

Syntaktisk informert frasesamanstilling

Kevin Brubeck Unhammer

31/08, 2010

Innhald

| | | |
|----------|---|-----------|
| 1 | Innleiing | 3 |
| 1.1 | Vegkart | 4 |
| 2 | Bakgrunn og omgrepsavklaring | 6 |
| 2.1 | SKRIV Relaterte metodar | 6 |
| 2.2 | SKRIV Eit kort oversyn over leksikalsk-funksjonell grammatikk og terminologi | 7 |
| 3 | Krav til frasesamanstilling | 9 |
| 3.1 | Innleiing | 9 |
| 3.2 | Formål med frasesamanstilling | 9 |
| 3.3 | Frasesamanstilling i ein LFG-trebank | 11 |
| 3.4 | Kva kan lenkjast? | 13 |
| 3.5 | Krav på ordnivå | 15 |
| 3.5.1 | Ordklasse | 16 |
| 3.6 | Krav på f-strukturnivå | 16 |
| 3.6.1 | Krav om lik argumentstruktur | 17 |
| 3.6.2 | SKRIV Sitere eigen korpusundersøking av variasjon i arg-str? | 19 |
| 3.6.3 | SKRIV Ulik følgje i argumentstruktur | 19 |
| 3.6.4 | SKRIV Kan adjunkt lenkjast til nodar <u>undermor</u> -lenkja? | 19 |
| | 1. Kausativar og inkorporering | 20 |
| | 2. Adposisjonsobjekt | 21 |
| 3.7 | Underordna c-strukturnodar | 22 |
| 3.7.1 | Alternativ formulering, utan å sjekke informasjonstap | 26 |
| 3.8 | Funksjonelle c-strukturnodar | 26 |
| 3.9 | SKRIV Rangering | 29 |
| 4 | Implementasjonen av lfgalign | 30 |
| 4.1 | Lenkjer mellom f-strukturar | 31 |
| 4.1.1 | Overflødige adverbial | 34 |
| 4.1.2 | SKRIV Funksjonsord utan LPT-korrespondanse bør eigentleg hindre f-strukturlenkjing | 34 |

| | | |
|----------|---|-----------|
| 4.1.3 | SKRIV Når f-lenkjene ikkje er 1-1 | 34 |
| | kausativ | 34 |
| | preposisjonsobjekt | 34 |
| 4.1.4 | Kan me gjere f-struktursamanstillinga bottom-up? | 35 |
| 4.2 | SKRIV Rangering | 36 |
| 4.2.1 | rekursivt lenkja > ulenkja, men LPT | 36 |
| 4.2.2 | argument-argument > argument-adjunkt | 36 |
| 4.2.3 | arg1-arg1 arg2-arg2 > arg1-arg2 arg2-arg1 (følge) | 36 |
| 4.2.4 | flest lenkja adjunkt | 36 |
| 4.2.5 | Prioritet på rangeringskriterium | 36 |
| 4.3 | Lenkjing av c-strukturnodar | 36 |
| 4.3.1 | SKRIV viss me har LPT, men ikkje rekursiv f-lenkje . . . | 37 |
| 5 | Avslutning | 39 |

Kapittel 1

Innleiing

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

TODO: abstract/samandrag

Omsetjingsforhold finn me mellom 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) 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 omsetje mellom lingvistiske skildringar (t.d. 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. Sjølve omsetjingsforholdet er au ein teoretisk storleik, og me kan leggje ulike kriterium til grunn for å kalle to uttrykk 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 meir teoretiske språkstudie. 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 relasjonar mellom grammatiske funksjonar, tematiske roller og kasusmarkering, ved hjelp av parallelle trebankar annotert med djupe grammatiske analysar. Samanstillinga planlagt der krev i større grad syntaktisk likskap for å kalle to uttrykk omsetjingar, men kan difor au tene på det at dei grammatiske ana-

lysane er utvikla med tanke på å vere så parallelle som mogleg. 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 frasestrukturtrea 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 krev 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.

I denne masteroppgåva spesifiserer eg kva for lenkjer mellom f-strukturar og c-strukturknodar me ønskjer, implementerer eit program `lfgalign` som automatisk finn samanstillingar med slike lenkjer, evaluerer resultatet av å køyre programmet mitt, og samanliknar dette med kva me kan få frå andre metodar.

Programmet `lfgalign` opprettar frasesamanstillingar med hjelp av f-strukturinformasjonen gitt av dei parallelle grammatikkane, og bottom-up-informasjon om kva for ordsamanstillingar som er moglege. F-strukturane avgrensar igjen kva for ordsamanstillingar som er moglege, og kva for c-strukturknodar (syntaktiske frasar) som kan lenkjast.

1.1 Vegkart

I neste kapittel ser eg på andre metodar for frasesamanstilling.

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 liste med «krav» som samanstillingane må fylle for å vere lovlege, og som implementasjonen av den automatiske frasesamanstillinga (kapittel 4) må følgje.

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

Eg evaluerer samanstillingane som kjem ut av denne metoden i kapittel ??, og samanliknar dei med det som er mogleg der me berre har konstituentstruktur (syntaktiske tre) i tillegg til ordsamanstilling.

Eg nyttar språka georgisk og norsk i evalueringa, hovudsakleg fordi dei er svært ulike syntaktisk og morfologisk; Georgisk er mellom anna eit pro-drop-språk, med friare ordfølgje og rikare morfologi enn norsk.

Sidan eg ikkje har tilgang på ferdig setningssamanstilt georgisk-norsk parallelltekst, blir det vanskeleg å køyre den statistiske ordsamanstillinga som er vanleg som første steg i N-grambaserte metodar (utan ein god del forarbeid). Difor konsentrerer eg meg i evalueringa om eit testkorpus kor eg manuelt gjer ordsamanstillinga. Eg veit heller ikkje enno om nokon statistisk parsar av høg kvalitet for georgisk, men testkorpuset er ferdig parsa med LFG-parsaren frå Meurer (2008), c-strukturnodane avgrensar då kva som er ein syntaktisk konstituent.

fortelje om
georgisk i
kap.5 heller

Kapittel 2

Bakgrunn og omgrepsavklaring

2.1 SKRIV Relaterte metodar

Automatisk frasesamanstilling er eit nytt felt. Det finst allereie veldig gode system for automatisk setningssamanstilling, og automatisk samanstilling av ord har komme langt, men nivåa mellom ord og setning ser ut til å by på fleire problem. «by på fleire problem» – weasel wording, TODO omskriv. Dei ulike tilnærmingane som finst er prega av formåla til utviklarane. Det er verdt å merkje seg at ordet «frase» ofte blir nytta i litteraturen om strenger av ord (N-gram) som ikkje treng vere syntaktiske konstituentar, igjen avhengig av formålet med metoden.

Innanfor korpuslingvistikken har t.d. Piao & McEnery (2001) nytta enkel kollokasjonsinformasjon for å først finne sannsynlege nominale frasar på engelsk og kinesisk (dvs. «chunking»), og så samanstillе desse; her er evalueringsgrunnlaget rett og slett ein manuell gjennomgang av dei mest sannsynlege omsetjingane dei får.

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

fleire slike?
meir om dette,
algoritmen

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 større grad om ein nyttar dependensinformasjon (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.

Så langt har eg ikkje komme over metodar som går i motsett retning, altså prøver å finne eller betre på frase- og ordsamanstilling ut 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).

Dette er motsett retning av det mitt program gjer, nemne seinare?

2.2 SKRIV Eit kort oversyn over leksikalsk-funksjonell grammatikk og terminologi

I dei følgjande kapitla nyttar eg ein del LFG-terminologi (i tillegg til eit par eigne termar). Difor gir eg her eit kort oversyn over det som kan vere nytt for dei som er meir vand med andre grammatiske rammeverk.

modellteoretisk (vs derivasjonelt) LFG er eit modellteoretisk, ikkje-derivasjonelt, rammeverk for grammatikk. Pullum & Scholz (2001) gir ein god gjennomgang av forskjellen mellom derivasjonelle (enumerative) grammatikkar og modellteoretiske grammatikkar, kor førstnemnde definerer *mengder av ut-*

trykk ved avleiing frå startsymbol, medan sistnemnde gir skildringar av *enkeltuttrykk*. 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 skildringa 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; ein 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.

f-struktur ...

c-struktur ...

endosentrisitetsprinsippa ...

\bar{X} ...

diskontinuerlege konstituentar ...

ϕ c-struktur-f-strukturavbildinga ϕ ...

ϕ^{-1} Det funksjonelle domenet til ein f-struktur er gitt ved ϕ^{-1} , inversen av c-til-f-strukturavbildinga, og tilsvare dei nodane i c-strukturen som projiserer denne f-strukturen, t.d. ein VP-node med dominerande IP og CP (Bresnan, 2001, s. 126). Sidan dette er inversen av ein funksjon, kan me ha diskontinuerlege konstituentar i same funksjonelle domene (på same måte som ulike argument til ein funksjon kan gi same verdi).

fraselenkjer vs frasesamanstilling Eg nyttar her termene *lenkjing* og *samanstilling* i omtrent same tyding som dei engelske termene *link* og *alignment*, kor 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 altså er meir ateoretisk / datanært.

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 å finne 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

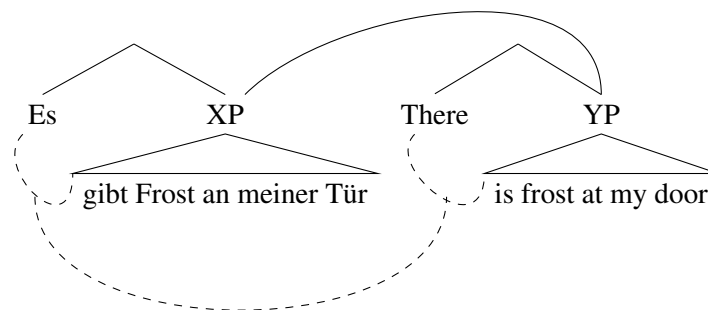
Ei frasesamanstilling er ein slag annotasjon av eit korpus. På same måte som oppbygginga av eit korpus avheng av formålet til korpuset, kan ein ikkje definere den ideelle annotasjonen av eit korpus utan å ta høgd for kva ein skal nytte annotasjonen til.

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 nyttige.

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 fraselenkje mellom N-grammet *es gibt* og *there is* (dvs. eit auka sannsyn for å nytte slike par i omsetjinga) kan gi ein høgare endeleg skåre i BLEU. Som vist i Koehn et al. (2003) fekk dei ein lågare BLEU-skåre når dei fjerna lenkjer mellom nodar som, i følgje ein robust statistisk PCFG-parsar, 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 omsetjingssteget¹ er det for dei ingen grunn til å berre halde seg til samanstilling mellom syntaktiske konstituentar; dei har i utgangspunktet meir nytte av kollokasjonsinformasjon.

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



Figur 3.1: N-gram-samanstilling versus syntaktiske frasar

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 semantiske² omsetjingsskifte, t.d.

¹Både 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).

²Sidan 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).

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 sjølvsagt ikkje lenkje på tvers av konstituentgrenser, sidan det er fullstendige konstituentar³ som fyller dei semantiske rollene.

Eit anna mogleg formål er å nytte desse frasesamanstillingane til maskinomsetjing. Riezler & Maxwell (2006) nyttar ein stokastisk frasesamanstilling til å oppdage transfer-reglar 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 transfer-reglar hindrar at ein får reglar som lenkjar ikkje-konstituentar, eg kjem tilbake til dette nedanfor.) Samanstillinga utvikla her burde au kunne nyttast til å finne slike transfer-reglar, men dette er ikkje noko eg har lagt vekt på.

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

3.3 Frasesamanstilling i ein LFG-trebank

Samanstilte frasar bør ha nok semantisk likskap til å kunne opptre som omsetjingar i liknande omgivnader (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 ordsamanstilling⁴, 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

³LFG tillèt som nemnt diskontinuerlege konstituentar, men dette er ikkje det same som ikkje-konstituentar av typen «es gibt» / «there is».

⁴(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.

må *tildele 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) avleidd 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 blir nytta i LFG⁵ (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 sjølvstøtt 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.

⁵I 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).

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 tre ulike nivå kor me kan finne ekvivalensforhold under setningsnivå:

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

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-strukturknodar impliserer at heile frasen under er lenkja. Det finst ingen c-strukturknoda som dominerer berre *there*, *is* og ingen andre ord (heller ikkje *es*, *gibt*), så dette er ikkje lenjekandidatar. *There is* og *Es gibt* i figur 3.1 kan då ikkje samanstillast aleine, men berre som del av ei ytre frasesamanstilling.

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* 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 predikerande ord, kan me – gitt to samanstilte setningar – la *kandidatane for samanstilling på f-strukturnivå* inkludere alle desse PRED-elementa i f-strukturane til setningane⁶. 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. **Diskutabelt, TODO:** PRED-elementa er i tillegg gjerne enklare å knyte direkte opp mot den konkrete, observerte tekststrengen, medan t.d. aspekt kanskje er umogleg å skilje frå tempus i affikset.

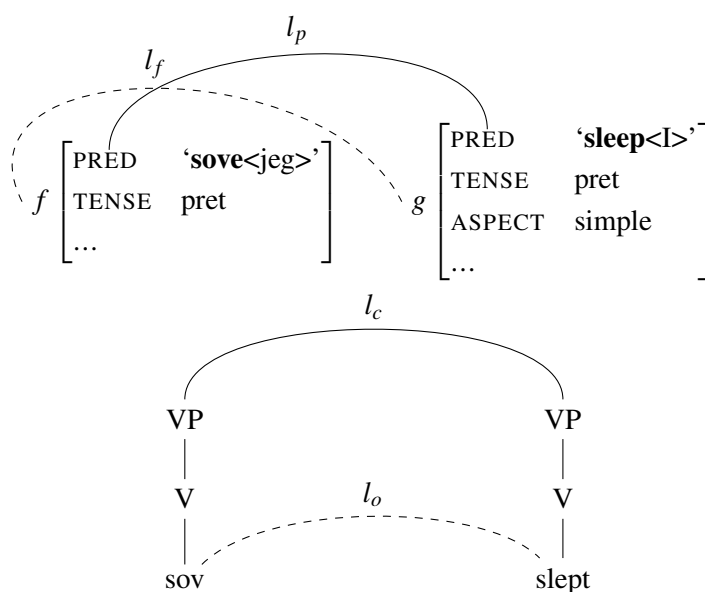
Samtidig er det au eit omsetjingsforhold mellom trekka i same f-struktur som dei lenkja PRED-elementa, og me ville kanskje ikkje ha omsett dei to PRED-elementa i andre f-strukturkontekstar. Difor bør me au sjå på ei PRED-lenkje som ei lenkje mellom *f-strukturane til desse PRED-elementa*⁷. Med dette i tankane,

⁶I del 3.8 kjem eg tilbake til spørsmålet om me vil inkludere visse f-strukturar utan PRED-element i kandidatane for samanstilling.

⁷Eventuelt kunne me ha definert lenkjingskandidatane på f-strukturnivå som alle PRED-haldande f-strukturar, resultatet blir det same.

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. iallfall dei øvste nodane i $\phi^{-1}(f)$ og $\phi^{-1}(g)$, dei funksjonelle domena til f-strukturane f og g , er lenkja



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).

Punkt (1-a) og (1-c) over seier at viss PRED-elementa projisert av t.d. to verb i verbfrasar er lenkja, vil VP-ane som heilskap vere lenkja (både VP-nodane som dominerer dei lenkja funksjonelle domena, og f-strukturane frå ytre PRED til verba), det er dette at heile VP-ane 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 øvste node dominerer alle nodar i det funksjonelle domenet; det er det funksjonelle domenet som spesifiserer informasjonen i f-strukturane, toppnodane dominerer heile dette domenet og bør difor lenkjast viss og berre viss f-strukturane er lenkja.

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 ikkje-øvste nodar i det funksjonelle domenet. I tillegg finst det nodar over ord som ikkje projiserer PRED-element, desse kjem eg tilbake til i del 3.8.

I følgje 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
 ↔
 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 ordnivå (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 passande 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 omsetjingar⁸. Dyvik et al. (2009, s. 74) krev dei at orda generelt, utan kontekst, må vere semantisk plausible omsetjingar, eller at målordet er eit medlem av mengda av *linguistically predictable translations* av kjeldeordet. Målordet har då *LPT-korrespondanse* med kjeldeordet. Informasjon

⁸Eigentleg 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.

om slik LPT-korrespondanse kan komme frå ei djup semantisk dekomponering av kvart ord, då blir det eit krav på f-strukturnivå (eller på ein semantisk struktur), eller han kan komme bottom-up, typisk frå automatisk ordsamanstilling. Bottom-up-informasjonen viser då om orda generelt (i ulike kontekstar) blir nytta som omsetjingar av kvarandre. Nedanfor reknar eg LPT-kravet som eit krav på ordnivå.

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).

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

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

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

Viss setningane i døme (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 me ha slike lenkjer der eitt ord projiserer fleire PRED-element⁹.

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 omsetjing¹⁰.

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,

problematiser: må me ha /PRED-lenkje/ frå arg til arg/adj? (ikkje krevd no...men skal me rangere ved det?) kryssande f-lenkjer? mange-mange-f-lenkjer?

⁹Me 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.

¹⁰Munday (Catford (1965), i 2001, s. 61) gir ein gjennomgang av slike *klasseskifte*, og andre typar omsetjingsskifte.

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 passande 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-strukturane. 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.

3.6.1 Krav om lik argumentstruktur

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

Om det alltid er slik at to predikat har like mange argument, som kjem i same rekkjefølgje i argumentstrukturen, vil det gjere den praktiske oppgåva med å lenkje predikata, og argument med argument, mykje enklare. Men kan me stille så sterke krav?

Sett at ein 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.

Slike omsetjingsrelasjonar gir data for verbsituasjonen, på eit meir generelt grunnlag enn det me kan få frå einspråklege analysar åleine. Om me har gode semantiske grunnar for å kalle ein deltakar i ein verbsituasjon eit argument på eitt språk, vil dei same grunnane gjelde for omsetjingsmessig korresponderande verb på andre språk. Ein kan då nytte unionen over alle argument til korresponderande verb til å karakterisere kva ein meiner med *deltakarane i verbsituasjonen*. Syntaktiske forhold i språket kan sjølvstendig gi grunnar til å *ikkje* kalle dette eit argument.

For å gjere dette konkret kan me sjå på setning 7 i test-suiten til XPar-prosjektet:

- (4) abramsi brouns daenajleva sigaretze, rom cvimda
 Abrams.NOM Brown.DAT vedde.3SG sigarett.om, at regne.3SG.IMP
 ‘Abrams veddet en sigarett med Brown på at det regnet’

I følgje LFG-parsen til desse setningane har hovudpredikata svært ulike argument-

struktur¹¹. Det norske *vedde* har fire argument, medan *da-najleveba* 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 tre argument, *with* blir her adjunkt, medan den tyske grammatikken, som au har tre 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:

- (5) a. Adams veddet en sigarett med Browne (norsk bokmål)
på at det regnet.

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

- b. abramsi brouns daenajleva sigaretze, rom cvimda. (georgisk)

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

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

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

- 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 tre 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.

Det same gjeld forøvrig lenkjing av adjunkt til adjunkt. Adjunkt plukker ut si eiga rolle der argument får rolla tildelt frå verbet, og f-strukturane har ingen hierarkisk inndeling av desse slik me har for verb og argument, dei er i staden representert som uordna mengder.

¹¹Analysane 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).

3.6.2 SKRIV Sitere eigen korpusundersøking av variasjon i arg-str?

Ei undersøking av den frasesamanstilte trebanken SMULTRON (Samuelsson & Volk, 2006) mot LFG-grammatikkane for engelsk og tysk fann at 2 av 15 korresponderande verbtoken¹² for høgfrekvente innhaldsverb fekk analysar kor argument korresponderte med adjunkt (?).

LCS, dorr

3.6.3 SKRIV 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 opplevar som objekt og tema som subjekt vil ha opplevar nedanfor tema i argumentstrukturen, medan ei omsetjing av dette verbet kan ha tema nedanfor:

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

Argumentstrukturane i (6) har omvendt intern følgje, og som vist ved dette dømet er det heller ikkje noko f-strukturinformasjon me kunne nytta til å sikre lenkjinga *sie/dem* og *ihnen/de*. Igjen ser det ut til at bottom-up-informasjon trengst.

3.6.4 SKRIV Kan adjunkt lenkjast til nodar under mor-lenkja?

Krav (vi) i Dyvik et al. (2009, s. 75) krev at viss F_s og F_t er lenkja, så kan ingen adjunkt D_s til F_s vere lenkja til nodar utanfor F_t . Men kan ein D_s lenkjast til ei dotternode av argument eller adjunkt til F_t ?

R_t er dotter til F_t , og må då vere lenkja til ei dotter av F_s , A_s . Då må au alle argument til R_t vere lenkja til døtre av A_s , så D_s kan ikkje lenkjast til argument av dotternodar til F_t . Kva med adjunkt? Om me finn eit ulenkja adjunkt til R_t kan me heller ikkje lenkje dette til D_s ved krav (vi) igjen, sidan D_s står utanfor A_s .

Men om D_t er ei ulenkja *adjunktdotter* av F_t , så vil døtre av D_t kunne lenkjast til D_s , så lenge D_t forblir ulenkja. Me kan altså sjå ned i adjunktdøtre av F_t for å lenkje D_s .

¹²25 om ein inkluderer analysar kor minst eitt av argumenta ikkje hadde korrekt analyse (t.d. eit PRO der grammatikken burde funne eit substantiv).

På same måte bør ein kunne rekursivt sjå ned i ulenkja adjunktdøtre av R_t , men ein bør kanskje ikkje kunne lenkje så djupt uansett? Ikkje automatisk, uansett.

Programmet mitt vil, gitt to initielle f-strukturar med LPT-korrespondanse, finne alle moglege kombinasjonar av lenkjer som inneheld alle argument og kanskje adjunkt, dvs. om me har

F_s [PRED p<1,2> ADJUNCT { 3 }]

F_t [PRED p<4> ADJUNCT { 5,6 }]

vil dette vere logisk moglege samanstillingar av «f-strukturdøtre»:

(((1 . 4) (2 . 5)) ((1 . 4) (2 . 6)) ((1 . 5) (2 . 4))
((1 . 5) (2 . 6) (3 . 4)) ((1 . 6) (2 . 4)) ((1 . 6) (2 . 5) (3 . 4))) ■

Me luker ut kombinasjonar som bryt med LPT-korrespondanse. Med full informasjon bør me sjølvstøtt berre ende opp med éin kombinasjon, t.d. ((1 . 4) (2 . 5)).

Så langt bør altså krav (i-iv) frå Dyvik et al. (2009) vere dekkja.

Me kan krevje at f strukturane-til f strukturdøtre-kan lenkjast rekursivt for at F_s og F_t skal lenkjast, t.d. både (1 . 4) og (2 . 5). Men her kjem det (iallfall) to problem.

1. Kausativar og inkorporering

Om me har

F_s [PRED p<SUBJ, 1, 2> XCOMP 2[PRED q<1>]]

F_t [PRED pq<SUBJ,OBJ>]

kor pq er t.d. ein kausativ som tilsvarende p<..., q>, så vil me ikkje kunne lenkje F_s og F_t sidan det bryt med krav (iii), F_s har eit argument for mykje. Men her vil det kanskje vere naturleg å ha ei ein-mange-lenkje:

((F_s 2) . F_t)

No kan me sjå på unionen av argument av F_s (minus XCOMP) og argument av XCOMP, alle argument i denne unionen må då ha LPT-korrespondanse med argument/adjunkt av F_t , og alle argument av F_t må ha LPT-korrespondanse med argument/adjunkt av unionen.

Det same bør kanskje skje ved vanleg inkorporering av substantiv, då må det altså vere mogleg å føye saman t.d. verb og objekt; ein kombinasjon av dette og kausativ bør vel vere mogleg, t.d.

F_s [PRED la<SUBJ, 1> XCOMP 2[få<1, 3:pengar>]]

F_t [PRED belønn<SUBJ, 1>]

Igjen ser me på argument frå unionen av (F_s 2 3) minus 2 og 3, og om det er mogleg å lenkje dei til argument/adjunkt av F_t , og omvendt.

Men det bør kanskje vere grenser for kor langt samanføyning kan gå... eg kan ikkje tenkje meg at me vil lenkje ($(F_s$ 2) . F_t) eller ($(F_s$ 1 2) . F_t) her:

F_s [PRED p<..., 1> XCOMP 1[PRED q<..., 2> XCOMP 2[PRED r<...>]]] ■

F_t [PRED pr<...>]

... men det kan jo hende det finst situasjonar der dette au vil vere rett. Problemet er altså kor me skal setje grensene i implementasjonen. Om me skal prøve å samanføye på alle moglege måtar (altså, der me ikkje har informasjon om LPT), i tillegg til «vanlege» lenkjer, blir det fort komputasjonelt vanskeleg. Me kan sjølvstøtt snu på LPT-kravet her, og seie at dette er berre lov der me har positiv informasjon om LPT-korrespondanse, i staden for at det ikkje er lov om me har motstridande LPT-informasjon, det vil nok hjelpe, men det er vanskeleg å finne prinsippelle avgrensingar her.

2. Adposisjonsobjekt

I følgjande setningspar har me eit objekt «sigarett» som svarer til PP-en «sigaretze» («sigareti» + «ze»), eit adjunkt:

Abrams veddet en sigarett med Browne på at det regnet.
 abramsi brouns daenajleva sigaretze, rom cvimda.

F_s [PRED sigarett]

F_t [PRED ze<1> 1[PRED sigareti]]

F_s og F_t er døtre av dei ytre predikata i kvar setning, krav (iii) seier at det må vere LPT-korrespondanse mellom desse for at me skal kunne lenkje «veddet» og «daenajleva». Her synest det feil å føye saman «sigareti» og «ze», (F_s . (F_t 1)), sidan «sigarett» ikkje inneheld informasjonen gitt av «ze».

Det finst då to løysingar. Me kan slakke på LMT-kravet ved å la $L'(F_t) = \{\text{sigaretze}, \text{ze}\}$ (evt. $\{\text{sigaret}, \text{ze}\}$), då kan me lenkje (F_s . F_t), medan 1 er ulenkja.

Eller me kan lenkje (F_s . 1), kor me har skikkeleg LMT-korrespondanse, men då må me slakke på (iii) og (iv), og altså ha lov til å «hoppe over» ein f-struktur for å lenkje «veddet» og «daenajleva». F_t er då ulenkja. Det er løysinga valt i Dyvik et al. (2009, s. 75, fotnote 3), og den løysinga eg følgjer vidare i oppgåva.

3.7 Underordna c-strukturnodar

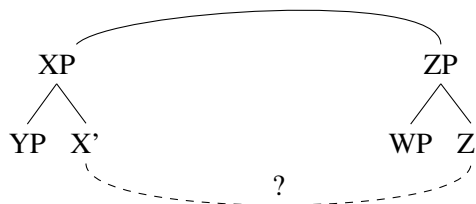
Toppnodane i to funksjonelle domene som er lenkja på f-strukturnivå vil ha ein informasjonsmessig korrespondanse, og kan, som nemnt i del 3.4, utan vidare lenkjast. Men det er mogleg å lenkje desse toppnodane, t.d. XP på kjeldespråket og ZP på målspråket, utan at nodane under (X' , Z') er lenkja til noko. I denne delen går eg gjennom kva som må til for å lenkje slike underordna (ikkje-øyste) c-strukturnodar.

Me kan byrje med eit par enkle krav, me seinare utvidar:

- (7) Enkle krav for lenkjing av underordna c-strukturnodar:
- c-strukturnodar som ligg under øvste node i to funksjonelle domena kan berre samanstillast med nodar som ligg innanfor desse domena,
 - c-strukturnodar kan berre samanstillast om deira funksjonelle domene er lenkja på f-strukturnivå

(7-a) seier at om XP og ZP er lenkja, der XP og ZP er toppnodar, kan ikkje dotter til XP, X' , lenkjast med t.d. mor eller søster til ZP. Og om ZP dominerer ein node WP, der $\phi(WP) \neq \phi(ZP)$, kan me heller ikkje ha ei c-strukturlenkje frå X' til WP. Grunnen er at f-strukturane er sett saman av informasjonsinnhaldet projisert av nodane, så det vil vere unaturleg å ha ei c-strukturlenkje som står i konflikt med f-strukturlenkjer.

Men sjølv der to underordna c-strukturnodar er i lenkja funksjonelle domene (altså, der dei respektive toppnodane er lenkja) er det ikkje alltid slik at me vil lenkje dei underordna nodane. I figur 3.3 er XP og ZP to lenkja toppnodar i som inkluderer høvesvis X' og Z' i dei funksjonelle domena. Her kan me ha gode grunnar for å ikkje lenkje X' og Z' .



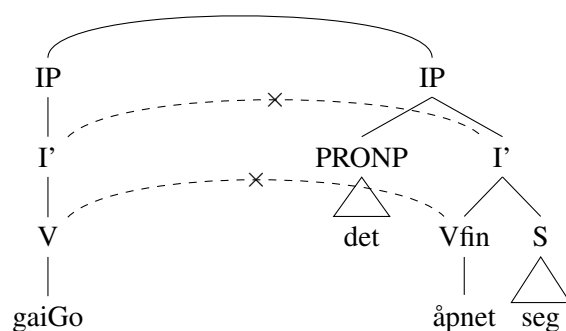
Figur 3.3: Lenkjing av underordna c-strukturnodar

Éin grunn til å ikkje lenkje desse underordna nodane, er viss YP og WP i figur 3.3 ikkje er lenkja, og det finst informasjon som korresponderer mellom X' og WP eller mellom Z' og YP; dette gir eit mangel på samsvar mellom X' og Z' . Ein annan situasjon der me ikkje vil lenkje X' og Z' er når WP ikkje finst og YP har informasjon som på målspråket blir spesifisert av nodar under Z' .

Når treet deler seg i to som i figur 3.3, 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

begge situasjonane nemnt i avsnittet ovanfor er det slik at X' og Z' gir ulike tilskot til f -strukturen, dei kan difor ikkje lenkjast. Likevel må me tillate litt slingringsmonn her, X' og Z' skal ikkje trenge projisere heilt like f -strukturar. Det som er relevant er det som blir lenkja i f -strukturen, eller *kva for lenkjer som finst mellom f -strukturane til nodane under*.

For å gjere dette meir konkret kan me sjå på dømet i figur 3.4. IP-nodane er her lenkja, men me kan ikkje lenkje I' -nodane i same funksjonelle domene. PRONP-noden, spesifikator på den norske sida, er ikkje lenkja med nokon spesifikator på den georgiske sida. Den informasjonen (her reint syntaktisk) som ordet *det* tilfører IP, ligg under I' på georgisk. Me kunne lenkja I' om me hadde ein georgisk spesifikator som var lenkja til den norske PRONP. Me kan heller ikkje lenkje V_{fin} til V , her manglar endå meir informasjon (både frå *det* og frå *seg*).



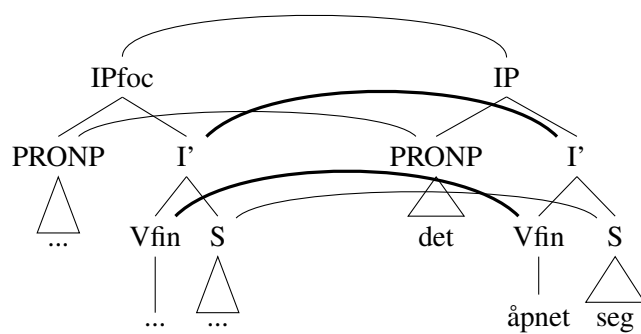
TODO: teikne inn f -domene her òg

Figur 3.4: Umogleg lenkjing av underordna c -strukturnodar mellom georgisk og bokmål

Hadde det georgiske treet hatt spesifikator og komplement som kunne lenkjast til spesifikator og komplement på norsk, kunne me ha lenkja I' og V_{fin} . For å tillate desse lenkjene, men ikkje dei i figur 3.4, ville det vore nok å krevje at søsternodane var lenkja. Hadde me hatt situasjon i figur 3.5 nedanfor ville dette stemt, V_{fin} -nodane kan lenkjast fordi S -nodane er lenkja, I' -nodane fordi PRONP-nodane er lenkja.

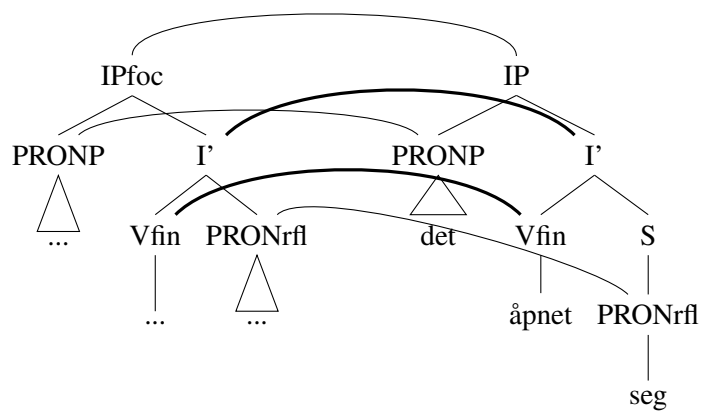
Men det blir for strengt å krevje at søsternodar er lenkja; S -noden i den norske analysen er ein del av det funksjonelle domenet til IP, medan komplementet i andre språk kanskje går rett på eit nytt funksjonelt domene. Figur 3.6 demonstrerer denne situasjonen. Her kan ikkje S lenkjast til PRONP sidan dei ikkje er i same funksjonelle domene, men me vil jo likevel lenkje V_{fin} -nodane.

Me treng altså eit litt meir nyansert krav. 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. For at me skal unngå å lenkje



TODO: teikne inn f-domene her òg

Figur 3.5: Ei enkel lenkjing av c-strukturnodar



TODO: teikne inn f-domene her òg

Figur 3.6: Her vil me lenkje Vfin-nodane utan å lenkje søstrene deira.

I' og $Vfin$ i figur 3.4 må altså *det* og *seg* òg vere ordlenkja til *gaiGo*. Dette blir ei ein-til-mange-lenkje på ordnivå, som må representerast som fleire lenkjer som alle byrjar i *gaiGo*. Viss georgisk er kjeldespråket (n_s , norsk: n_t) blir $LL(IP_s) = LL(I'_s) = LL(V_s) = \{(det, gaiGo), (åpnet, gaiGo), (seg, gaiGo)\} = LL(IP_t)$. Mengdene $LL(I'_t) = \{(åpnet, gaiGo), (det, gaiGo)\}$ og $LL(Vfin_t) = \{(åpnet, gaiGo)\}$ på den norske sida har ikkje korresponderande mengder på georgisk og blir ikkje lenkja¹³.

Men viss me vil unngå å referere til ordlenkjer, går det au an å definere kravet i form av f-strukturlenkjer på preterminale nodar¹⁴:

- (8) For å lenkje dei underordna c-strukturnodane n_s og n_t (kor toppnodane i dei funksjonelle domena er lenkja ved krav (1-c)):
 La $L_c(n_s)$ vere alle f-strukturlenkjer frå $\phi(n'_s)$ for alle preterminale n'_s som er dominert av n_s . Ulenkja nodar n'_s er ikkje med i L_c . Nodane n_s og n_t kan då lenkjast viss
- mor av n_s , m_s , har same funksjonelle domene som n_s , og mor av n_t , m_t , har same funksjonelle domene som n_t ,
 - m_s og m_t er lenkja,
 - og $L_c(m_s) - L_c(n_s) = L_c(m_t) - L_c(n_t)$.

(8-a og -b) seier til saman det same som (7-a) og (7-b). Det (8-c) seier er at me må ha same *informasjonstap* når me går nedover i c-strukturreet.

Figur 3.6 illustrerer dette kravet. IP_{foc} og IP er toppnodar, lenkja ved krav (1-c). I' -nodane er i lenkja funksjonelle domene og har lenkja mødre. I tillegg har dei same informasjonstap:

$$\begin{aligned} L_c(IP_{foc_s}) &= \{(\mathbf{PanJara}, \mathbf{den}), (\mathbf{ga-Geba}, \mathbf{åpne})\}, L_c(I'_s) = \{(\mathbf{ga-Geba}, \mathbf{åpne})\} \\ L_c(IP_t) &= \{(\mathbf{PanJara}, \mathbf{den}), (\mathbf{ga-Geba}, \mathbf{åpne}), (\mathbf{pro}, \mathbf{pro})\}, L_c(I'_t) = \{(\mathbf{ga-Geba}, \mathbf{åpne}), (\mathbf{pro}, \mathbf{pro})\} \\ L_c(IP_{foc_s}) - L_c(I'_s) &= \{(\mathbf{PanJara}, \mathbf{den})\} = L_c(IP_t) - L_c(I'_t) \end{aligned}$$

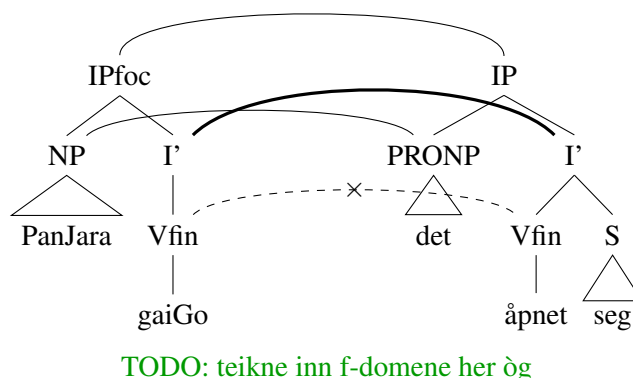
$Vfin$ -nodane er au i same funksjonelle domene, og har lenkja mødre, men her har dei ulikt informasjonstap:

$$\begin{aligned} L_c(I'_s) &= \{(\mathbf{ga-Geba}, \mathbf{åpne})\}, L_c(Vfin_s) = \{(\mathbf{ga-Geba}, \mathbf{åpne})\} \\ L_c(I'_t) &= \{(\mathbf{ga-Geba}, \mathbf{åpne}), (\mathbf{pro}, \mathbf{pro})\}, L_c(Vfin_t) = \{(\mathbf{ga-Geba}, \mathbf{åpne})\} \\ L_c(I'_s) - L_c(Vfin_s) &= \{\} \\ L_c(I'_t) - L_c(Vfin_t) &= \{(\mathbf{pro}, \mathbf{pro})\} \end{aligned}$$

Me ikkje har altså ein c-strukturnode som direkte projiserer f-strukturen til *pro*

¹³Eg har ikkje gjort noko forskjell på nodar som dominerer like mengder på same språk, dette kan tolkast som at alle dei ikkje-terminale nodane på georgisk er lenkja til IP på norsk.

¹⁴Då kan me au representere mange-til-mange-ordlenkjer som «udelelege», $(\{gaiGo\}, \{det, åpnet, seg\})$ blir den einaste ordlenkja i dømet over, sidan me ikkje må samanlikne ordlenkjene frå IP_t , I'_t og $Vfin_t$.



Figur 3.7: Delvis mogleg lenkjing av underordna c-strukturknodar mellom georgisk og bokmål

på georgisk, lenkja til *seg* på norsk; f-strukturen til *pro* er spesifisert via verbet. Men sjølv om me ikkje kan sjå ein slik node, kan me sjå at det er *noko* som er spesifisert via dette fragmentet av den georgiske c-strukturen, og at det er lenkja til *seg*, difor mister me informasjon når me går frå *I'* til *Vfin*.

3.7.1 Alternativ formulering, utan å sjekke informasjonstap

F-strukturen til *gaiGo* i dømet over har *pro*-elementet (lenkja til *seg*) som argument, difor burde det au vere mogleg å finne f-strukturlenkja til dette argumentet når ein finn $L_c(I'_s)$. Om ein legg til denne lenkja i L_c , er det nok at $L_c(n_s) = L_c(n_t)$ for at n_s og n_t skal lenkjast (dvs. ein treng ikkje sjekke forskjellen til mornodane). Men skal ein formulere eit slikt krav må ein passe på å spesifisere kor djupt ned i f-strukturen ein skal kikke, og avgjere om adjunkt-lenkjer skal med, osv. I implementasjonen min har eg valt å følge krav (8) for å unngå å måtte kode denne typen informasjon inn i systemet.

3.8 Funksjonelle c-strukturknodar

Ikkje alle ord tilsvarer PRED-element i f-strukturen, dette gjeld typisk funksjonsord (t.d. *som*, *at*). 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 metoden frå del 3.7 i dette tilfellet er at knodar over funksjonsord er i det same funksjonelle domenet som komplementet, og knodane 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-strukturtreet i buntar med likt informasjonstap.

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 ikkje ein gong lenkje komplementa¹⁵. 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 element som ikkje er PRED i denne oppgåva).

Me krev 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.

Då får me følgjande:

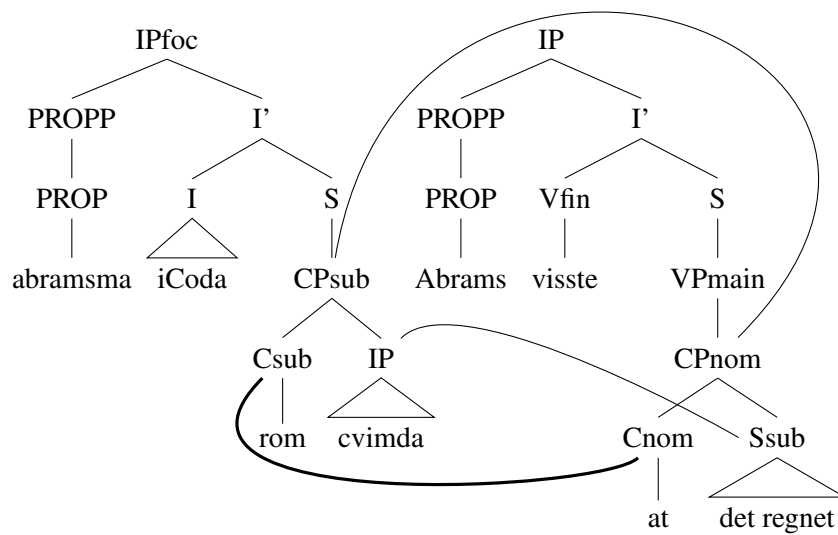
- (9) Krav for lenkjing av funksjonelle kategoriar i c-strukturen:
- a. Gitt ei mogleg lenkjing av FP og GP, kor F og G er funksjonelle kategoriar der komplementa elles kan lenkjast, introduserer me eit «falskt informasjonstap» mellom FP og F' og mellom GP og G'; orda under F' og G' må ha LPT-korrespondanse 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 noko forskjell i informasjonstap mellom FP og F', er F' medlem av nodemengden som blir lenkja til XP.

Om (9-a) er oppfylt, kan me få samanstillinga vist i figur 3.8. Her vil dei funksjonelle domena til CPsub og CPnom kvar kunne delast opp i to deler, kor den funksjonelle delen har LPT-korrespondanse medan komplementa er lenkja på f-strukturnivå. Det er ingen informasjonstap frå CPsub til Csub som ikkje er reflektert i CPnom til Cnom, og det er ingen informasjonstap frå CPsub til IP som ikkje er reflektert i CPnom til Ssub.

(Alle nodane under S vist i dei to trea er i same funksjonelle domene, så om dei funksjonelle domena er lenkja, vil krav~(7-a) og krav~(7-b) vere oppfylt kva gjeld CP-komplementa – lenkjinga går ikkje ut over dei funksjonelle domena, medan krav~(8) er dekkja for IP og Ssub med unntaket over.)

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 oppgå-

¹⁵Skal 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 vere samanstilt via toppnodane (t.d. CP).



Figur 3.8: Mogleg samanstilling av funksjonelle c-strukturnodar mellom georgisk og norsk (bokmål)

va; 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-strukturlenkjinga avhengig av c-strukturforhold, og i implementasjonen min har eg difor lagt mindre vekt lenkjing av funksjonelle kategoriar.

3.9 SKRIV Ranging

(meir om dette i del 4.2)

Kapittel 4

Implementasjonen av `lfgalign`

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 Lisp¹-program.

Ei implementering gjer det svært synleg om det finst manglar i eit formelt krav, eller om noko ikkje er godt nok spesifisert.

Programmet `lfgalign`² 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 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 rangering). Nodane i kvar av desse mange-til-mange-lenkjene definerer no den endelege frasesamanstillinga.

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

²Tilgjengeleg frå <http://github.com/unhammer/lfgalign> som fri og open programvare under GNU General Public License.

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.

intro TODO,
kanskje noko
om kva eg
faktisk har fått
ut av imple-
mentasjonen

treng eg ein
eigen del om
LPT i dette
kapittelet?
Implementa-
sjonen er jo
veldig enkel
iallfall.

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.

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 (iii) og (iv) i Dyvik et al. (2009, s. 75). Elles er krav (i) er tautologisk oppfylt, medan me som nemnt føreset at krav (ii) er oppfylt før alle kall på *f-align*.

TOGROK:
kan me i tillegg sjekke funksjonelle nodar i same f-domene som PRED-ordformen har LPT-korrespondanse? ■

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 $\{(SUBJ, ADJ), (OBJ, SUBJ)\}$. 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 *f-align* går så gjennom kvar lenkje i kvar argumentpermutasjon, og 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, me unngår dobbeltarbeid ved å lagre alle delvise samanstillingar i tabellen *alignable*. Dette føreset at *f-align(s, t)* er uavhengig av konteksten rundt; t.d. må mengda av samanstillingar som kjem ved å lenkje subjektet til F_s mot subjektet

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 kan vere nyttig for å rangere samanstillingar, og *f-align* blir mykje meir oversiktleg av å jobbe med eit slikt tre. Ein funksjon *flatten* omformar 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.

til F_t vere uavhengig av om objektet til F_s er lenkja mot eit objekt eller eit adjunkt osb. av F_t .

```

alignments  $\leftarrow \emptyset$  ;
forall the argperm in argalign( $F_s$ ,  $F_t$ ) do
     $p \leftarrow \emptyset$  ;
    forall the  $A_s$ ,  $A_t$  in argperm do
        if not(aligntable[ $A_s$ ,  $A_t$ ]) then
            | aligntable[ $A_s$ ,  $A_t$ ]  $\leftarrow$  f-align( $A_s$ ,  $A_t$ );
        if aligntable[ $A_s$ ,  $A_t$ ] then
            | add aligntable[ $A_s$ ,  $A_t$ ] to  $p$ ;
        else
            | add ( $A_s$ ,  $A_t$ ) to  $p$ 
    end
    add  $p$  to alignments ;
    forall the adiperm in adajalign(argperm,  $F_s$ ,  $F_t$ ) do
         $a \leftarrow$  copy-of( $p$ ) ; // optional adjunct links
        forall the  $A_s$ ,  $A_t$  in adiperm do
            if not(aligntable[ $A_s$ ,  $A_t$ ]) then
                | aligntable[ $A_s$ ,  $A_t$ ]  $\leftarrow$  f-align( $A_s$ ,  $A_t$ );
            if aligntable[ $A_s$ ,  $A_t$ ] then
                | add aligntable[ $A_s$ ,  $A_t$ ] to  $a$ ;
            else
                | add ( $A_s$ ,  $A_t$ ) to  $a$ 
            end
        end
        add  $a$  to alignments ;
    end
end
// loop through adajalign if no arguments exist
if alignments =  $\emptyset$  then return  $\emptyset$  ; // Fail
else return (( $F_s$ ,  $F_t$ ), alignments) ;

```

Funksjon 1: f-align(F_s , F_t)

TODO:
nemne
føresetnaden
om
uavhengnad i
kapittel 3

Sjølvs 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 *aligntable* for å finne ut av om kvar av f-strukturane kunne lenkjast; i dette tilfellet vil *aligntable*[OBJ, ADJ] vere tom.

Om me i tillegg krev at substrukturar kan samanstillast kan me utelukke len-

TODO:
forskjellen
mellom
LPT-krav og
rekursjons-
krav på
argument må
inn i kapittel 3

usage: Kalt av argalign slik:

argalign-p(arguments(F_s), 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 $t \in adj_t$ **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: $argalign-p(args_s, adj_s, args_t, adj_t)$

kjing av f-strukturane F_s og F_t i (1) under:

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

Men det kan 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 medlem i *aligntable*, 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. Hjelpefunksjonen *adjalign* (ikkje vist her **TODO: implementere :->**) konstruerer moglege permutasjonar av lenkjer mellom adjunkt som ikkje er inkludert i *argperm*, og *f-align* prøver desse rekursivt 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.

og så er det spørsmålet om me kan lenkje adjunkt på ulike nivå i f-strukturane

Viss F_s og F_t ikkje hadde argument i det heile teke, går me au gjennom moglege permutasjonar av adjunktdøtre, på same måte (ikkje vist i kodefigur 1).

4.1.2 SKRIV Funksjonsord utan LPT-korrespondanse bør eigentleg hindre f-strukturlenkjing

4.1.3 SKRIV Når f-lenkjene ikkje er 1-1

kausativ

preposisjonsobjekt

“sigaretten” og “sigaretze” er ikkje på same nivå i dei respektive f-strukturane nedanfor:

```
0 [ PRED vedde<28,29,27,30>
  29 [ PRED sigarett<> ] ]
```

Dette må 1. spesifiserast (kap.3), og 2. implementerast...

```

0[ PRED da-najleveba<37,10,46>
  ADJUNCT { 2 }
  2[ ze<5>
    OBJ 5[ sigareti ] ] ]

```

Men dette er fordi sigaretze er adjunkt, og i følge fotnote 3 i Dyvik et al. (2009) er P(ADJ) lik den semantiske formen til preposisjonsobjektet til ADJ. (Er dette heilt enkelt? Kan me ha adjunkt med fleire «preposisjonsargument»?)

I motsetning til å endre på P(ADJ) slik at pred-verdien til eit adjunkt hoppar over preposisjonar og inn i preposisjonsobjekt, har eg i implementasjonen min gjort det slik at funksjonen som henter ut *f-strukturen* til adjunkt-døtre av ein *f-struktur* hoppar over preposisjonsobjekt (m.a. fordi ved eit kall P(X) kan ikkje P vite om X er eit adjunkt eller argument).

Sjå au del 3.6.4.

4.1.4 Kan me gjere *f-struktursamanstillinga bottom-up*?

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, ville maskinen raskt fått 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.

4.2 SKRIV Rangering

Rangering er ikkje implementert *enno* (akkurat no gir `rank(f-alignments)` berre ut første samanstilling.) Rangering foregår etter ulike kriterium. Her er eit par forslag:

4.2.1 rekursivt lenkja > ulenkja, men LPT

alignable seier om noko er rekursivt lenkja eller ikkje, plusspoeng viss me har klart å lenkje rekursivt.

4.2.2 argument-argument > argument-adjunkt

Plusspoeng for argument-argument-lenkjer, burde vere eit bra kriterium, men me får sjølvstøtt problem viss LPT ikkje seier noko i døme (5) med
`da-najleveba<Abrams,Browne,regne> adjunkt: sigarett`
`bet<Abrams,sigarett,regne> adjunkt: Browne`

4.2.3 arg1-arg1 arg2-arg2 > arg1-arg2 arg2-arg1 (følgje)

Dette kjem til å gi problem når me vil lenkje «behage» og «like», viss me ikkje har motstridande LPT-informasjon (og argumentfølgje i leksikon ikkje er basert på semantikk, men syntaks). Men elles er det vel OK.

Enklaste implementasjon: Levenshtein-avstand. Men burde visse argument vek-
 tast? (T.d. vekte subjekt om alt anna er likt.)

Andre forslag: http://en.wikipedia.org/wiki/Edit_distance

4.2.4 flest lenkja adjunkt

Usikker på dette... avheng av om me tillèt lenkjer på tvers av f-strukturar.

4.2.5 Prioritet på rangeringskriterium

Dette bør sjølvstøtt testast empirisk, blir kanskje utanfor denne oppgåva (diskusjonsdel?), men kan jo prøve meg litt rundt.

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 sist⁶.

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

⁶Som nemnt i del 3.8 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.

eit par der første element er ei mengd c-strukturknoder på kjeldespråket, og andre element ei mengd knoder 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-strukturknoder at dei dominerer same mengd med ordlenkjer⁷. 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 knoder er dei mest ordnære nodane som kan ha ei f-strukturlenkje (ved \emptyset); når formålet er å lenkje c-strukturknoder kan me nytte f-strukturlenkja til den preterminale noden i staden for ordlenkjer. *To problem (kva vil me ha med?)*

1. me får *ikkje* med LPT-korrespondansar som er OK, men ikkje med i *f-alignment*;
2. me får med LPT-korrespondansar som er med i *f-alignment* men ikkje *aligntable* (ikkje er rekursivt lenkja).

Programmet `lfgalign` følgjer krav (8) og lenkjer øvste knoder i funksjonelle domene, og subordinate knoder som har same informasjonstap. Prosedyren `c-align` i kodefigur 3 implementerer dette kravet.

Hjelpeprosedyren `add-links` (kodefigur 4) 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 (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` kalla på kvar av døtrene. Viss ein node dominerer ei lenkjemengd *links*, legg me til denne noden i tabellen `splits[links]`. Merk at kvar c-strukturknode berre opptre éin gong i tabellen.

Sidan `c-align` kallar `add-links` for kvar av sidene, får me to tabellar `splitss` og `splitst`. Me går så gjennom kvar f-strukturlenkje; viss

Proessen er no ferdig, mange-til-mange-lenkjene mellom c-strukturknoder definerer frasesamanstillinga

til diskusjonsdel: Det er ikkje berre ei N-gramsamanstilling; sidan lenkjene er mellom c-strukturknoder kor kvar node dominerer ein konstituent, kunne me kalla det ei konstituentsamanstilling..

4.3.1 SKRIV viss me har LPT, men ikkje rekursiv f-lenkje

Dette bør kanskje vere valfritt i programmet, for å sjå kva det fører til: vil du ta med LPT-korrespondansar som ikkje har f-lenkjer i `add-links`?

Og omvendt, finst det LPT-korrespondansar som ikkje kjem med i *f-alignment* i det heile teke, men som likevel burde ha noko å seie for c-strukturlenkjinga? (Men burde dei ikkje då vere med i *f-alignment* au?)

⁷Dette 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.

Kan me fjerne visse f-samanstillingar pga. c-strukturinfo? dvs. disambiguere... (dette er vel heller stoff for diskusjonsdelen?)

```

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 src, trg in f-alignment do
    tops  $\leftarrow$  topnode(src) ;
    topt  $\leftarrow$  topnode(trg) ;
    if tops =  $\emptyset$  or topt =  $\emptyset$  then
        | continue;
    add (tops, topt) to c-alignments ;
    forall the link being the keys of splitss do
        if link in keys of splitst and link  $\neq$  (src, trg) then
            // We try to remove link from splitss and splitst
            linkss  $\leftarrow$  keys(src, splitss) - link ;
            linkst  $\leftarrow$  keys(trg, splitst) - link ;
            nodess  $\leftarrow$  splitss[linkss] ;
            nodest  $\leftarrow$  splitst[linkst] ;
            if nodess and nodest are non-empty then
                // nodess and nodest have the same info loss
                | add (nodess, nodest) to c-alignments ;
        |
    end
end
return c-alignments ;

```

Funksjon 3: c-align(*f-alignment*, *tree_s*, *tree_t*)

```

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  $\leftarrow$  {link}
    else
        | links  $\leftarrow$  add-links(f-alignment, left-branch(node))  $\cup$ 
        | add-links(f-alignment, right-branch(node)) ;
        | add node to splits[links] ;
return links ;

```

Funksjon 4: add-links(*f-alignment*, *node*, *splits*)

Kapittel 5

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>
- 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
- 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., 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>
- 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.
- 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
- 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., 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>
- 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.

- Thunes, M. (2003). *Ekserpering av leksikalske oversettelsekorrespondanser fra parallelltekst*. Tilgjengelig frå <http://www.hf.uib.no/i/LiLi/SLF/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.
- XPar. (2008). *XPAR: Language diversity and parallel grammars*. (Submitted to the Research Council of Norway.)