Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2003

Luennot: **Timo Honkela ja Krista Lagus**

Laskuharjoitukset: **Vesa Siivola**

Luentokalvot: Krista Lagus ja Timo Honkela

10. Probabilistinen jäsentäminen ja PCFG:t 3
10.1 Mikä on PCFG 6
10.2 Lauseen todennäköisyyden laskeminen 16
10.3 Inside-algoritmi 18
10.4 Todennäköisimmän jäsennyksen etsiminen 20
10.5 PCFG:n estimointi 21
10.6 (P)CFG:n ongelmia 26
10.7 Probabilistinen leksikalisoitu CFG 35

10. Probabilistinen jäsentäminen ja PCFG:t

PCFG=Probabilistic Context Free Grammar
Todennäköisyyksiin perustuva yhteysvapaa (kontekstivapaa) kielioppi

Motivaatio

- Tähän asti kielen säännönmukaisuuksia kuvattu vain sanatasolla (n-grammimallit) tai sanakategoriatasolla (sanaluokat ja HMM)
- Sekventiaalisten riippuvuuksien lisäksi muunkinlaisia ilmiöitä, esim. rekursiivisia.
- Esim. 'The velocity of the seismic waves rises to...'
  P('waves rises') on hyvin pieni esim. sanaluokkiin perustuvalla HMM-taggerilla, koska yksiköllistä verbimuotoa harvoin edeltää monikollinen substantiivi (yleensä sitä edeltää yksiköllinen substantiivi, verbin subjekti; ilmiötä kutsutaan nimellä 'verb agreement').

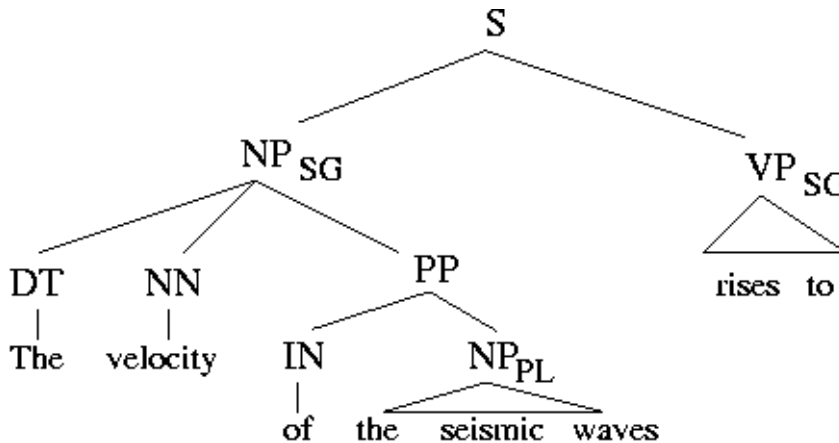
Motivaatio, jatkoa

- Keskeinen havainto: yksittäisten sanojen sijaan riippuvuudet voidaan ehkä paremmin kuvata suurempien kokonaisuuksien välillä → oletetaan hierarkkinen rakenne myös lauseiden tai virkkeiden sisällä. (Hierarkkisuus yleisemmällä tasolla: dokumentti, kappale, lause, sana, kirjain, veto).

- Vastaavaa hierarkkisuutta myös muualla kuin kielessä, esim. konenäön hahmontunnistusongelmat, joihin myös voidaan soveltaa syntaktisia hahmontunnistusmenetelmiä.

Lauseen hierarkkinen rakenne

Seuraavassa on esimerkki lauseen sisäisestä hierarkkisesta rakenteesta (kirjan kuva 11.2). Voidaan ajatella että riippuvuus on tässä peräkkäiden rakenneosien NP_{SG} ja VP_{SG} välillä.



10.1 Mikä on PCFG

CFG:n luonteva laajennus: CFG, jonka sääntöihin on liitetty laukeamistodennäköisyydet. [CFG = kontekstivapaa kielioppi]

PCFG on probabilististen kielioppien eräs alalaji, ts. vain eräs tapa mallintaa hierarkkista rakennetta probabilistisesti. Tekemällä erilaisia riippumattomuusoletuksia päädytään erilaiseen malliperheeseen.

PCFG:n osat

PCFG koostuu seuraavista osista:

- terminaalisymbolien joukko,
- ei-terminaalisymbolien joukko,
- joukko korvaussääntöjä $N^i \rightarrow \gamma^j$ jossa γ^j on terminaali- ja ei-terminaalisymboleista koostuva jono
- Kullekin säännölle ehdollinen laukeamistodennäköisyys:

$$\sum_j P(N^i \rightarrow \gamma^j | N^i) = 1 \tag{1}$$

Notaatiota

G	kielioppi
$\mathcal{L}_G$	kieli jonka kielioppi G tuottaa
t	jäsennyspuu
tulostus(t)	jäsennyspuun t jäsentämä sanajono
$\{N^1, \ldots, N^n\}$	ei-terminaalisymbolien aakkosto (N^1 on alkusymboli)
$\{w^1, \ldots, w^V\}$	terminaalisymbolien aakkosto
$w_1 \ldots w_m$	jäsennettävä lause (sanajono)
N^j_{pq}	ei-terminaalisymboli N^j kattaa jonossa positiot p:stä q:hun.

Riippumattomuusoletukset

PCFG:ssä korvaussäännön tn on ehdollinen ainoastaan sille,
missä ei-terminaalissa nyt ollaan, eli N^i.

Tarkastellaan tarkemmin riippumattomuusoletuksia, joita tässä tehdään.

- Rakenteellinen paikkariippumattomuus: alipuun tn ei riipu siitä, missä osassa isompaa puuta alipuu sijaitsee (vrt. HMM:n ajallinen stationaarisuus).
- Leksikaalinen kontekstiriippumattomuus: alipuun tn ei riipu alipuuhun kuulumattomista kontekstin sanoista.
- Riippumattomuus esivanhemmista: Alipuun tn ei riipu siitä, mitä sääntöjä käyttäen alipuu generoitiin.

Huomioita probabilistisesta kieliopista

- Antaa probabilistisen mallin kielen generoinnille.
- Arvioi kunkin jäsennyksen todennäköisyyden. Kattavissa kieliopeissa yhdelle lauseelle yleensä hyvin monta mahdollista jäsennystä, jopa tuhansia per lause.
- Prob. kielioppi (esim. PCFG-malliperhettä käyttäen opittu) ei yritäkään jakaa lauseita ei-kieliopillisiin ja kieliopillisiin. 'Virheelliset' lauseet vain ovat paljon epätodennäköisempiä.
- Jos dataa on riittävästi, PCFG:llä kieliopin oppiminen mahdollista ilman negatiivista evidenssiä (ei-kieliopillisia lauseita).

Peruskysymykset PCFG:lle

- Lauseen todennäköisyyden laskeminen
- Todennäköisimmän jäsennyksen valinta lauseelle
- PCFG:n parametrien estimointi joko jäsennetystä (ohjattu oppiminen) tai jäsentämättömästä datasta (ohjaamaton oppiminen)

Chomskyn normaalimuoto

Keskitytään tästedes kontekstivapaisiin kielioppeihin, jotka ovat *Chomskyn normaalimuodossa* eli säännöt ovat binäärisiä tai unaarisia:

$N^i \rightarrow N^j N^k$
$N^i \rightarrow w^j$

PCFG:n parametrien määrät ovat tällöin:

$P(N^j \rightarrow N^r N^s \mid G)$	Jos n ei-terminaalia, n^3 parametria
$P(N^j \rightarrow w^k \mid G)$	Jos V terminaalia, nV parametria

Kaikille j pätee lisäksi:

$$\sum_{r,s} P(N^j \rightarrow N^r N^s) + \sum_k P(N^j \rightarrow w^j) = 1 \qquad (2)$$

Tämä johtuu siitä että yo. parametrit ovat ehdollisia todennäköisyyksiä ehdolla N^j.

Chomskyn hierarkia kielille

Yhteys-/kontekstivapaus on eräs kielen kompleksisuutta kuvaava taso. Chomsky jakaa kielet seuraavanlaiseen kompleksisuushierarkiaan: (Taulukko on kirjasta Jurafsky & Martin, s. 479)

Tässä A on yksittäinen ei-terminaalisymboli ja α, β ja γ ovat mitä tahansa terminaalien ja ei-terminaalien jonoja.

Tyyppi	**Nimi**	**Sääntörunko**
0	Turing-ekvivalentti	$\alpha \rightarrow \beta$ s.e. $\alpha \neq \epsilon$
1	Kontekstiherkkä	$\alpha A \beta \rightarrow \alpha \gamma \beta$ s.e. $\gamma \neq \epsilon$
2	Kontekstivapaa	$A \rightarrow \gamma$
3	Säännöllinen	$A \rightarrow xB$ tai $A \rightarrow x$

Chomskyn hierarkia kielille, selitykset

Tyyppi 0 vastaavat niitä kieliä, joiden symbolijonot voidaan tuottaa (listata) Turing-koneella. Ainoa rajoitus säännöille on, että "tyhjästä ei voi nyhjästä".

Kontekstiherkät kielet muuntavat symbolin toiseksi riippuen sen oikean- ja vasemmanpuoleisesta kontekstista. Sääntöjen on myös tuotettava jotakin.

Kontekstivapaissa kielissä ei-terminaalisymboli voidaan korvata millä tahansa ei-terminaalien ja terminaalien jonolla (mukaanlukien tyhjä jono). Näitä toteuttavat esim. lauserakennekieliopit.

Säännölliset kielet ovat ekvivalentteja säännöllisten lausekkeiden (regular expressions) kanssa. Ne voivat olla oikea- tai vasenkätisesti lineaarisia. Niitä toteuttavat esim. äärelliset tilakoneet.

Yhtäläisyys HMM:n ja PRG:n välillä

PRG = probabilistic regular grammar, eli todennäköisyyksiä hyödyntävä säännöllinen kielioppi.

Esimerkkilause *P(John decided to bake a)* saisi luultavasti korkean todennäköisyyden HMM:ssä (koska se on järkevä sanajono) mutta matalan PRG:ssä (koska ei ole kokonainen lause).

PRG voidaan kuitenkin implementoida HMM:n avulla lisäämällä HMM:ään alkutilan lisäksi lopputila, joka vastaa lauseen päättymistä.

10.2 Lauseen todennäköisyyden laskeminen

$$P(w_{1m}|G) = \sum_{t} P(w_{1m}, t|G) \quad (3)$$

$$= \sum_{t, tulostus(t)=w_{1m}} P(t|G)g \quad (4)$$

jossa t on kieliopin G säännöillä tuotettu jäsennyspuu lauseelle w_{1m}.

Esimerkki lauseen todennäköisyyden laskemisesta

Oletetaan että t:n aikaansaamiseksi sovellettiin G:n sääntöjä s^i,s^k, ja s^a ja tulostuksena oli sanajono w_{pq}. Tällöin

$$\begin{aligned} P(t|G) &= P(s^i s^k s^a) = P(s^i, s^k, s^a) \\ &= P(s^a|s^i, s^k)P(s^k|s^i)P(s^i) \\ &= P(s^a)P(s^k)P(s^i) \end{aligned}$$

Ylläolevassa kaikki tn:t ovat ehdollisia kieliopille G, joten tämä on jätetty merkitsemättä.

Eli $P(w_{pq}|G)$ on niiden G:n sääntöjen todennäköisyyksien tulo joita t:n muodostamiseksi sovellettiin.

Yleisessä tapauksessa ei kannata (laskennallisista syistä) summata yli kaikkien mahdollisten jäsennysten. Samoin kuin HMM:llä mahdollisia 'polkuja' on liikaa. Ongelma voidaan paloitella osaongelmiksi, ratkaista palat kerrallaan ja vierittää palojen tuloksia eteenpäin.

10.3 Inside-algoritmi

Inside-algoritmi vastaa HMM:ien forward-algoritmia ja sillä voidaan laskea lauseen todennäköisyys.

$b_j(p,q)$ tarkoittaa N_j:n alla olevan alipuun, joka kattaa sanajonon w_{pq}, tn:n laskemista

Perustapaus: lasketaan tn, kun muunnetaan ei-terminaali terminaaliksi:
$b_j(k,k) = (w_k|N_k^j k, G)$

Induktioaskel: tn, kun muunnetaan ei-terminaali kahdeksi ei-terminaaliksi: oletetaan sääntö $N^j \rightarrow N^r N^s$ ja jonon jako $w_{pq} = w_{pd} w_{dq}$

Tällöin $b_j(p,q) = \sum_{r,s} \sum_{d=p}^{q-1} b_r(p,d) \times b_s(d+1,q)$

Eli samalla, kun puu jakautuu kahdeksi osapuuksi, jono w_{pq} jaetaan kahtia, ja rekursiivisesti sovelletaan induktioaskelta kumpaankin osapuuhun.

Lisäksi summataan yli kaikkien mahdollisten jonon jakojen ja kaikkien ei-terminaalisymbolien nimien, jotka voidaan muuntaa N^j:stä kieliopissa G.

Inside-todennäköisyydet voidaan laskea tehokkaasti alhaalta ylös (bottom-up).

10.4 Todennäköisimmän jäsennyksen etsiminen

Todennäköisimmän jäsennyksen etsimiseen (samoin kuin HMM:llä) voidaan soveltaa Viterbi-tyypistä algoritmia.

Tämä on kompleksisuudeltaan $\mathcal{O}(n^3m^3)$ jossa m on jonon pituus ja n eiterminaalien lukumäärä kieliopissa.

10.5 PCFG:n estimointi

Annettuna sääntöjoukko:

Tapaus 1: Jokin olemassaoleva CFG, joka hyväksyy koko aineiston.

Tapaus 2: Kaikkien mahdollisten sääntöjen joukko.

Tapaus 3: Kaikkien mahdollisten sääntöjen joukko paitsi, että oletetaan valmiiksi jotakin rakennetta, esim. jokin ei-terminaalisymbolien osajoukko varattu generoimaan pelkästään terminaalisymboleja.

Estimointi, kun käytettävissä jäsennettyä aineistoa

Annettuna sääntöjoukon Tapaus 1 sekä 'puupankki' (**treebank**) eli suuri määrä jäsennettyä aineistoa.

Lasketaan suoraan sääntöjen soveltamiskertoja ja normalisoidaan.
ML-estimaatti:

$$P(\alpha \rightarrow \beta) = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

Estimointi, kun käytettävissä vain jäsentämätöntä aineistoa

Sääntöjoukon tapaus 1:

Jäsennetään aineisto annetulla CFG:llä ja tämän jälkeen kuten edellä.

Käytännössä jäsennykset moniselitteisiä: yhden jäsennyksen sijaan painotetaan eri jäsennyksiä niiden tn:llä.

Sääntöjoukon tapaukset 2 ja 3:

Kuten HMM:llä, sovelletaan erästä EM-algoritmia, Inside-Outside-algoritmia, joka muistuttaa forward-backward-algoritmia.

Estimointialgoritmi

Mallin ML-estimointi: maksimoidaan datan todennäköisyys.

Parametreja ovat sääntöjen tn:t.

Perusperiaate:

- Alustus esim. satunnaisilla parametreilla.
- Vaihe 1: Sovelletaan tämänhetkistä mallia (PCFG:tä) datan tn:n laskemiseen ja samalla pidetään kirjaa siitä, kuinka paljon mitäkin sääntöä käytettiin.
- Vaihe 2: Päivitetään sääntöjen tn:t vaiheen 1 kirjanpidon perusteella.
- Toistetaan vaiheita 1 ja 2.

Ongelmia probabilistisen kieliopin ei-ohjatussa oppimisessa

- Ongelman kompleksisuus ja tästä johtuva estimoinnin hitaus. Jokaiselle lauseelle jokainen iteraatio on kompleksisuudeltaan $\mathcal{O}(m^3n^3)$, jossa m on lauseen pituus ja n ei-terminaalien määrä lauseessa.

- Kustannusfunktion lokaalit minimit ovat hyvin todennäköisiä. Esim. yksinkertaisella keinotekoisella PCFG:stä generoidulla aineistolla estimoitaessa 300 eri ajolla päädyttiin jokaisella eri lokaaliin minimiin. Alustuksella on siis hyvin suuri vaikutus.

- Mallien koko kasvaa helposti suureksi, koska mallin kokoa ei kustannusfunktiossa mitenkään huomioida. Ratkaisu: kustannusfunktion vaihto sellaiseksi, joka huomioi myös mallin koon. Perusteltuja lähestymistapoja tähän ovat MDL ja Bayesilainen estimointi.

- Ei-terminaaleilla, joita malli oppii, ei välttämättä ole mitään tekemistä lingvistisen teorian ei-terminaalien kanssa.

10.6 (P)CFG:n ongelmia

- Säännön soveltamisen todennäköisyys ei riipu siitä, missä kohtaa koko puuta alipuu sijaitsee
- (P)CFG ei huomioi leksikaalista kontekstia

Säännön soveltamisen todennäköisyys ei riipu siitä, missä kohtaa koko puuta alipuu sijaitsee.

CFG:n riippumattomuusoletukset → säännön soveltaminen riippuu vain senhetkisestä ei-terminaalisymbolista. Kuitenkin esim. englannissa pronominilausekkeen esiintymistn riippuu voimakkaasti sijainnista koko lauseessa:

Väitelauseiden subjektina pronominin P=91% (loput 9% leksikaalisia), kun taas objektina P=34% (loput 66% leksikaalisia).

Usein lauseen alkupäässä viitataan siihen mitä jo tiedetään aiemman keskustelun pohjalta ja loppupäässä tuodaan ilmi ko. asiaa koskevaa uutta tietoa:

Presidentti Tarja Halonen aloitti kautensa maakuntakierroksella.
Hän tapasi ensin Joensuun kaupungin johtavia virkamiehiä ja jatkoi ...

Ongelma ratkeaisi, jos P(NP → Pronomini) ja P(NP → Nomini) ehdollistettaisiin sillä, onko NP subjekti- vai objektipositiossa. Mutta juuri tätä eivät PCFG:n riippumattomuusoletukset salli.

(P)CFG ei huomioi leksikaalista kontekstia

Englannissa prepositiolausekkeen kiinnittymisongelma:
'I shot an elephant in my pyjamas'
'USA sent more than 10,000 soldiers into Afghanistan'

Sotilaiden lähettämisen tapauksessa leksikaalinen tieto eli että kyseessä on send-verbi ja into-prepositio auttavat disambiguoimaan esimerkin. Elefantti-esimerkkiin saatetaan tarvita maailmantietoa: elefantit eivät yleensä käytä yöpukuja paitsi saduissa.

Suomessa vastaavanlainen monitulkintaisuus:

'Ammuin elefantin yöpuvussa'
→ Ammuin elefantin jolla oli päällään yöpuku.
→ Ammuin yöpukusillani elefantin.

Vastaava ilmiö: Koordinaatiomonitulkintaisuus:

'dogs in houses and cats' $\rightarrow$
1. 'dogs in [NP houses and cats]'
2. '[NP dogs in houses] and cats'

Semanttinen preferointi: koirat eivät mahdu kissoihin, joten 2. vaihtoehto olisi oikea.

(P)CFG:ssä kuitenkin ylläoleviin lukutapoihin sovellettavat säännöt ovat rakenteisesti identtisiä, eli ei kykene preferoimaan vaihtoehtoja.

Joitain havaintoja toimivuudesta englannille:

- PCFG:n ennustuskyky tavallisesti parempi kuin HMM:n, kun sama määrä parametrejä.
- Englannille PCFG yleensä huonompi kielimalli kuin n-grammimalli. Johtuu siitä, ettei kontekstin leksikaalista tietoa huomioida (riippumattomuusoletukset).
- Puhdas PCFG ei siis ole hyvä, mutta ehdollistaminen leksikaalisella tiedolla tai puun rakennekontekstilla voi parantaa tulosta.

Onko PCFG aito kielimalli?

Jotta PCFG olisi aito kielimalli, täytyisi päteä:
$\sum_{w \in \mathcal{L}} = \sum_t P(t) = 1$
eli kieliopin tuottamien lauseiden saaman todennäköisyysmassan pitäisi olla 1.

Kuitenkin tämän toteutuminen edellyttää, että mallissa ei ole mahdollisuutta äärettömään rekursioon. Jos ääretön rekursio on mahdollista, osa tn-massasta katoaa sinne, eikä päädy koskaan kielen lauseiksi.

Esimerkki

Tarkastellaan esimerkiksi kielioppia:
S $\rightarrow$ raparperi, P=1/3
S $\rightarrow$ S S, P = 2/3

Tässä suuri osa eli noin 42% jäsennysten tn-massasta katoaa äärettömään rekursioon. Kyseessä on *epäkonsistentti* tn-jakauma.

Ongelmattomia tapauksia

Todennäköisyysmassan katoaminen rekursiossa ei ole ongelma, kun

- etsitään todennäköisin jäsennys annetulle lauseelle,
- vertaillaan kahden lauseen tn:ää annetussa kieliopissa tai
- estimoidaan PCFG:n parametrit jäsennetystä korpuksesta (Chi & Geman, 1998)

Ongelmallisia tapauksia

Todennäköisyysmassan katoaminen on ongelma, kun tehdään vertailuja eri kielioppien/kielimallien välillä, siis kun:

- lasketaan jonkin lauseen tn tässä kieliopissa ja verrataan tn:ään jossain toisessa kieliopissa (jossa tn-massa kokonaan tai sen erisuuri osuus kieliopin lauseilla),
- estimoidaan PCFG:n parametrit ohjaamattomasti toisin sanoen jäsentämättömästä korpuksesta, tai
- tehdään mallin valintaa eli vertaillaan eri CFG:n sääntöjoukkoja

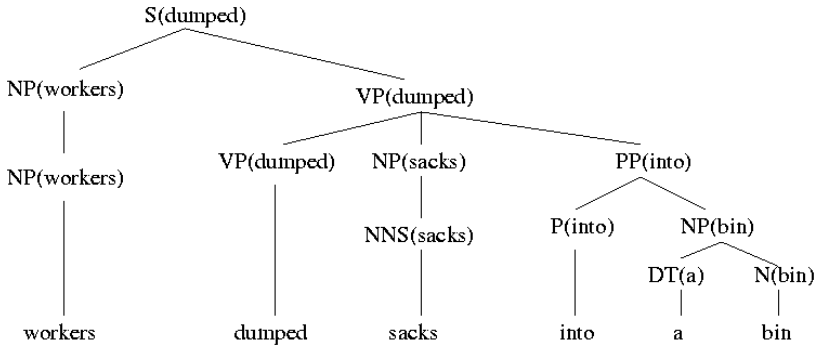
10.7 Probabilistinen leksikalisoitu CFG

Leksikalisointi tarkoittaa sanatiedon ottamista huomioon.

Menetelmä:

- Jokaiseen ei-terminaalisymboli N^i jäsennyspuussa t liitetään ko. alipuun 'pääsana' (head) (Charniak 1997, Collins 1999)
- CFG:ssä jokaisen säännön kohdalle merkitään, mikä säännön oikealla puolella olevista symboleista on 'päätytär' ts. sisältää pääsymbolin (esim. NN on NP:n päätytär).
- Päätyttären valinta suoraviivaista oppikirjalauseille, mutta yleisesti ottaen hankalaa (lingvistiikan kirjat saattavat spesifioida säännöt tämän tekemiseen kyseiselle kielelle).

Puuesimerkki



Attribuuttikieliopeista

- Kielioppia, jossa jokaisella ei-terminaalinoodilla on myös jokin terminaalisymboliarvo, kutsutaan myös *attribuuttikieliopiksi*.

- PCFG:ssä jokaista päätyttären mahdollista arvoa on varten oma todennäköisyys (ts. oma sääntö). Periaatteessa siis esim:

  $VP(dumped) \rightarrow VBD(dumped)NP(sacks)PP(into)[3 \times 10^{-10}]$
  $VP(dumped) \rightarrow VBD(dumped)NP(cats)PP(into)[8 \times 10^{-11}]$
  $VP(dumped) \rightarrow VBD(dumped)NP(hats)PP(into)[4 \times 10^{-10}]$
  $VP(dumped) \rightarrow VBD(dumped)NP(sacks)PP(above)[1 \times 10^{-12}]$

- Käytännössä mikään aineisto ei riitä näin valtavan mallin estimointiin $\rightarrow$ tehdään taas tiettyjä riippumattomuusoletuksia, joiden avulla ryhmitellään (sidotaan) osa todennäköisyyksistä samoiksi.

- Esim. voidaan käyttää leksikalisointia vain säännön vasemmalla puolella:
  $VP(dumped) \rightarrow VBD\ NP\ PP$

- Eri tilastolliset jäsentimet eroavat toisistaan siinä, minkälaisia riippumattomuusoletuksia niissä tehdään.
- Suuremmasta sääntömäärästä johtuen leksikalisoidun PCFG:n estimointi ohjaamattomalla (ilman jäsennettyä dataa) oppimisella käytännössä mahdotonta, ainakin toistaiseksi.