End

Funksjon 2: argalign-p($args_s$, adj_s , $args_t$, adj_t)

```

alignments  $\leftarrow \emptyset$  ;
if argperms non-empty then
    forall the argperm in argperms do
         $p \leftarrow (M_s, M_t) \cup \text{sub-f}(\text{argperm})$  ; // optional merged arg
        add  $p$  to alignments ;
        forall the adjperm in adjalign(argperm,  $F_s$ ,  $F_t$ ) do
             $a \leftarrow p \cup \text{sub-f}(\text{adjperm})$  ; // optional adjunct links
            add  $a$  to alignments;
        end
    end
else
    // no arguments that need matching, add any adjuncts
    forall the adjperm in adjalign( $\emptyset$ ,  $F_s$ ,  $F_t$ ) do
        add sub-f(adjperm) to alignments ;
    end
return alignments

```

Funksjon 3: argloop(argperms, (M_s, M_t))

```

 $p \leftarrow \emptyset$  ;
forall the  $A_s, A_t$  in perm do
    if aligntable[ $A_s, A_t$ ] then
        alignment  $\leftarrow$  aligntable[ $A_s, A_t$ ] ;
    else
        alignment  $\leftarrow$  f-align( $A_s, A_t$ );
        aligntable[ $A_s, A_t$ ]  $\leftarrow$  alignment ;
    end
    if alignment then
        add alignment to  $p$ ;
    else
        add  $(A_s, A_t)$  to  $p$  ;
    end
return  $p$ 

```

Funksjon 4: sub-f(perm, aligntable)

dette tilfellet vil *aligntable*[OBJ, ADJ] vere tom.

Om me i tillegg krev at substrukturar kan samanstillast kan me utelukke lenkjing av f-strukturane F_s og F_t i (1) under:

- (1) a.
$$F_s \left[\begin{array}{l} \text{PRED 'planlegge'} \langle eg, [1:gi] \rangle \\ \text{XCOMP}_1 \left[\text{PRED 'gi (opp)'} \right] \end{array} \right]$$
- b.
$$F_t \left[\begin{array}{l} \text{PRED 'plan'} \langle I, [2:give] \rangle \\ \text{XCOMP}_2 \left[\text{PRED 'give'} \langle I, him, it \rangle \right] \end{array} \right]$$

Men som del 3.8 nemner kan det vere at me ikkje *vil* krevje dette i alle moglege tilfelle. Ei tryggare løysing er å rangere ulike løysingar i etterkant, ved å spørje etter dei argumentsamanstillingane som har flest lenkja substrukturar, dette kjem eg tilbake til i 4.2 nedanfor.

4.1.1 Overflødige adverbial

Argumentpermutasjonane frå *argalign* prøver som nemnt ikkje reine adjunkt-adjunkt-lenkjer, sidan me ikkje vil forkaste lenkjing av F_s og F_t berre på grunn av at ikkje alle adjunkt kunne lenkjast. Men når me har prøvd ein argumentpermutasjon, kan me lage ein kopi av denne som i tillegg inneheld lenkjer mellom «overflødige» adverbial, altså dei adjunkt-adjunkt-lenkjene som *argalign* ikkje prøver. Hjelpesfunksjonen *adjalign* (ikkje vist her) konstruerer moglege permutasjonar av lenkjer mellom adjunkt som ikkje er inkludert i *argperm*, og *argloop* prøver desse rekursivt via *sub-f* på same måte som med argumentlenkjene. Lenkjene blir lagt til ein *kopi* av argumentpermutasjonane, sidan det ikkje er sikkert at me ønskjer å lenkje alle adjunktdøtre. Viss me har to overflødige adjunkt på kvar side, og kravet om LPT-korrespondanse er dekkja for alle fire moglege par, får me seks moglege permutasjonar, sidan me inkluderer dei fire permutasjonane der eitt adjunktpar er ulenkja.

Viss F_s og F_t ikkje hadde argument i det heile teke, går *argloop* au gjennom moglege permutasjonar av adjunktdøtre, på same måte.

4.1.2 Når f-strukturlenkjene ikkje er ein-til-ein

Som nemnt i del 3.6.4 hoppar me over adposisjonar som plukkar ut adjunkt/argument, dette skjer t.d. i *adjalign* ved at funksjonen som henter ut f-strukturen til adjunktdøtre av ein f-struktur hoppar over adposisjonar.

Der det er umogleg å finne ein kombinasjon av argument- og adjunkt-lenkjer slik at alle argument har LPT-korrespondanse med ein unik f-struktur, dvs. der *argalign* feilar, prøver me ein-mange/mange-ein-lenkjer. Dette er ikkje verre enn at ein kallar *argalign-p* med unionen av argument frå dei to samanføyde f-strukturane (minus desse f-strukturane sjølve); men me sjekker i tillegg at kjeldeargumentet

som blir samanføyd med kjeldepredikatet har LPT-korrespondanse med målpredikatet. Sidan dette berre skjer viss argalign feilar, blir det naturleg nedprioritert, som forklart i del 3.6.5.

4.1.3 Kan me gjere f-struktursamanstillinga bottom-up?

Denne metoden går top-down frå ytre PRED og ned i underordna argument/adjunkt. Me vil difor aldri prøve å lenkje ein ytre f-struktur med ein indre f-struktur, utanom ved ein-mange-lenkjing.

Ein alternativ metode for lenkjing av f-strukturane er å byrje med alle logisk moglege permutasjonar av LPT-korrespondansar, og så sile ut dei som ikkje svarer til krava. Prosessen ville nok blitt mykje meir oversiktleg på denne måten, sidan det då berre er snakk om å sjekke krav for kvar enkelt lenkje. Men ein slik metode er vanskeleg i praksis; når avskjeringa skjer så seint, blir det alt for mange moglege kombinasjonar for lengre setningar med mange ukjende ord til at ein vanleg datamaskin kan halde styr på dei.

Me må i alle tilfelle vere klar for ei setning der alle ord er ukjende (me har ingen informasjon om LPT-korrespondanse), slik at kvart kjeldeord kan lenkjast til kvart målord. Viss begge setningane er 4 ord, får me 16 moglege samanstillingar der alle ord er med i nøyaktig éi lenkje (2^l , kor l er setningslengd). Men ofte har me null-lenkjer, me må altså i tillegg tillate samanstillingar der minst eitt ord er ulenkja, utan at me treng å vite kva for ord det er; med desse kortare listene inkludert får me endå fleire moglege samanstillingar per setning (4 ord gir 26, 8 ord gir 2186 moglege samanstillingar). Sjølv om me heile tida vel dei samanstillingane som lenkjar flest ord, vil maskinen raskt få problem. I tillegg har me problemet med 1-mange-lenkjer, som skaper endå fleire moglege samanstillingar.

Ein sideverknad av å byrje med ytre lenkjer og gå innover (prosessen skildra i del 4.1) er at me automatisk unngår å prøve «kryssande» lenkjer, t.d. å lenkje F_s med XCOMP av F_t , og XCOMP av F_s med F_t (denne kombinasjonen av lenkjer vil jo vere ein del av alle logisk moglege permutasjonar). Me får au prioritert å lenkje ytre element, som jo er sikrare lenkjer: gitt to f-strukturar for setningar der alt me veit om lenkjinga er at *setningane* er omsetjingar av kvarandre, vil dei to ytre f-strukturane ha størst sjanse for å korrespondere med kvarandre. For kvart steg du går innover må du multiplisere inn sjansen for å trå feil i argumentpermutasjonane.

Det finst altså både praktiske og meir ideelle grunnar til å gjere det på denne måten, men om det faktisk fungerer er eit spørsmål eg kjem tilbake til i del 5.

4.2 Rangering

Rangering foregår etter kriteria formalisert i del 3.8. Implementasjonen av kriteria følger formaliseringa ganske eksakt. Funksjonen *rank-f*, i kodefigur 5, tek ei urangert f-struktursamanstilling, representert som eit avgjerdstre kor førsteelement er ei lenkje (*link*) mellom to f-strukturar, og andreelement er ei mengd (*branches*) med

moglege måtar å samanstille argumenta og adjunkta til desse f-strukturane. Kvar enkelt grein i *branches* er ei mengd med nye avgjerdstre, eitt tre for kvart lenkja argument/adjunkt i den moglege delsamanninga. Så viss me har ei lenkje mellom to ytre pred F_s og F_t , og desse har to argument kvar $(a1_s, a2_s, a1_t, a2_t)$, vil *link* vere (F_t, F_s) medan *branches* er $\{\{a1a1tree, a2a2tree\}, \{a1a2tree, a2a1tree\}\}$; her har *a1a1tree* har som *link* $(a1_s, a2_t)$, osv. Viss desse argumenta ikkje har argument/adjunkt sjølve, vil deira *branches* vere tomme, og deira *rank-f* er då 1. For F_t og F_s vil funksjonen *rank-f* returnere den greina som har best skåre⁸, gitt ved *rank-branch*.

Kodefigur 6 illustrerer funksjonen *rank-branch*. Her har me fått ei viss lenkje *link* mellom PRED-element, og ser på alle dei moglege måtane å lenkje argument/adjunkt på – kvar måte er representert ved eit par *sublink*, *subbranches*. Me finn først den rekursive skåren for *subbranches* via *rank-f* igjen, og summerer dette inn i *subrate-sum*. Me returnerer produktet av skårene frå rangeringskriterium (19) og (20) med *subrate-sum* vekta på kor mange lenkjer det var i greina.

```

if branches =  $\emptyset$  then
  | return (link, 1)
else
  | best-rate  $\leftarrow$  0 ;
  | best-branches  $\leftarrow$   $\emptyset$  ;
  | forall the branch  $\in$  branches do
  |   | newbranch, rate  $\leftarrow$  rank-branch(seen, link, branch) ;
  |   | if rate  $\geq$  best-rate then
  |   |   | best-branch  $\leftarrow$  newbranch ;
  |   |   | best-rate  $\leftarrow$  rate ;
  |   |
  |   end
  |   add best-branch to seen ;
  | return seen, best-rate ;

```

Funksjon 5: *rank-f*(*seen*, *link*, *branches*)

I tillegg til skåren, returnerer desse funksjonane den beste greina⁹, slik at me ender opp med ei enkel, «flat» liste med lenkjer. Denne blir sendt vidare til c-strukturlenkjinga.

⁸Det kan godt hende me har fleire greiner med same skåre – kodefiguren viser ikkje dette, men me samlar opp alle med høgast skåre og returnerer den *lengste* av desse for å fylle opp med så mange adjunkt-lenkjer som mogleg. Viss det er fleire lengste, vel me berre den første, sidan det ikkje er meir å rangere på.

⁹Alternativt kunne me returnert eit tre som var annotert med skårene.

```

subs  $\leftarrow \emptyset$  ;
substrate-sum  $\leftarrow 0$  ;
forall the sublink, subbranches  $\in$  branch do
    newsub, substrate  $\leftarrow$  rank-f(seen, sublink, subbranches) ;
    add newsub to subs ;
    substrate-sum  $+$  = substrate ;
end
rate  $\leftarrow$  sub-f-rate(seen, branch)  $\cdot$  arg-order-rate(link, seen, branch)
     $\cdot \frac{\text{substrate-sum}}{\text{count}(\text{subs})}$  ;
return subs, rate ;
Funksjon 6: rank-branch(seen, link, branch)

```

4.3 Lenkjing av c-strukturnodar

Samanstilling mellom f-strukturar treng i `lfgalign` ikkje informasjon om c-strukturen, medan lenkjing av c-strukturnodar skjer på grunnlag av f-struktursamanstillinga. Programmet utfører difor samanstilling av c-strukturar sist¹⁰.

Funksjonen `c-align` har som inndata c-strukturanalysane av kjelde- og målsetninga, og éi f-struktursamanstilling; utdata er ei mengd med lenkjer. Ei lenkje er eit par der første element er ei mengd c-strukturnodar på kjeldespråket, og andre element ei mengd nodar på målspråket. Det er ingen overlapp mellom medlem av lenkjer (ein node er aldri med i meir enn eitt par).

I Dyvik et al. (2009, s. 77) er kravet for å lenkje to c-strukturnodar er at dei dominerer same mengd med ordlenkjer¹¹. Ein node n dominerer ei mengd lenkjer l viss unionen av lenkjene dominert av døtrene til n er lik l . I `lfgalign` opererer eg ikkje med *ordlenkjer* i seg sjølv; f-struktursamanstillinga er basert på LPT-korrespondansar, som definerer moglege ordlenkjer utan å sjå på kontekst, og f-struktursamanstillinga avgrensar vidare moglege ordlenkjer gitt f-strukturinformasjon. Preterminale nodar er dei mest ordnære nodane som kan ha ei f-strukturlenkje (ved \emptyset); når formålet er å lenkje c-strukturnodar kan me nytte f-strukturlenkja til den preterminale noden i staden for ordlenkjer.

Programmet `lfgalign` følgjer krav (17) og lenkjer øvste nodar i funksjonelle domene, og subordinate nodar som har same informasjonstap. Prosedyren `c-align` i kodefigur 7 implementerer dette kravet.

Hjelpeprosedyren `add-links` (kodefigur 8) utfører hovudjobben. Inndata er rotnoden til c-strukturreet for eitt av språka, og f-samanstillinga. Prosedyren kappar opp treet i nodemengder, kor kvar nodemengd dominerer same lenkjemengd

¹⁰Som nemnt i del 3.7.3 kan funksjonsord gjere f-strukturlenkjinga avhengig av forhold i c-strukturen, evt. krevje meir nyansert f-strukturlenkjing. Dette har eg ikkje teke høgd for i implementasjonen, så der eit funksjonsord burde blokkert ei f-strukturlenkje vil `lfgalign` gi feil samanstilling.

¹¹Dette er ein litt enklare måte å definere kravet på; ei *lenkje* refererer til både kjelde og mål, dimed blir det mogleg å seie at ein node på kjeldespråket kan dominere same mengd som ein node på målspråket.

```

c-alignments  $\leftarrow \emptyset$  ;
splitss  $\leftarrow$  new table ;
add-links(f-alignment, trees, splitss) ;
splitst  $\leftarrow$  new table ;
add-links(f-alignment, treet, splitst) ;
forall the links being the keys in splitss do
    if (links in splitst) then
        | add (splitss[links], splitst[links]) to c-alignments ;
    end
end
return c-alignments ;

```

Funksjon 7: c-align(f-alignment, tree_s, tree_t)

(som definert over). Nodemengdene blir lagra i ein tabell, indeksert på lenkjemengdene. Prosedyren går rekursivt gjennom treet frå rot til lauv; lenkjemengden for kvar node er unionen av lenkjemengdene returnert av `add-links` kalt på kvar av døtrene. Viss ein node dominerer ei lenkjemengd *links*, legg me til denne noden i tabellen *splits*[*links*]. Merk at kvar c-strukturnode berre opptre éin gong i tabellen.

```

links  $\leftarrow \emptyset$ ;
if node then
    if preterminal?(node) then
        | let link  $\in$  f-alignment s.t.  $\phi(\text{node}) \in \text{link}$  ;
        | if link then links  $\leftarrow \{\text{link}\}$  ;
        | if *pro-args-affect-c-links* then
            | forall the a  $\in$  args( $\phi(\text{node})$ ) s.t.  $\phi^{-1}(a) = \emptyset$  do
                | let linka  $\in$  f-alignment s.t. a  $\in$  linka ;
                | if linka then links  $\leftarrow \{\text{link}_a\}$  ;
            | end
        | else
            | links  $\leftarrow$  add-links(f-alignment, left-branch(node))  $\cup$ 
            | add-links(f-alignment, right-branch(node)) ;
            | add node to splits[links] ;
        | end
    end
return links ;

```

Funksjon 8: add-links(f-alignment, node, splits)

Sidan `c-align` kallar `add-links` for kvar av sidene, får me to tabellar *splits_s* og *splits_t*. Me hentar så ut alle dei lenkjemengdene som er i begge tabellane (dvs. snittet av oppslagsnøkklene til tabellen); nodane som er lagra med same mengd med f-strukturlenkjer (same nøkkel i tabellen) skal lenkjast på c-strukturnivå. Alle desse mange-til-mange-lenkjene blir til slutt returnert av `c-align`. Om brukarvariabelen `*pro-args-affect-c-links*` er sann, vil me leggje til lenkja pro-argument i

lenkjemengdene; denne variabelen styrer forskjellen mellom dei to alternative løysingane på ulenkja c-strukturnodar diskutert i del 3.7.1.

Prosesen er no ferdig, mange-til-mange-lenkjene mellom c-strukturnodar definerer frasesamanstillinga

I neste kapittel går eg gjennom resultat av å køyre `lfgalign` på eit lite testsett.

Kapittel 5

Diskusjon

I denne delen gir eg ei evaluering av resultata frå å køyre `lfgalign` på LFG-analysar av parallelle setningar. Eg ser på manglar ved implementasjonen i forhold til dei ideelle krava frå kapittel 3; og kor avhengig `lfgalign` er av bottom-up-informasjon; og samanliknar dei resultata som er moglege å få frå `lfgalign` med dei som er mogleg å få med andre metodar, då spesielt metodar som nyttar N-gramtabellar som kjelde til lenkjingsinformasjon – altså kor fraselenkjer hovudsakleg kjem frå bottom-up-informasjon. I tillegg diskuterer eg kort ulike bruksområde for samanstillingane, og problem som enno er uløyste.

Allereie utan å sjå på materialet, kan me sjå at det er visse føresetnader i implementasjonen (gjort for å forenkle metoden), som kan føre til feil i samanstillinga. T.d. følgjer implementasjonen krav (10) i kapittel 3 (og lenkjar altså berre f-strukturar som er argument/adjunkt av lenkja f-strukturar, eller ikkje er argument/adjunkt av noko). Dette er nok for strengt, men det avskjerer ein god del umotivererte lenkjingar. I tillegg er c-strukturlenkjinga i implementasjonen direkte avleidd av f-strukturlenkjinga; men, som nemnt i del 3.7.3 kan det finnast c-strukturforhold som bør påverke f-strukturlenkjinga. I dette kapittelet ser eg på korleis føresetnadene ved implementasjonen påverkar kva for samanstillingar me får, og kjem fram til at svært ulike eller fragmentariske f-strukturar fører til store feil i lenkjingane.

burde skrive
noko om c-
strukturlenkjinga;
men den ser
jo grei ut...
kan evt.
sjekke kor
mange
lenkjingar
som blir ulike
når me endrar
på variabelen
`pro-affects-c-linking`

5.1 Materiale

Eg nyttar språka georgisk og norsk i evalueringa, hovudsakleg fordi dei er svært ulike syntaktisk og morfologisk. Som nemnt er georgisk eitt av språka nytta i Xpar-prosjektet; og det er nok det som har mest typologisk avstand frå norsk av språka i prosjektet. Georgisk er mellom anna eit pro-drop-språk, med friare ordfølgje og rikare morfologi enn norsk. Georgisk-norsk burde difor vere eit passande språkpar for evalueringa.

Kjeldematerialet mitt for dette språkparet er ei mengd med omtrent tredve LFG-analyserte testsetningar på norsk og georgisk, frå eit testsett kor setninga-

ne er valde for å illustrere ei vid rekkje ulike syntaktiske situasjonar (kalla `mrs` nedanfor), i tillegg til eit par setningar frå Jostein Gaarders *Sofies Verden* (kalla `sofie` nedanfor) på norsk og georgisk. Analysane er manuelt disambiguerte og setningssamanstilte.

Sidan eg ikkje har tilgang på nokon større ferdig setningssamanstilt georgisk-norsk parallelltekst, blir det vanskeleg (utan ein god del forarbeid) å køyre den statistiske ordsamanstillinga som er vanleg som første steg i N-grambaserte metodar¹. Materialet i `mrs` og `sofie` er sjølvstekt alt for lite for statistisk samanstilling, så det vil ikkje vere mogleg å empirisk samanlikne presisjon/dekning med N-grambaserte metodar her. I staden har eg prøvd å sjå på kva for fenomen som *kan* gi problem, og kva for informasjon som for ein spesifikk N-grambasert metode kan vere vanskeleg å hente ut; men dette er ein veikskap med evalueringa.

Eg gjer i tillegg ei overflatisk samanlikning med materialet nytta i RIA Open Source Rule Induction Tool (Graham & Genabith, 2009; Graham et al., 2009).² Kor språka er tysk og engelsk. Dette materialet inkluderer f-strukturlenkjer for 4000 setningspar, fått via ein N-grambasert metode; her samanliknar eg overlapp i samanstillingane. Setningane, og setningslenkjene, er frå Europarl-korpuset (Koehn, 2005). LFG-analysane er automatisk disambiguerte og setningssamanstilte (og inkluderer ein god del fragmentariske analysar); det er difor ganske ulikt det materialet eg elles har sett på.

5.2 N-grambaserte metodar

Dei fleste metodane for frasesamanstilling er N-grambaserte, dvs. at hovudkjelda for lenking er bottom-up-informasjon; difor er det naturleg å samanlikne metoden i `lfgalign` med slike metodar. Eg gir først ein kort introduksjon til korleis N-grambaserte metodar gir ordlenkjer og fraselenkjer, og ser så på kva slags lenkjer dette fører til. Det finst sjølvstekt svært mange ulike moglege metodar og variasjonar på temaet; for å avgrense diskusjonen konsentrerer eg meg om metoden nytta i Samuelsson & Volk (2007), som har som formål å konstruere ein parallell, fraselenkja trebank.

Och & Ney (2003, s. 20–21) gir eit grundig oversyn over ulike samanstillingsmetodar, først og fremst statistiske (og helst med maskinomsetjing som formål). Dei definerer ei samanstilling på det mest generelle som ei delmengd av det kartesiske produktet av ordposisjonane i to setningar. Viss $f_1^J = f_1, \dots, f_j, \dots, f_J$ er orda i setninga på kjeldespråket og $e_1^I = e_1, \dots, e_i, \dots, e_I$ er orda i setninga på målspråket², er ei samanstilling \mathcal{A} gitt ved $\mathcal{A} \subseteq \{(j, i) : j = 1, \dots, J; i = 1, \dots, I\}$. Eitt kjeldeord kan altså vere lenkja til eitt eller fleire målord, og omvendt.

¹Korpuset måtte i tillegg vore LFG-analysert og disambiguert for at eg skulle kunne samanlikne med `lfgalign`. Eg veit heller ikkje enno om nokon statistisk frasestrukturparsar av høg kvalitet for georgisk; testsettet mitt derimot er ferdig parsa med LFG-parsaren frå Meurer (2008), c-strukturnodane avgrensar då kva som er ein syntaktisk konstituent.

²Kor e står for engelsk og f for fransk, sjølvstekt.

Å finne alle slike delmengder er ein komputasjonelt tung jobb, og ikkje eigentleg handterbart for vanlege setningslengder. I praksis vil spesifikke modellar prøve å avgrense dette, t.d. ved å krevje maksimalt eitt målord per kjeldeord, kor ein nyttar «det tomme ordet» for kjeldeord som ikkje er lenkja med noko målord. I litteraturen om statistiske metodar er ei slik mange-ein-samanstilling ei *ordsamanstilling*; når mange-mange-lenkjer er med har me ei *frasesamanstilling*.

Ei samanstilling er her altså ei mengd par av ordposisjonar. Det finst mange ulike måtar å komme fram til samanstillinga på; men grunntanken er at ord som oftare enn forventa opptrer saman i omsette setningar i eit korpus, sannsynlegvis er omsetjingar.

Dei vanlegaste metodane er basert på sannsynsmodellar³, kor den beste samanstillinga a er den som får høgast skåre på $p(f_1^I, a|e_1^I)$, dvs. sannsynet for samanstillinga a og kjeldesetninga gitt målsetninga og parametrane i modellen. Verdien til parametrane finn ein ved å trene modellen på eit parallellkorpus, kor ein god treningsmetode aukar den totale p for heile korpuset (dvs. produktet av p for alle setningane). For å finne p for ordsamanstillingar er det vanleg å nytte ein skjult Markov-modell, dvs. at p for heile setninga blir dekomponert til p for eitt enkelt kjeldeord, kor p av enkeltord er basert på p av orda som kjem før. (I tillegg vil ein ofte vekte på sannsynet for ei viss setningslengd gitt målsetninga, sidan lange setningar ofte blir omsett til like lange setningar, modulo språkforskjellar.) Det er mogleg å forenkle Markov-modellen til at p for eit enkeltord berre er avhengig av ordet som kjem rett før, noko som gjer utrekningane enklare (og lèt ein generalisere på bakgrunn av mindre data), men sjølvsagt gir ein mindre nyansert modell. Om me har forenkla modellen slik at p berre er avhengig av dei N orda som kjem før, har me ein N -grambasert modell.

Ein av dei vanlegaste treningsmetodane er *sannsynsmaksimering* (utførleg forklart i Prescher, 2004), som lèt ein rekne ut elles uhandterlege sannsynsproblem ved å iterativt endre modellparametrane slik at skårene blir betre. Eg gir her eit *svært forenkla* oversyn over korleis dette går føre seg. Me byrjar med å setje p_0 for alle moglege samanstillingar og ordpar til ein viss verdi, kanskje på ein tilfeldig måte. Så genererer me alle moglege samanstillingar til målkorpuset basert på kjeldekorpuset, kor kvar samanstillingsførekomst førekjem med ein frekvens som er vekta ved hjelp av p_0 . Me tel opp samanstillingsførekomstene her i ein tabell; når me ser ordet *Es* i ei setning omsett til *There is*, vil tabellraden med *Es* få auka frekvensskåre i kolonnane til *There* og *is* – men sidan dette var vekta på p_0 treng ikkje dette talet vere 1 eller 0. Så går me gjennom korpuset slik og gir skårar for alle moglege ordomsetjingar, til slutt vil sannsynlegvis raden til *es* og *there* ha høgare skåre enn *es* og *is*. Me reknar så ut ein ny p_1 ved å telje opp dei relative frekvensane til alle tabellradene; dette blir nytta i neste iterasjon for å generere den neste mengda med samanstillingar. Det totale korpussannsynet⁴ vil auke for kvar iterasjon, og

³Det finst au *heuristiske* metodar, t.d. basert på strenglikskap, eller på mål som *Mutual Information* eller Dice-koeffisienten (som kan vise om to ord førekjem oftare saman i omsetjingar enn ein ville forventa); men desse synest å gi dårlegare resultat enn statistiske metodar (Och & Ney, 2003).

⁴Korpussannsynet er produktet av p for alle setningane i heile korpuset, kor p for ei setning er

til slutt flate ut. Som med fjellklatringssøk er me ikkje garantert å finne den beste løysinga med denne metoden (verdiane me har i p_0 har ofte mykje å seie), men me er garantert at korpussannsynet aldri vil bli mindre i neste iterasjon.

Om ein har funne mange-ein-lenkjene mellom ord frå begge sider av eit korpus, kan ein utleie mange-mange-lenkjer på ulike måtar. Gitt ei (kanskje usannsynleg) samanstilling frå tysk til engelsk $\{(Es, is), (gibt, is)\}$, dvs. ei mange-ein-lenkje frå *Es gibt* til *is*, og ei frå engelsk til tysk $\{(There, gibt), (is, gibt)\}$, kan me ta unionen av desse (med den eine retninga reversert), altså $\{(Es, is), (gibt, is), (gibt, There)\}$, som representerer ei mange-mange-lenkje mellom *Es gibt* og *There is*. Me kan au nytte berre snittet, som gir høgare presisjon, men lågare dekning. Koehn et al. (2003), allereie nemnt i kapittel 2, finn mange-mange-lenkjer for frasebasert statistisk maskinomsetjing ved å først ta snittet, og så leggje til frå unionen ved hjelp av ulike heuristikkar.

I Samuelsson & Volk (2007) nyttar dei liknande metodar for å finne ein N-gramtabell, altså ein tabell med kjelde-N-gram, mål-N-gram og sannsyn. Denne sender dei, saman med to frasestrukturannoterte einspråklege trebankar, gjennom eit *lingvistisk samanstillingsfilter* for å opprette ein frasesamanstilt parallell trebank. Viss ein frasestrukturnode i kjeldetrebanken svarer til ein frase i tabellen, og målfrasen i tabellen au er dekkja av ein node på målspråket, vil filteret opprette ei lenkje mellom nodane. N-gramtabellen er ikkje kopla til setningane, så teoretisk sett kan ein opprette lenkjer mellom alle moglege nodepar⁵.

Lenkjene mellom frasestrukture i Samuelsson & Volk (2007) er ein-til-ein; dei skil seg slik formelt frå lenkjene i *lfgalign*. Sidan strengpara dominert av nodelenkjene er ei delmengd av mengda med N-gramkorrespondansar, kor desse altså er kontinuerlege ordstrenger utan kontekst, kan dei ikkje innehalde lenkjer mellom diskontinuerlege konstituentar. Sidan nodelenkjene er basert på ordstrenger vil dei heller aldri ha evidens for eller imot å leggje saman to nodar i ei mange-mange-lenkje (uansett om dei har dominans mellom seg, eller er diskontinuerlege⁶). Dette står i kontrast til c-strukturlenkjing basert på f-strukturforhold – her gir dei funksjonelle domena (saman med dominans) evidens for når ein skal sjå på to nodar som del av same mange-mange-lenkje. Med desse mange-mange-lenkjene mellom c-strukturknodar kan me utan problem tillate lenkjer mellom diskontinuerlege konstituentar. I tillegg vil nodar som, for skuld lenkjinga, bør sjåast på som like, bli handsama som like ved å vere ein del av same mange-mange-lenkje.

Metoden i RIA (Graham & Genabith, 2009; Graham et al., 2009) finn f-strukturlenkjer⁷ frå ordlenkjer, med formålet å lage overføringsreglar for maskinomsetjing. Dei

summen av p for kvar samanstilling i setninga. Om modellen er god, vil auka sannsyn over heile korpuset seie at parametran gir eit betre estimat av dei «faktiske» samanstillingane.

⁵Altså kunne ein ha lenkja frå to søsternodar på kjeldesida til høvesvis mor og dotter på målsida – noko som ikkje ville ha gitt meining. Men dette er lett å unngå ved å krevje at viss ein node n_s er lenkja til n_t , og er dominert av m_s som er lenkja til m_t , må m_t dominere n_t .

⁶Sjølve trebanken (Samuelsson & Volk, 2006) og annotasjonsverktøyet for TreeAligner (tilgjengeleg frå <http://kitt.cl.uzh.ch/kitt/trealigner> under GNU GPL) opnar for manuelle mange-mange-lenkjer, men det er klart at når me går frå par av N-gram til par av frasestrukturnodar som dominerer same N-gram kan me ikkje få mange-mange-lenkjer.

fjernar alle kantar i f-strukturgrafen som kan skape sirkularitet eller som deler sluttnode med andre kantar (og unngår dimesd problemet med *reentrancy*, nemnt nedanfor). F-strukturen blir då, i lenkjingsprosessen, ekvivalent med ein enkel tre-struktur, slik at dei kan nytte ein liknande framgangsmåte som Samuelsson & Volk (2007). Dei nyttar f-strukturhierarkiet til å «normalisere» ordfølgja i setningane, før den automatiske ordlenkjinga. Viss f-strukturane er korrekte og bygd på felles prinsipp, vil altså ordfølgja i inndata til ordlenkjinga bli nokolunde lik (ytre predikat først, så første argument, så andre, osv.), noko som lettar den statistiske ordsamanstillingsprosessen. Dei lenkjer ikkje c-strukturnodar, sidan målet er å lage overføringsreglar på f-strukturnivå.

I del 5.3.2 ser eg på kor stor overlapp det er mellom dei f-strukturlenkjene metoden i RIA får, og dei `lfgalign` får. Kva f-strukturlenkjer `lfgalign` får er direkte avhengig av informasjonen me har om LPT-korrespondansar, så eg byrjar med å sjå på kor viktig denne avhengnaden er.

5.3 Kor avhengig er `lfgalign` av bottom-up-informasjon?

Eitt mål med denne oppgåva er å finne ut av kor lite bottom-up-informasjon ein kan klare seg med, når ein har LFG-analysane å stø seg på. Viss analysane er korrekte, og prinsippa for samanstilling er dekkjande, og implementasjonen av prinsippa er korrekt, er det berre LPT-informasjonen som avgjer om samanstillinga blir korrekt eller ikkje. Sidan c-struktursamanstillinga i `lfgalign` er avleidd av f-struktursamanstillinga, utan påverknad i andre retninga, vil det få følgjer for *heile* samanstillinga dersom manglar i bottom-up-informasjonen gir feil argument/adjunkt-lenkjing. Feil i c-struktursamanstillinga, derimot, vil ikkje få følgjer andre stader (gitt ei viss f-struktursamanstilling har me heller ingen «val» å ta for lenkjinga her).

Setningsparet i (1) illustrerer problemet.

- (1) a. `abramsma brouns sigareti miacoda.`
 \leftrightarrow
b. Abrams rakte Browne sigaretten.

Om me ikkje har LPT-informasjon, blir den urangerte f-struktursamanstillinga:

- (2) `{('mi-codeba' . 'rekke-hand')
{('Browne' . 'Abrams') ('sigareti' . 'Browne') ('Abrams' . 'sigarett')}
{('Browne' . 'Abrams') ('sigareti' . 'sigarett') ('Abrams' . 'Browne')}
{('Browne' . 'Browne') ('sigareti' . 'Abrams') ('Abrams' . 'sigarett')}
{('Browne' . 'Browne') ('sigareti' . 'sigarett') ('Abrams' . 'Abrams')}
{('Browne' . 'sigarett') ('sigareti' . 'Abrams') ('Abrams' . 'Browne')}
{('Browne' . 'sigarett') ('sigareti' . 'Browne') ('Abrams' . 'Abrams')}
}`

Etter rangering får me då

- (3) {('mi-codeba' . 'rekke-hand') ('Browne' . 'sigarett') ('sigareti' . 'Browne')
('Abrams' . 'Abrams')}

som jo er feil (me får då au feil c-struktursamanstilling). Dette skjer pga. argumentstrukturane til dei to verba *ikkje* er parallelle⁷ – 'mi-codeba' har <'Abrams', 'sigareti', 'Browne'>, medan 'rekke-hand' har <'Abrams', 'Browne', 'sigarett'>, og me har eit rangeringskriterium som føretrekkjer lik følgje over ulik. Men ved å t.d. leggje til informasjonen om at *brouns* og *Browne* har LPT-korrespondanse, får me:

- (4) {('mi-codeba' . 'rekke-hand') ('Browne' . 'Browne') ('sigareti' . 'sigarett')
('Abrams' . 'Abrams')}

Me kunne au klart oss med berre informasjonen om at *sigareti* og *sigaretten* har LPT-korrespondanse – så sjølv om LPT-korrespondanse kan vere viktig, viser dette dømet at ein ikkje treng *fullstendig* LPT-korrespondanse for å forbetre resultatet i enkelttilfelle.

Sidan dette er ei openberr feilkjelde, går eg her manuelt gjennom setningane i hovudtestsettet mitt⁸ og utdata frå `lfgalign` for å sjå kor mange av dei som krev LPT-informasjon for å unngå feil lenkjer mellom argument/adjunkt av same predikat. Men merk: sjølv om dette testsettet viser mange ulike syntaktiske fenomen, er det langt frå å vere eit representativt korpus.

Tabell 5.1 viser, for kvart korresponderande og disambiguert setningspar i testsettet, kor mange ord setningane hadde og kor mange LPT-korrespondansar som måtte til for å ikkje få feil i argument/adjunkt-lenkjinga (kolonnen «min. LPT»). For dei setningane der LPT-korrespondansar måtte til, viser kolonnen «moglege LPT-par» kva for moglege mengder med minimal LPT-informasjon som er nok for å få rett analyse (mange mengder i denne kolonnen vil altså seie at det er mange måtar å få rett analyse på). Der eit tal står i parentes har setningsparet andre problem enn berre LPT-korrespondanse som gjer at ikkje alt som skal bli lenkja, blir lenkja. Neste del gir eit par utdjupande kommentarar til analysane, medan i del 5.3.2 ser eg på forskjellane mellom f-strukturlenkjene frå `lfgalign` og dei lenkjene ein kan få ved å gå frå ordsamanstillingar frå f-strukturnormaliserte setningar.

5.3.1 Kommentarar og feilanalyse

Testsettet `mrs` har berre konstruerte døme og eit svært lite ordforråd. Det illustrerer ei vid rekkje syntaktiske fenomen, men omsetjingane er nok svært direkte.

Setning 4.pl og 5.pl på norsk er to alternative måtar å seie setning 4.pl på georgisk (men f-strukturane er like nok til at det ikkje gjer nokon forskjell for lenkjin-

⁷I utgangspunktet kan ein tenkje på *spesifikator*, *adjunkt* og *komplement* som tomme symbol her, definert ved deira posisjon i forhold til XP og X'.

⁸Tilgjengeleg frå mappa `eval` i kjeldekoden til `lfgalign`, som finst på <http://github.com/unhammer/lfgalign>.

ga)⁹.

Berre tri setningspar fekk mange-ein-lenkjer på f-strukturnivå (2.pl/2.pl, 34.pl/35.pl og 38.pl/39.pl); dette skjedde i omsetjing frå verb til hovudverb+hjelpeverb, eller frå verb til verb+refleksivt pronomem. Mange-ein-lenkjene ser ikkje ut til å ha vore noko problem i dette testsettet.

Men setning 67.pl og 68.pl, vist i (5) nedanfor, er problematiske:

- (5) a. *abramsis suraTi movida.*
↔
b. Bildet av Abrams ankom.

Her er *abramsis* ein POSS under SPEC av f-strukturen til *suraTi*, i staden for å vere eit adjunkt/argument (f-strukturen til *Abrams* står som argument til '**bilde**'). Programmet finn ikkje f-strukturar via andre stigar enn adjunkt eller argument til tidlegare kjende f-strukturar, eller via «yttarste» f-strukturar – dette er krav (10) frå kapittel 3. Sidan *abramsis* ikkje er mogleg å finne via argument/adjunkt av rotpredikatet, ender det opp i ei eiga liste for fragment ol. som kan lenkjast, men kor me ikkje veit noko om konteksten. Det blir altså handsama som eit av fleire yttarste predikat – men her burde det jo vere mogleg å finne det via *suraTi*. På den norske sida er *Abrams* mogleg å finne gjennom *bilde*, og er ikkje sett på som noko yttarste predikat, så *abramsis* og *av Abrams* blir ikkje lenkja.

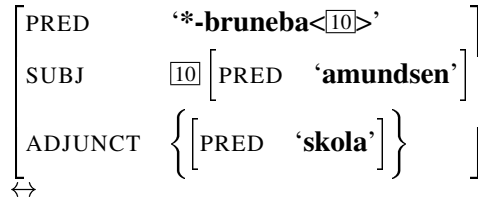
Dette dømet demonstrerer ein veikskap med *lfgalign*: om ein skal køyre programmet med andre LFG-grammatikkar, som kanskje har andre retningslinjer for analyse eller har gjort ting på litt spesielle måtar, må ein kanskje leggje til fleire unntak eller meir «kunnskap» (evt. preprosessere analysane før lenkjing). Generelt kan dette forsåvidt vere eit problem med alle metodar som er svært kunnskapsbaserte.

Testsettet *sofie* er frå ein roman, og har difor litt friare omsetjingar enn *mrs.*

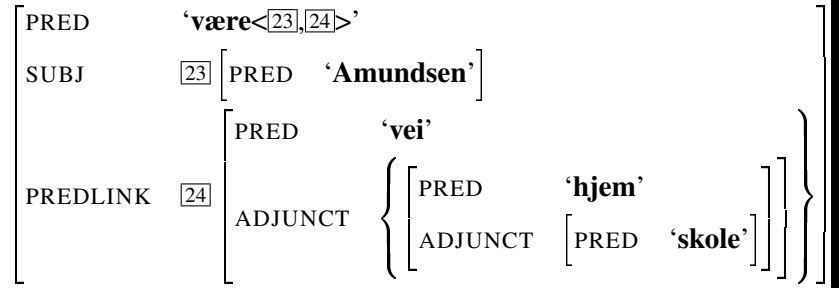
I setning 2.pl og 0.pl har konstruksjonen *være på vei hjem* blitt omsett til *brune-ba*, direkte omsett 'dreie'. F-strukturane i (6) illustrerer forskjellen (adposisjonane er fjerna for å få plass til figuren); i (7) ser me f-struktursamanstillinga som me får etter at *amundsen/Amundsen* og *skola/skole* er lagt til i LPT-tabellen.

- (6) a. *soPi amundseni skolidan brundeboda.*
Sofie Amundsen skole.fra dreie.3SG.IMP.

⁹Det finst ingen 8.pl på georgisk i tabellen sidan denne (som hadde tilsvart 9.pl på norsk) ikkje hadde analyse, difor manglar dette setningsparet – andre som manglar frå tabellen var anten ikkje disambiguerte eller hadde ikkje analysar. Eit par som ikkje var disambiguerte har eg sjølv manuelt disambiguert (norsk 0.pl og georgisk 3.pl). I setning 57.pl og 58.pl er eg usikker på kva den rette analysen bør vere (eller om har rette LFG-analysar), men meir LPT-informasjon vil iallfall ikkje endre på det.



b. Sofie Amundsen var på vei hjem fra skolen.



(7) {('*-bruneba' . 'være') ('*-bruneba' . 'vei') ('amundsen' . 'Amundsen'))}

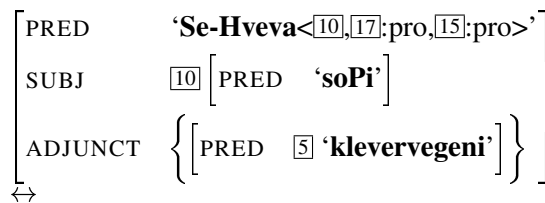
Samanstillinga kjem ikkje djupare inn i f-strukturane enn dette. Her burde kanskje heile *være på vei hjem* lenkjast med *bruneba*, altså ei ein-til-tri-lenkje (på tri ulike nivå) på f-strukturnivå, for at me skal kunne lenkje 'skola' og 'skole'¹⁰. Om det finst mange slike situasjonar bør me kanskje opne for meir kompliserte lenkjer på f-strukturnivå. Merk at om me ikkje legg til *skola/skole* i LPT-tabellen får me samanstillinga i (8):

(8) {('*-bruneba' . 'være') ('skola' . 'vei') ('amundsen' . 'Amundsen'))}

Altså, slik implementasjonen av LPT-kravet fungerer kan positiv LPT-informasjon (det at me veit at eit ordpar har LPT-korrespondanse) føre til at lenkjer kan fjernast, medan me tillét lenkjene om me manglar informasjon. Så om eit ord har eit oppslag i LPT-tabellen i det heile, bør det eigentleg ha alle moglege oppslag, for å unngå å fjerne for mange gode lenkjer. Eg kjem tilbake til dette i del 5.4 nedanfor.

Setning 13.pl og 10.pl i *sofie* har liknande problem; for store forskjellar i argumentstruktur gir feil løysing.

(9) a. soPim klevervegenisken SeuHvia.
Sofie Kløverveien.mot svinge.3SG.PERF.



¹⁰Eller så kunne ein tenkje seg å «hoppe over» mellomliggande argument for å komme fram til det lenkbare argumentet, altså berre lenkje 'være' med '*-bruneba' og 'skole' med 'skole' – men noko slikt ville involvert store endringar i krava for f-strukturlenkjing.

- b. Nå svingte hun inn i Kløverveien.

$$\left[\begin{array}{ll} \text{PRED} & \text{'svinge'} \langle \boxed{18} : \text{hun} \rangle \\ \text{ADJUNCT} & \left\{ \boxed{5} \left[\text{PRED} \quad \text{'inn'} \right], \boxed{6} \left[\text{PRED} \quad \text{'nå'} \right] \right\} \end{array} \right]$$

$$(10) \quad \{(\text{'Se-Hveva'} \cdot \text{'svinge'}) (\text{'pro'} \cdot \text{'hun'}) (\text{'pro'} \cdot \text{'nå'}) (\text{'soPi'} \cdot \text{'inn'})\}$$

F-strukturane i (9) gir lenkjinga i (10). Desse pro-elementa på georgisk skal sjølv-sagt *ikkje* lenkjast med adjunkta på norsk; her har me «overgenerering» av lenkjer pga. me opnar for argument-adjunkt-lenkjer. Slik *lfgalign* fungerer no, kan ein unngå at pro blir lenkja til adjunkt ved å leggje inn LPT-informasjon om adjunkta (evt. krevje at pro *berre* kan lenkjast til nominalar). Om me legg inn ei omsetjing for *nå*, får me samanstillinga i (11):

$$(11) \quad \{(\text{'Se-Hveva'} \cdot \text{'svinge'}) (\text{'pro'} \cdot \text{'svinge'}) (\text{'pro'} \cdot \text{'hun'}) (\text{'soPi'} \cdot \text{'inn'})\}$$

Denne samanstillinga er kanskje er hakket betre, men sidan mange-mange-lenkjing berre er to-til-ein så langt, vil LPT-informasjon om *både* *'nå'* og *'inn'* resultere i at denne setninga *ikkje får nokon lenkjing*. Dette er kanskje å føretrakkje, når me først skal krevje likskap i argumentstruktur for å lenkje¹¹. Då er det opp til grammatikarane om dei vil sjå på omsetjinga som argument for at det georgiske verbet har ei alternativ tyding med berre eitt argument, og endre grammatikkane til ei seinare utgåve av trebanken.

5.3.2 Overlapp med RIA

Eg samanliknar her f-strukturlenkjene ein får ved å køyre *lfgalign* på ei mengd med 4000 automatisk disambiguerte og setningslenkja setningar på tysk og engelsk, med dei lenkjene som kjem frå RIA-metoden¹² i Graham & Genabith (2009); Graham et al. (2009). Eg prøvde både utan nokon informasjon i LPT-tabellen, og med ein ganske stor LPT-tabell med alle oppslaga frå Ding-ordboka¹³. I tillegg køyrte eg testane med delmengder av testsettet kor eg filtrerte ut analysar med mange «yttarste f-strukturar», for å sjå kva slags innverknad dei hadde på resultatet.

¹¹Om eit slikt setningspar er gitt til ein N-grambasert metode, vil det sjølv-sagt gi evidens for at t.d. *nå* og det georgiske verbet skal lenkjast – ikkje noko me vil ha i ein trebank. Men, om *nå* sjeldan står i omsetjingar av *SeuHvia* vil det sannsynlegvis aldri bli oppretta slike lenkjer.

¹²Prolog-filene med f-strukturar, og lenkjer mellom f-strukturar, for 4000 parallelle setningar finst i kjeldekoden til RIA, tilgjengeleg frå <http://www.computing.dcu.ie/~ygraham/software.html> under GNU Lesser General Public License. I tillegg finst Prolog-filer som inkluderer c-strukturar for 1000 setningar på http://www.computing.dcu.ie/~ygraham/sample_de.tar.gz og http://www.computing.dcu.ie/~ygraham/sample_en.tar.gz.

¹³Tilgjengeleg frå <http://www-user.tu-chemnitz.de/~fri/ding/> under GNU GPL, konvertert til Prolog-format med skriptet *dev/ding-to-LPT.py* (i kjeldekoden til *lfgalign*). Ordboka har omtrent 270 000 oppslag; men sidan ein del av dei involverer alternative omsetjingar tok eg det kartesiske produktet av alle slike alternativ, og endte opp med 418 492 par av token (men 190 284 av desse har mellomrom i seg, og fungerer sannsynlegvis ikkje som LPT-omsetjingar).

Samanlikninga skjer ved å, for kvar setning i korpuset, telje opp kor mange lenkjer som finst i snittet mellom dei f-strukturlenkjene som er i RIA og dei som lfgalign gir. Dette kan me samanlikne med kor mange f-strukturlenkjer RIA har i det heile for å få ein slags «dekning», og med kor mange lenkjer lfgalign har for å få ein slags «presisjon» – dette føreset sjølvsagt at RIA-lenkjene er korrekte.

Viss me har to setningar som kvar har tri PRED-element, finst det seks måtar å lenkje desse på slik at alle blir lenkja. Viss lfgalign har same samanstilling som RIA, vil snittet av dei to ha like mange lenkjer som RIA har, og like mange som lfgalign har (medan viss det er eitt av desse orda som lfgalign ikkje har lenkja, vil snitt / RIA gå ned, medan snitt / lfgalign held seg likt). For å få ein peikepinn på kor mykje likskapen har å seie (ein slags indikasjon på effektstyrke), har eg au køyrt ein baseline-modell på testsettet, som berre opprettar tilfeldige lenkjer mellom lenkbare PRED-element (slik at flest mogleg ein-til-ein-lenkjer blir oppretta; denne modellen har ingen mange-mange-lenkjer).

Desse tala seier sjølvsagt ingenting om kva for lenkjer som er *korrekte*. Men, lenkjene i RIA-testmaterialet kjem via ordsamanstillingsmetoden i Och & Ney (2003), kor «even on a tiny corpus of only 500 sentences, alignment error rates under 30% are achieved for all models [for English-German], and the best models have error rates somewhat under 20%» (Och & Ney, 2003, s. 36). Feilraten til modell 6, som RIA nyttar, går under 10 % ved korpora på fleire tusen ord. Om f-strukturlenkjene i lfgalign har stor overlapp med dei i testsettet er det altså sannsynleg at mange av dei er korrekte, om det er stor forskjell er det sannsynleg at lfgalign tek feil.

Tabell 5.2 viser resultatet av samanlikninga¹⁴. Kolonnen *snitt* viser summen av kor mange lenkjer som var felles for kvart setningspar, mellom RIA og dei tri testmetodane (lfgalign med og utan informasjon i LPT-tabellen, og ein tilfeldig baseline). Ved å dele dette talet på kor mange lenkjer RIA hadde, får me ein enkel samanlikning av dei to testmetodane – viss forholdet var 100 % ville altså alle lenkjene vore like, og testmetoden ville ha funne alle lenkjene som RIA fann (dekning). Viss forholdet mellom snittet og lenkjene i testmetoden var 100 % ville alle lenkjene i testmetoden ha vore inkludert i RIA (presisjon). Samanlikninga er køyrt med ulike delmengder av testsettet; i dei tri første radene har eg fjerna alle setningar som hadde meir enn ein yttarste PRED utanom rot-f-strukturen (som alltid har indeks 0 i dette formatet); i dei neste har eg i tillegg med dei som hadde maksimalt ein annan yttarste PRED, osb. (kolonnen *maks. yttarst*).

Det er tydeleg at overlappet mellom RIA og lfgalign er ganske lågt på dette materialet, men det er iallfall meir enn dobbelt så stort som ein ville forvente om det ikkje var nokon likskap. Ved manuelt gjennomsyn ser det ut til at der dei har ulike lenkjer, er det lfgalign (av og til begge) som tek feil.

Det som au er slåande er kor negativt ordboka spelar inn på dekninga. Det

¹⁴Testen utan ordbok blei køyrt på revisjon fd14d902262a9274b42131cf83ade74d51e23863 av lfgalign, testen med ordbok blei køyrt på revisjon 796824e8f1a0bd643caf9d3928fe0f979a360790. Sjå tillegg A.

er `lfgalign` som har færrest lenkja PRED-element av alle metodane der me har med det fulle testsettet (dekninga er spesielt låg der ordboka er med). Viss me derimot har fjerna analysar som har fleire «yttarste PRED», har `lfgalign` (utan ordbok) faktisk fleire enn RIA (`lfgalign` kan føye saman eitt predikat og argument i lenkjinga, noko RIA og baseline-modellen ikkje gjer, men det ser ikkje ut til at dette speler så veldig sterkt inn på kor mange lenkjer me får).

Tidlegare nemnte eg at positiv LPT-informasjon kan føre til at lenkjer blir fjerna – dette blir ganske tydeleg her. Når ordboka er med, får `lfgalign`, relativt konsekvent, berre ein fjerdedel så mange lenkjer som RIA. Det er kanskje ein del feil blant lenkjene i RIA-testsettet; men sjølv om halvparten av lenkjene til RIA var feil, burde me sjå høgare snitt mellom `lfgalign` og RIA viss ordboka førte til betre lenkjer. Det viser seg allereie med første ord i testsettet kvifor dette skjer: ordboka omset engelsk *adoption* med tysk *Adoption* (og gir ingen andre omsetjingar), medan *adoption* i denne setninga er nytta i ei anna tyding og blir omsett til *Genehmigung* (som har ti andre omsetjingar i Ding, men altså ikkje den me er ute etter). Det er sannsynleg at dei gjenståande lenkjene er av høg presisjon, men med ei så låg dekning (12 % viss RIA har rett, 8 % viss me ser på alle lenkbare predikat) er nok ikkje samanstillinga veldig nyttig. Eg diskuterer ei mogleg løysing på dette problemet i del 5.4.

Dette testsettet har svært mange «yttarste f-strukturar» (gjerne over ti i kvar setning, i snitt fire per setning, medan gjennomsnittleg setningslengd i materialet er 9,8 ord). Implementasjonen min reknar ein f-struktur for å vere ein av desse yttarste om han ikkje er referert til som argument/adjunkt av andre f-strukturar; dette kan vere fordi analysane er fragmentariske, eller analysane rett og slett ikkje har ein sti til desse PRED-elementa som går via argument/adjunkt. Det siste gjeld t.d. POSS- og SPEC-element, som nemnt over, t.d. kvantorar, possessivar og artiklar. Det ser ut til at 1473 av 4000 setningspar har fragmentariske analysar på den eine sida; men 3948 av 4000 har meir enn eitt yttarste PRED (dvs. at det berre er 52 setningspar i dei første tri radene i tabell 5.2). Hovudgrunnen til at setningar i dette testsettet har fleire yttarste PRED-element er altså at slike POSS- og SPEC-element, eller liknande, ikkje er mogleg å finne gjennom argument eller adjunkt frå rot-f-strukturen.

Yttarste f-strukturar blir handsama som om dei er på same nivå i f-strukturen – viss det eigentleg finst ein intern struktur mellom desse blir det ignorert. Det vil seie at for store delar av analysane får `lfgalign` ingenting ut av f-strukturen. Sidan programmet er meint for trebankbygging med høg presisjon treng det ikkje vere negativt at dekninga er *litt* lågare, men når det gjeld SPEC- og POSS-element bør det ikkje vere alt for vanskeleg å utvikle litt meir nyanserte lenkjingskriterium som au kan handtere desse.

Ein annan tydeleg, men kanskje ikkje overraskande, trend i tabellen er at «presisjonen» til `lfgalign` med og utan ordbok nestan er heilt lik der me ikkje har med fleire yttarste PRED (kor dei resterande f-strukturane altså er svært informative), 52 % utan og 56 % med ordbok. Men med ei gong me legg til «flate» f-strukturar, får me store forskjellar mellom dei to metodane i presisjonskolonnen – 30 % utan og

46 % med ordbok. Jo mindre struktur analysane gir oss (eller, jo mindre me er i stand til å hente ut), jo meir avhengig er me av bottom-up-informasjon.

I Xpar-prosjektet vil me nok forvente meir «komplette» analysar enn dei som finst i dette testsettet, som altså hadde svært mange fragment (og i tillegg til automatisk disambiguering). Likevel bør det vere mogleg å gjere noko for å betre på yteevna når analysane er fragmentariske; eg kjem tilbake til dette nedanfor.

Eit problem som blir tydeleg med denne evalueringa er at lange lister med argument/adjunkt raskt blir komputasjonelt vanskeleg. For «vanlege» lister med argument/adjunkt (t.d. to argument og tri-fire adjunkt), er det ikkje merkbart at alle moglege kombinasjonar av argument/adjunkt blir prøvd lenkja; men når det er snakk om ti «adjunkt» (her: yttarste f-strukturar) på kvar side, og ingen LPT-tabell, tek det `adjoin` ca. 46 sekund (på ein vanleg datamaskin med 2,1 GHz, 3GB RAM) å finne alle moglege lenjekombinasjonar. For å i det heile teke komme meg gjennom testsettet måtte eg setje ei grense på `lfgalign` i denne evalueringa slik at me berre vel første lenjekombinasjonen viss adjunktlistene har over åtte medlem¹⁵. Med så flate strukturar og ingen LPT-informasjon er det nok usannsynleg å ende opp med rett kombinasjon uansett. Med denne «avskjeringa» på plass tok heile testsettet på 4000 setningspar rett under tri timar å samanstillе når eg ikkje hadde nokon informasjon i LPT-tabellen med (med ordboka tok det under eitt minutt, her blir altså *svært mange* løysingar avskjert; me får sjølvsagt au raskare samanstilling med færre yttarste PRED). Det kan hende at lenkjeoverlappet hadde vore litt høgare med eit fullstendig søk, men avskjeringa skjer berre på omtrent kvart tiande setningspar, så den endelege verknaden på resultata i tabell 5.2 (dei siste tri radene) er nok ikkje stor.

Om ein skal konkludere noko frå denne samanlikninga må det vere at

1. Fragmentariske f-strukturar gir dårleg samanstilling (garbage in, garbage out)
2. Det finst fleire PRED-element i f-strukturane som me ikkje finn via argument-lister/adjunktmengder, men som me burde kunne lenkje (krava frå kapittel 3 må utvidast for å handtere desse)
3. Me kan ikkje stole på at ei ordbok har alle dei moglege omsetjingane av eit ord – implementasjonen av LPT-kravet må bli litt meir nyansert
4. Jo mindre top-down-informasjon f-strukturane gir, jo viktigare er bottom-up-informasjonen i LPT-tabellen

Nedanfor ser eg på moglege måtar å betre på systemet.

5.4 Opne problem

I denne delen diskuterer eg nokre opne problem, og moglege strategiar for å betre metoden.

¹⁵Eg gjorde ingen slike avskjeringar for argumentpermutasjonane.

5.4.1 Fragmentariske analysar

Som nemnt i del 5.3.2 har `lfgalign` problem med fragmentariske f-strukturanalysar.

Ein mogleg strategi for å handsame desse vil vere å dele opp lenkjingsoppgåva slik at me først finn fragment som sannsynlegvis korresponderar (ein god heuristikk er ganske enkelt ordfølgje), og så sender kvar av desse til vanleg f-strukturlenkjing. Men der fragmenta ikkje overlappar vil me få problem. Ein av grunnføresetnadene til `lfgalign` er at analysane har ein viss likskap, og er korrekte; ei god løysing på dette problemet vil nok gå utover oppgåva.

veit ikkje kor mykje meir eg skal seie her...

Ein annan strategi, som iallfall løyser problemet med effektivitet, er å gjere eit ufullstendig søk etter adjunktpermutasjonar. Det å finne alle kombinasjonar av lenkjer mellom «flate» f-strukturar der me har over åtte ord på kvar side tek for lang tid til at det blir nyttig. Ein filtrering av moglege lenkjer basert på t.d. LPT-korrespondanse kan gjere ei betre avskjering enn me får med berre ordfølgje.

5.4.2 «Mjuk» LPT-korrespondanse, og rangering som avskjering

Ufullstendig informasjon i LPT-tabellen kan, som nemnt i del 5.3.1 og 5.3.2, føre til at programmet avskjerer for aggressivt. Jo friare omsetjingane er, jo større blir dette problemet.

I implementasjonen bør kanskje kravet om LPT-korrespondanse difor bli eit rangeringskriterium i implementasjonen, slik at me ikkje treng stole på at bottom-up-informasjonen vår er perfekt. Eg har ikkje prøvd å implementere dette enno. Ved å berre finne dei k beste argument-/adjunkt-permutasjonane kan dette kanskje gjere lenkjinga litt raskare au¹⁶.

5.4.3 Top-down-lenkjing av f-strukturar, og problemet med reentrancy

Setningsparet i (12) bør kunne lenkjast, men gir nokre intrikate problem.

- (12) a. Dafür bin ich zutiefst dankbar
for.det er eg djupt takknemleg

$$\left[\begin{array}{ll} \text{PRED} & \text{'bin}<\boxed{5}, \boxed{1}>' \\ \text{SUBJ} & \boxed{5} \left[\text{PRED} \text{'ich'} \right] \\ \text{XCOMP} & \boxed{1} \left[\text{PRED} \text{'dankbar}<\boxed{5}, \boxed{3}>' \right. \\ & \left. \text{OBJ} \quad \boxed{3} \text{'für}<\text{da}>' \right] \end{array} \right]$$

↔

Standarddømet med reentrancy er i `mrs/ka/37.pl` `mrs/nb/38.pl`; `jaGli/hund` er subjekt både for 'ville' og 'bjeffe', difor får løysinga ut `jaGli-hund` to gonger.

¹⁶For å ta dette eit steg vidare kan me flytte sjølve rangeringa inn i f-strukturlenkjinga, for å gjere implementasjonen litt meir effektiv. Dette kan relativt enkelt gjerast ved å endre rekursive kall på `f-align` til kall på `rank(f-align)`, og endre `rank` til å levere ei liste med dei k beste alternativa. Men, det synest som om flaskehalsen er generering av argument-/adjunkt-permutasjonar, så eg veit ikkje kor mykje dette vil ha å seie.

- b. I have a deep appreciation for that
Eg har ei djup takksemd for det

$$\left[\begin{array}{ll} \text{PRED} & \text{'have'} \langle \boxed{4}, \boxed{2} \rangle \\ \text{SUBJ} & \boxed{4} \left[\text{PRED} \text{'I'} \right] \\ \text{XCOMP} & \boxed{2} \left[\begin{array}{ll} \text{PRED} & \text{'appreciation'} \\ \text{ADJUNCT} & \left\{ \left[\text{PRED} \text{'for'} \langle \text{that} \rangle \right] \right\} \end{array} \right] \end{array} \right]$$

I lfgalign vil me kunne lenkje **'bin'** og **'have'**, med LPT-korrespondanse mellom **'ich'** og **'I'** og mellom **'dankbar'** og **'appreciation'**. Når me prøver å lenkje **'dankbar'** med **'appreciation'** har me LPT-korrespondanse mellom **'da'** og **'that'**, men me finn ingenting å lenkje **'ich'** med. Dette argumentet har LPT-korrespondanse med **'I'**, og er jo allereie lenkja med **'I'** sidan det er identisk med subjektet til **'bin'** – men når me ser det igjen på eit djupare nivå har me ingen element på det nivået å lenkje med. Verken krava i kapittel 3 eller implementasjonen av dei tek innover seg slike problem.

Ein *kan* gjere unntak i krav (8) om at argument som allereie har ei lenkje på ytre nivå ikkje skal lenkjast – altså må eit argument berre lenkjast «ein gong». Dette bør løyse problemet over, og bør ikkje vere for vanskeleg å implementere heller.

Men ein variant av situasjonen over kan gi litt større problem. I (13) kan me tenkje oss at den ønskelege lenkjinga er $\{('p', 'r'), ('q', 's'), ('ich', 'I')\}$. Den einaste relevante forskjellen frå (12) er at i (12) står **'I'** berre i ytre PRED, medan i (13) står **'I'** berre i indre PRED.

$$(13) \quad a. \quad \left[\begin{array}{ll} \text{PRED} & \text{'p'} \langle \boxed{1}, \boxed{2} \rangle \\ \text{SUBJ} & \boxed{1} \left[\text{PRED} \text{'ich'} \right] \\ \text{XCOMP} & \boxed{2} \left[\text{PRED} \text{'q'} \langle \boxed{1} \rangle \right] \end{array} \right]$$

↔

$$b. \quad \left[\begin{array}{ll} \text{PRED} & \text{'r'} \langle \boxed{3} \rangle \\ \text{XCOMP} & \boxed{3} \left[\begin{array}{ll} \text{PRED} & \text{'s'} \langle \boxed{4} \rangle \\ \text{SUBJ} & \boxed{4} \left[\text{PRED} \text{'I'} \right] \end{array} \right] \end{array} \right]$$

Då er det eit problem at me følgjer krav (10) og går top-down i f-strukturlenkjinga i lfgalign. Når me skal lenkje **'p'** og **'r'** vil me ikkje finne ein argumentpermutasjon (ein-til-ein LPT-korrespondanse) som fyller krav (8) – **'q'** og **'s'** har LPT-korrespondanse, men **'ich'** har ikkje noko som svarer til seg blant argument/adjunkt av **'r'**.

Det er sjølvsagt mogleg å køyre top-down-metoden frå fleire startpunkt i f-strukturane dersom dei første ikkje går, men om ein må gjere det er det kanskje eit

teikn på at heile strategien må endrast. På den andre sida, viss desse situasjonane ikkje dukker opp i faktiske analysar, så kan det hende at metoden er god nok. Til no har eg ikkje sett slike situasjonar, men det er altså logisk mogleg at dei kan dukke opp.

5.5 Bruksområde

dette blei tynt; flytte til oppsummering kanskje?

5.5.1 Oppdage argumentstrukturalternasjon

I døme (5) i kapittel 3.6.1 viste eg at f-strukturar og LPT-korrespondanse ikkje alltid har nok informasjon til å kunne handtere ulik følgje i argumentstruktur. Programmet mitt vil her gi begge løysingar, ei rangering basert på lik argumentfølgje vil gi feil løysing på topp.

Kanskje kan me nytte data frå fleire førekomstar med andre subjekt og objekt til å lære slike argumentstrukturalternasjonar. Om me observerer *sie gefällt mir/jeg liker henne* vil me jo ha f-strukturinformasjon som kan nyttast til å informere argumentstrukturalternasjon (*sie/henne* er hokjønn, etc.), om det var substantiv der ville LPT-korrespondanse kunne informere dette.

5.6 Oppsummering

- Sidan avbildinga frå c-strukturknodar til f-struktur er mange-til-ein, kan me t.d. innanfor eitt tre ha fleire N-gram per f-strukturhovud; ein metode som berre ser på enkle N-gram-lenkjer vil ikkje registrere desse relasjonane (t.d. metoden i Samuelsson & Volk (2007)).
- Det er ikkje berre ei N-gramsamanstilling; sidan lenkjene er mellom c-strukturknodar kor kvar node dominerer ein konstituent, kunne me kalt det ei konstituentsamanstilling.
- Lenkjer mellom f-strukturar vil gi informasjon som er kvalitativt forskjellig frå det ein kan få med å berre sjå på lenkjer mellom ord, n-gram eller konstituentar.
- Ein kan kanskje få betre skåre på kvantitative mål som presisjon og gjenkjenning, spesielt kor språkforskjellane er store men analysane tufta på felles prinsipp, men her må det meir testing (og ikkje minst data) til.
- `lfgalign` stiller høge krav til inndata: det må vere ferdig disambiguert og setningssamanstilt. I tillegg må grammatikkane som gav analysane vere fundert på like prinsipp, for å få best mogleg resultat. Det kan au oppstå problem viss ein køyrer programmet på analysar frå grammatikkar som, sjølv om dei

er fundert på like prinsipp, har svært ulike måtar å representere ulike grammatiske fenomen enn det grammatikkane i Xpar-prosjektet her.

Tabell 5.1: Kor mykje bottom-up-informasjon treng me for å lenkje argument/adjunkt korrekt? Kolonnane l_s og l_t er lengd på setningane.

setning	l_s	l_t	min. LPT	moglege LPT-par
mrs 0.pl 0.pl	1	2	0	
mrs 1.pl 1.pl	2	2	0	
mrs 2.pl 2.pl	3	2	1	{(ga-Geba,pro)}, {(PanJara,vindu)}
mrs 3.pl 3.pl	3	3	0	
mrs 4.pl 4.pl	4	4	1	{(Browne,Browne)}, {(sigareti,sigarett)}
mrs 4.pl 5.pl	4	5	1	{(Browne,Browne)}, {(sigareti,sigarett)}
mrs 5.pl 6.pl	6	10	1	{(Abrams,Abrams)}, {(sigareti,sigarett)}
mrs 6.pl 7.pl	4	5	0	
mrs 7.pl 8.pl	3	4	0	
mrs 9.pl 10.pl	3	3	0	
mrs 10.pl 11.pl	5	5	0	
mrs 11.pl 12.pl	3	3	0	
mrs 12.pl 13.pl	2	2	0	
mrs 16.pl 17.pl	2	2	0	
mrs 19.pl 20.pl	2	2	0	
mrs 22.pl 23.pl	3	3	0	
mrs 23.pl 24.pl	2	2	0	
mrs 24.pl 25.pl	3	3	0	
mrs 25.pl 26.pl	3	4	0	
mrs 26.pl 27.pl	2	2	0	
mrs 34.pl 35.pl	2	3	0	
mrs 37.pl 38.pl	3	3	0	
mrs 38.pl 39.pl	2	5	0	
mrs 57.pl 58.pl	3	4	0	
mrs 63.pl 64.pl	3	4	0	
mrs 67.pl 68.pl	3	4	(0)	
mrs 71.pl 72.pl	4	4	0	
mrs 73.pl 74.pl	4	4	0	
sofie 2.pl 0.pl	4	8	(2)	{(amundsen,Amundsen),(skola,skole)}
sofie 13.pl 10.pl	3	6	(0)	

Tabell 5.2: Overlapp mellom RIA og lfgalign, og mellom RIA og ein tilfeldig baseline. Det fulle testsettet har totalt 40343 lenkbare PRED-element på kjeldesida; 337 når alle setningspar med fleire yttarste PRED er fjerna.

metode	maks. yttarst	snitt	union	lenkjer	RIA-lenkjer	snitt / denne	snitt / RIA
tilfeldig	0	43	526	307	262	14,01 %	16,41 %
lfgalign u/ordbok	0	142	386	272	262	52,21 %	54,20 %
lfgalign m/ordbok	0	51	302	91	262	56,04 %	19,47 %
tilfeldig	1	292	4151	2426	2017	12,04 %	14,48 %
lfgalign u/ordbok	1	1066	3029	2127	2017	50,12 %	52,85 %
lfgalign m/ordbok	1	409	2327	725	2017	56,41 %	20,28 %
tilfeldig	2	682	12153	7005	5830	9,74 %	11,70 %
lfgalign u/ordbok	2	2586	9028	5890	5830	43,90 %	44,36 %
lfgalign m/ordbok	2	985	6684	1856	5830	53,07 %	16,90 %
tilfeldig	3	1191	23789	13765	11215	8,65 %	10,62 %
lfgalign u/ordbok	3	4262	17700	10949	11215	38,93 %	38,00 %
lfgalign m/ordbok	3	1652	12796	3261	11215	50,66 %	14,73 %
tilfeldig	∞	2544	60544	35069	28019	7,25 %	9,08 %
lfgalign u/ordbok	∞	7296	44587	24161	28019	30,20 %	26,04 %
lfgalign m/ordbok	∞	3382	31970	7375	28019	45,86 %	12,07 %

Kapittel 6

Avslutning

Referansar

- Bresnan, J. (2001). *Lexical-Functional Syntax*. Oxford, UK: Blackwell Publishers. Tilgjengeleg frå <http://books.google.com/books?id=7elu0CcxQWkC> (ISBN: 0631209743)
- Brown, P.F., Della Pietra, S.A., Della Pietra, V.J. & Mercer, R.L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 263–311. Tilgjengeleg frå <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.8919>
- Butt, M. (1998). Constraining Argument Merger Through Aspect. I E. Hinrichs, A. Kathol & T. Nakazawa (red.), *Complex predicates in nonderivational syntax* (vol. 30, kap. 1). New York: Academic Press.
- Butt, M., Dyvik, H., King, T., Masuichi, H. & Rohrer, C. (2002). The Parallel Grammar Project. I *COLING-02 on Grammar engineering and evaluation* (vol. 15, s. 1–7). Morristown, NJ: Association for Computational Linguistics. Tilgjengeleg frå <http://portal.acm.org/citation.cfm?id=1118783.1118786>
- Chen, S. (1993). Aligning sentences in bilingual corpora using lexical information. I *Proceedings of the 31st annual meeting on association for computational linguistics* (s. 9–16).
- Cheung, L., Lai, T., Luk, R., Kwong, O., Sin, K., Tsou, B. et al. (2002). Some Considerations on Guidelines for Bilingual Alignment and Terminology Extraction. , 1–5. Tilgjengeleg frå <http://www.aclweb.org/anthology-new//W/W02/W02-1802.pdf>
- Dyvik, H., Meurer, P., Rosén, V. & Smedt, K.D. (2009). Linguistically motivated parallel parsebanks. I M. Passarotti, A. Przepiórkowski, S. Raynaud & F.V. Eynde (red.), *Proceedings of the eighth international workshop on treebanks and linguistic theories* (s. 71–82). Milan, Italy: EDUCatt. Tilgjengeleg frå http://tlt8.unicatt.it/allegati/Proceedings_TLT8.pdf#page=83
- Graham, Y., Bryl, A. & Genabith, J. van. (2009). F-structure transfer-based statistical machine translation. I M. Butt & T.H. King (red.), *Proceedings of lfg09*

- (s. 317–337). Trinity College, Cambridge: CSLI Publications. Tilgjengeleg frå <http://pargram.b.uib.no/references/2009s/>
- Graham, Y. & Genabith, J. van. (2009). An Open Source Rule Induction Tool for Transfer-Based SMT. *The Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for Machine Translation*, 91, 37–46.
- Graham, Y. & Genabith, J. van. (2010). Deep Syntax Language Models and Statistical Machine Translation. Beijing, China: The 23rd International Conference on Computational Linguistics. Tilgjengeleg frå <http://www.computing.dcu.ie/~ygraham/graham-vangenabith-10.pdf>
- Hearne, M., Ozdowska, S. & Tinsley, J. (2008). Comparing Constituency and Dependency Representations for SMT Phrase-Extraction. I *Actes de la 15e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN '08)*. Avignon, France. Tilgjengeleg frå <http://www.computing.dcu.ie/~mhearne/publications.html>
- Koehn, P. (2005). Europarl: A multilingual corpus for evaluation of machine translation. I *Conference proceedings: the tenth machine translation summit* (s. 79–86). Phuket, Thailand.
- Koehn, P., Och, F. & Marcu, D. (2003). Statistical phrase-based translation. I *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (s. 48–54). Morristown, NJ, USA. Tilgjengeleg frå <http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/phrase2003.pdf>
- Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. I *Proceedings of the 31st annual meeting on association for computational linguistics* (s. 17–22). Morristown, NJ, USA: Association for Computational Linguistics. Tilgjengeleg frå <http://portal.acm.org/citation.cfm?id=981577&dl=GUIDE>,
- Manning, C.D. & Schütze, H. (1999). *Foundations of statistical natural language processing*. The MIT Press. Hardcover.
- Meurer, P. (2008, March). *A Computational Grammar for Georgian*. Tilgjengeleg frå <http://maximos.aksis.uib.no/~paul/articles/Tbilisi2007-LNAI.pdf>
- Munday, J. (2001). *Introducing Translation Studies: Theories and Applications*. London: Routledge.
- Och, F.J. & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), 19–51. Tilgjengeleg frå <http://dblp.uni-trier.de/db/journals/coling/coling29.html#OchN03>

- Piao, S. & McEnery, T. (2001). Multi-word Unit Alignment in English-Chinese Parallel Corpora. I P. Rayson, A. Wilson, T. McEnery, A. Hardie & S. Khoja (red.), *Proceedings of the Corpus Linguistics 2001 Conference* (s. 466–475). Lancaster, UK. Tilgjengeleg frå http://personalpages.manchester.ac.uk/staff/scott.piao/research/papers/mwu_align4.pdf
- Prescher, D. (2004). A tutorial on the expectation-maximization algorithm including maximum-likelihood estimation and em training of probabilistic context-free grammars. *CoRR, abs/cs/0412015*, 49. Tilgjengeleg frå <http://dblp.uni-trier.de/db/journals/corr/corr0412.html#abs-cs-0412015> (informal publication)
- Pullum, G. & Scholz, B. (2001). On the Distinction between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. *Logical Aspects of Computational Linguistics: 4th International Conference, Lacl 2001, Le Croisic, France, June 27-29, 2001, Proceedings*. Tilgjengeleg frå <http://portal.acm.org/citation.cfm?id=645668.665062>
- Riezler, S. & Maxwell, J. (2006). Grammatical Machine Translation. I M. Butt, M. Dalrymple & T.H. King (red.), *Intelligent Linguistic Architecture: Variations on themes by Ronald M. Kaplan* (s. 35–52). Stanford, CA: CSLI Publications. Tilgjengeleg frå <http://www.parc.com/research/publications/details.php?id=5675>
- Rosén, V. & De Smedt, K. (2007). Theoretically Motivated Treebank Coverage. I J. Nivre, H.-J. Kaalep, K. Muischnek & M. Koit (red.), *Proceedings of the 16th nordic conference of computational linguistics nodalida-2007* (s. 152–159). Tartu: University of Tartu. Tilgjengeleg frå <http://hdl.handle.net/10062/2566>
- Rosén, V., Meurer, P. & Smedt, K. de. (2009). LFG Parsebanker: A Toolkit for Building and Searching a Treebank as a Parsed Corpus. I F.V. Eynde, A. Frank, G. van Noord & K.D. Smedt (red.), *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT7)* (s. 127–133). Utrecht: LOT. Tilgjengeleg frå <http://ling.uib.no/~desmedt/papers/tlt7rosen-submitted.pdf>
- Sag, I.A., Wasow, T. & Bender, E. (2003). *Syntactic theory: A formal introduction* (2nd utg.). Stanford: Center for the Study of Language and Information. Tilgjengeleg frå <http://csli-publications.stanford.edu/site/1575864002.html> (ISBN: 1-57586-399-5)
- Samuelsson, Y. & Volk, M. (2006). Phrase Alignment in Parallel Treebanks. I *Proceedings of Treebanks and Linguistic Theories (TLT '06)*. Prague. Tilgjengeleg frå http://ling16.ling.su.se:8080/new_PubDB/doc_repository/229_align.pdf

- Samuelsson, Y. & Volk, M. (2007). Automatic Phrase Alignment: Using Statistical N-Gram Alignment for Syntactic Phrase Alignment. I *Proceedings of Treebanks and Linguistic Theories (TLT '07)*. Bergen, Norway.
- Stodden, V. (2009). Enabling reproducible research: Licensing for scientific innovation. *International Journal of Communications Law and Policy*(13). Tilgjengeleg frå http://www.ijclp.net/issue_13.html
- Thunes, M. (2003). *Ekserpering av leksikalske oversettelsekorrespondanser fra parallelltekst*. Tilgjengeleg frå <http://www.hf.uib.no/i/LiLi/SIF/ans/Dyvik/marthaex.pdf>
- Tinsley, J., Hearne, M. & Way, A. (2007). Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. I *Proceedings of Treebanks and Linguistic Theories (TLT '07)*. Bergen, Norway.
- Unhammer, K.B. (2009). *Do arguments and adjuncts ever align? LINGMET semester assignment*. Tilgjengeleg frå <http://www.student.uib.no/~kun041/doc/argstr.pdf>
- Volk, M., Marek, T. & Samuelsson, Y. (2008). Human judgements in parallel treebank alignment. I *Proceedings of the workshop on human judgements in computational linguistics* (s. 51–57). Tilgjengeleg frå <http://www.aclweb.org/anthology-new/W/W08/W08-1208.pdf>
- XPar. (2008). *XPAR: Language diversity and parallel grammars*. (Submitted to the Research Council of Norway.)

Tillegg A

Kode for å køyre RIA-evaluering

Her står koden for å køyre analysane frå tabell 5.2. Først må du hente ut Ding-ordboka og ekstrahere testsettet. Viss du har installert Ding-ordboka til `/usr/share/dict/de-en.txt` og har lasta ned og ekstrahert RIA til `/home/brukarnamn/ria_12_06_09`, kan du klargjere ordboka og testsettet med

```
$ ./prepare-ria.sh /usr/share/dict/de-en.txt /home/brukarnamn/ria_12_06_09
```

Eventuelt kan du opne fila `prepare-ria.sh` og køyre kvar kommando linje for linje.

For å køyre testane som gav tabell 5.2 kan du laste inn `lfgalign` i Lisp-tolken din og køyre følgjande kommandoar; kommandoane står her i same følge som tabellradene:

```
(ev-ria (ria-analyses nil 0) #'random-f-align #'random-rank nil)
(ev-ria (ria-analyses nil 0) #'f-align          #'rank          nil)
(ev-ria (ria-analyses nil 0) #'f-align          #'rank          (ding-LPT))

(ev-ria (ria-analyses nil 1) #'random-f-align #'random-rank nil)
(ev-ria (ria-analyses nil 1) #'f-align          #'rank          nil)
(ev-ria (ria-analyses nil 1) #'f-align          #'rank          (ding-LPT))

(ev-ria (ria-analyses nil 2) #'random-f-align #'random-rank nil)
(ev-ria (ria-analyses nil 2) #'f-align          #'rank          nil)
(ev-ria (ria-analyses nil 2) #'f-align          #'rank          (ding-LPT))

(ev-ria (ria-analyses nil 3) #'random-f-align #'random-rank nil)
(ev-ria (ria-analyses nil 3) #'f-align          #'rank          nil)
(ev-ria (ria-analyses nil 3) #'f-align          #'rank          (ding-LPT))

(ev-ria (ria-analyses nil nil) #'random-f-align #'random-rank nil)
(ev-ria (ria-analyses nil nil) #'f-align          #'rank          nil)
(ev-ria (ria-analyses nil nil) #'f-align          #'rank          (ding-LPT))
```

(Dette går litt raskare om ein lagrar (ding-LPT) og dei ulike resultata av ria-analyses i variablar i staden for å hente dei ut på nytt kvar gong.)

For kvart kall på ev-ria får du (etter nokre blinkenlights) ei oppsummering av samanlikninga, t.d.¹

```
Intersections: 12
Unions: 70
links made by #<FUNCTION F-ALIGN>: 20
links in RIA: 62
Linkable source PRED's: 77
Link possibilities (linkable srcs * linkable trgs): 605
Unreferenced sources: 18
Unreferenced targets: 17
```

kor tala frå dei fire første linjene tilsvare kolonnene *snitt*, *union*, *lenkjer*, *RIA-lenkjer*; Linkable source PRED's er kor mange PRED-element på kjeldesida som kunne vore lenkja, Unreferenced sources/targets er kor mange «yttarste PRED» me fann.

¹Dette er frå kallet (ev-ria (ria-analyses 100 1) #'f-align #'rank (ding-lpt)); kor 100 og 1 vil seie «berre dei 100 første analysane, med maksimalt 1 yttarste PRED (utanom 0)».