Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2004

Luennot: **Timo Honkela**
Laskuharjoitukset: **Vesa Siivola**

Luentokalvot: Krista Lagus (päivityksiä: Timo Honkela)

10. Sanaluokkien taggaus

Esimerkki:

The-AT representative-NN put-VBD chairs-NNS on-IN the-AT table-NN.

Yllä sanoille 'put' ja 'chairs' on olemassa sekä verbi- että substantiivitulkinta.

Ilmiö on yleinen: yleensä substantiivista voi helposti tehdä myös verbin ja useilla pääasiassa verbeillä on myös harvinaisempi substantiivikäyttö.

Next, you **flour** the pan.
I want you to **web** our annual report.

10.1 Syntaktinen taggaus

Syntaktisen taggauksen sovelluskohteita:

- Information extraction (jonkin nimenomaisen tyyppisen tiedon esiinkaivaminen), esim. erisnimien tunnistaminen tekstidokumenteissa)
- Kysymyksiin vastaaminen (question answering)
- Osittainen jäsentäminen (shallow/partial parsing)
- Yleisesti: tilanteet joissa ei tarvita täydellistä kielen analyysiä ja ymmärtä

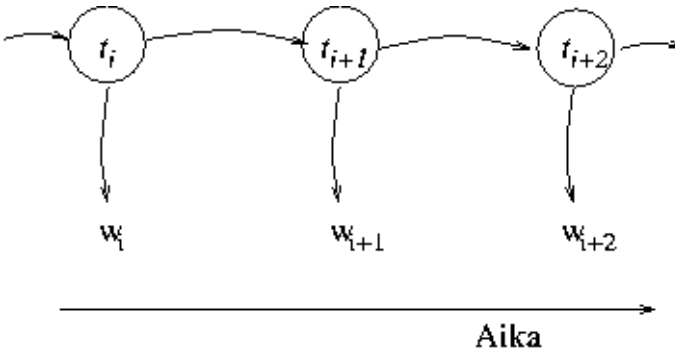
10.2 Taggauksessa käytettävä informaatio

- Rakenneinformaatio, eli lähiympäristön syntaktiset tagit: tietyt tagisekvenssit tavallisempia kuin toiset
- Sanakohtainen informaatio: Sanan prioritn kuulua tiettyyn syntaktiseen luokkaan: eri luokkien tn-jakauma yleensä hyvin epätasainen kullekin yksittäiselle sanalle (vastaavasti kuin semanttisessa moniselitteisyydessä jokin tulkinta on hyvin tyypillinen, vaikka useita mahdollisia eri tulkintoja)

10.3 Huomioita koskien englannin kieltä

- Taggaus helpompi ongelma kuin jäsentäminen, tarkkuudet korkeita
- Parhaat taggerit luokkaa 96%-97% (oikein tagattujen sanojen osuus - tarkoittaa, että lauseissa keskimäärin 1-2 taggausvirhettä, jos lausepituus keskim. 20).
- Pelkkää rakenteista informaatiota käyttävälle sääntöpohjaiselle taggerille raportoitu (vain) 77% tarkkuus.
- Yksinkertainen menetelmä joka ei käytä tietoa rakenteista lainkaan: Luokitellaan sana aina yleisimpään POS (part of speech) -luokkaansa. Englannille raportoitu 90% tarkkuus → käytetään usein baseline-menetel

10.4 Markov-malli-taggerit



- Tagijono mallinnetaan Markov-ketjuna, eli $P(X_{i+1} = t^j | X_1, \ldots, X_i) = P(X_{i+1} = t^j | X_i)$ jossa i on ajanhetki ja t^j on tila jonka indeksi j. X:n arvojoukko on tilojen joukko, $\{t^1 \ldots t^n\}$.

- Sanat ovat havaintoja (observations), todennäköisyysmalli $P(w_i | X_i)$ eli sanan generointitn riippuu vain kulloisestakin tilasta.

- Bigrammitaggerissa jokainen tagi vastaa yhtä tilaa. Tällöin tämänhetkise sanan tagi riippuu ainoastaan edeltävän sanan tagista.
- Malli opetetaan tagatulla datalla näkyvänä Markov-mallina eli tila-muuttujan arvo tunnetaan kullakin hetkellä (kullakin sanalla).
- Tagattaessa uutta dataa mallia käytetään HMM-mallina: tiloja ei tunneta vaan todennäköisin tilajono annetulla sanajonolla lasketaan mallista Viterbi-algoritmilla. Nyt Viterbin käyttö ei ongelmallista, koska ollaan kiinnostuttu nimenomaan todennäköisimmästä tilajonosta, ei sanajonosta.
- Markov-mallin rajallinen horisontti-riippumattomuusoletus ei aivan päde, edes englannille. Vielä vähemmän kielille, joissa sanajärjestyksen määräyt syntaktiset tekijät eivät ole yhtä keskeisiä.
- Sanojen generointitodennäköisyydet tageista:

$$P(w_{1\ldots n}|t_{1\ldots n}) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \quad (1)$$

Estimoinnissa tehtäviä muita riippumattomuusoletuksia:

  - Sanat riippumattomia toisistaan.
  - Sanan tn. riippuu ainoastaan sen tagista (eli tagi generoi sanan).

- Optimaalisen tagijonon estimointi lauseelle, sovelletaan Bayesin sääntöä:

$$\hat{t}_{1\ldots n} = \arg\max_{t_{1\ldots n}} P(t_{1\ldots n}|w_{1\ldots n}) \quad (2)$$

$$= \arg\max_{t_{1\ldots n}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \quad (3)$$

  Intuitiivisesti: Valitaan maksimaalisen todennäköinen reitti, jolla päästää markov-ketjua pitkin tagista t_{i-1} sanaan w_i, eli maksimaalisen todennäköinen tapa generoida havaittu sana w_i kun on kiinnitetty tagi t_{i-1}.

- Huom: Kannattaa käyttää tasoitusta laskettaessa $P(t^k|t^j)$ ja $P(w^l|t^j)$ (ei siis pelkkiä ML-estimaatteja).
- Todennäköisimmän tagijonon haku tehokkaasti Viterbi-algoritmilla.

10.5 Tuntemattomien sanojen käsittely

Tunnetuille sanoille estimointi helppoa. Kuitenkin tuntemattomat sanat aiheuttavat usein suurimmat erot taggerien välillä.

Strategioita

- Yksinkertaisin lähestymistapa: tuntemattoman sanan tagitn:t seuraavat tagijakaumaa koko datan yli (ts. tuntemattoman sanan malli on painotettu keskiarvo kaikkien sanojen malleista).
  Ongelma: ei kovin hyvä estimaatti.

- Muiden piirteiden, esim. morfologian hyväksikäyttö: valitaan morfologialtaan samankaltaisten sanojen osajoukko ja lasketaan tn:t siitä. Esimerkki: jos sana loppuu '-ed', keskiarvoistetaan tagijakauma -ed -loppuisten sanojen yli.

Lisää strategioita

- Eräs malli (Weichschedel jne): katsotaan todennäköisyyttä jolla tagi generoi tuntemattomia sanoja, sekä sanan eri piirteiden generointitn:iä tästä tagista:

$$P(w^l|t^j) = \frac{1}{Z} P(tuntematon|t^j) P(isoalkuk|t^j) P(lopuke_i|t^j)$$

  jossa Z on normalisointitekijä ja $lopuke_i$ sanan w^l lopuke (esim. '-ed'). Joissain kokeissa mallin todettiin pudottavan tuntemattomien sanojen virhetn:iä 40%:sta 20%:een (tosin ei kerrota mikä oli vertailumenetelmä, tai datan koko).

- Useimmat mallit olettavat piirteet riippumattomiksi (ns. Naive Bayes-malli, kuten yllämainittu), mikä yleensä ei pidä paikkaansa. Esim. yllä isoalkukirjaimiset sanat ovat melko todennäköisesti myös tuntemattomia, koska ovat todennäköisesti erisnimiä, joten ko. piirteet eivät toisistaan riippumattomia.

Variantteja

- Tagin tn. voi riippua pidemmästä historiasta, esim. 2:sta ed. tagista.
- Esim. '...was clearly marked...' ja '...he clearly marked...' tagattaisiin 'BEZ RB VBN' ja 'PN RB VBD'.
- Kahden edellisen tagin (ja sanan itsensä) perusteella ennustamista voidaan kutsua trigrammitaggaukseksi.
- Historian pidentämisestä ei aina ole hyötyä: esim. syntaktiset riippuvuudet harvoin kulkevat pilkkujen yli. Harvan datan ja puutteellisen tasoituksen takia pitkästä historiasta voi olla haittaakin, ja trigrammitaggeri voi pärjätä bigrammitaggeria huonommin.
- Kuten puheentunnistuksessa, voidaan myös käyttää lineaarista interpolointia eri n-grammitaggerien yli, tai muita tasoitusmenetelmiä.
- Variable-Memory Markov Model (VMMM): eri tiloissa voi olla tiedossa eri pituinen historia. Opetusvaiheessa tilan historian pituus valitaan informaatioteoreettisella kriteerillä. Voidaan rakentaa joko topdown (tiloja halkomalla) tai bottom-up (yhdistelemällä).

10.6 HMM-taggerit

- Piilo-Markov-mallia voidaan soveltaa myös opetusvaiheessa, jos ei ole tagattua esimerkkidataa. Esim. kieli jolle ei olemassa tagattua dataa, tai tunnetun kielen osa-alue jossa sanojen generointitn:t ja/tai kielen tyypilliset rakenteet erilaisia kuin mitä opetusdatassa.
- Mallin rakennusosat: Tilat S, havainnot O, tilojen lähtötodennäköisyydet π_i, tilasiirtymätn:t a_ij, havaintojen generointitn:t b_{ijk}
- Kuten näkyvillä Markov-malleilla, tilat vastaavat jälleen tageja ja havainnot ovat sanoja.
- Periaatteessa voidaan initialisoida mallin parametrit eli todennäköisyydet π_i,a_ij, ja b_{ijk} satunnaisesti ja estimoida niitä iteratiivisesti yhä paremmiksi (esim. forward-backward-algoritmilla).
- Kuitenkaan tällä tavalla ei välttämättä päädytä taggaukseen joka vastaisi jotain olemassaolevaa lingvististä tagijoukkoa ja tagien syntaktisia rooleja. Pikemminkin tällä tavalla voidaan 'keksiä' taggaus joka toteuttaa mallin riippumattomuusoletukset.

- Tavallisemmin käytettävissä on tunnettu tagijoukko, sekä sanakirja jossa kerrotaan mitkä tagit mahdollisia tai mahdottomia millekin sanalle (esim. JJ ei mahdollinen sanaluokka sanalle 'book'). Eri tyyppisen sanakirjatiedon vaihtoehdot on kuvattu taulukossa alempana.

- Vaihtoehtoisesti ryhmitellään sanat, joille sallittu samat tagit, ekvivalenssiluokiksi, joille yhteiset parametrit (ainakin mallin initialisointivaiheessa). Voidaan soveltaa myös pelkästään harvinaisille sanoille, koska yleisten osalta dataa on riittävästi.

	Leksikaalinen resurssi	Strategia
L_0	Jokaiselle sanalle tunnetaan sallitut tagit	Sallituille tageille T^s $p(w\|t^j) = 1/(\#T^s)$, muille p=0.
L_1	Sallitut tagit tn-järjestyksessä	Annetaan satunnaiset tn-arvot, mutta järjestyksessä.
L_2	Tagien tn:t annettu kullekin sanalle	Käytetään näitä.

Koska tn-malli ei täysin vastaa todellisuutta, jos on käytettävissä riittävästi tagattua dataa, kannattaa opettaa pääasiassa sillä.

Täysin ohjaamattomasta oppimisesta vaikuttaisi olevan hyötyä lähinnä mikäli tagattua dataa on kovin vähän tai ei ollenkaan, tai jos tagattu testiaineisto (tai sovelluskäyttö) on melko erilaista kuin tagattu opetusaineisto.

Eräs tapa hahmottaa syy tähän: mallin rakenne sinällään ei vastaa kovin hyvin tarkoitusta, ja ilman tagattua opetusaineistoa mallin rakenteen merkitys dominoi.

Parametrien optimoinnissa voidaan käyttää erillistä (tagattua) validointijoukkoa, jolla varmistetaan, että opetusta jatketaan vain niin kauan kuin tarkkuus validointijoukolla paranee.

10.7 Muunnoksiin perustuva taggaus

- Edellä kuvatut mallirakenteet, eli mallien tekemät riippumattomuusoletukset, eivät erityisen hyvin soveltuneet luonnollisen kielen kuvaamiseen. Tarvitaan siis parempia malleja.

- Kontekstia (n-grammin n) voitaisiin pidentää. Tai tagin tn voisi riippua myös edeltävistä sanoista (ei pelkästään tageista). Ongelma: parametrien määrä moninkertaistuu, estimointiongelmia.

Toisenlainen lähestymistapa:
Muunnoksiin perustuva taggaus
(transformation-based tagging)

Soveltaminen ohjatun oppimisen ongelmaan:

- Tagattua dataa

- Sanakirja, jossa kerrotaan sanalle sallitut tagit ja näiden tn:t.

- joukko kontekstiriippuvia muunnossääntöjä ('virheenkorjauksia') joita taggaukselle voidaan tehdä.
- Algoritmi, jolla valitaan mitä muunnoksia milloinkin kannattaa soveltaa (ohjattu oppiminen)

Perusalgoritmi:

1. Tagataan aluksi jokainen sana todennäköisimmän taginsa mukaan.
2. Valitaan muunnoksia, jotka vievät taggausta vähitellen lähemmäksi oikeaa (opetusaineistossa olevaa) taggausta.

Muunnokset

- Muunnossääntö joka kertoo mikä tagi korvataan millä. Esim. 'korvaa tagi VBD tagilla NN'

- Heräteympäristö (triggering environment): olosuhteet, joissa muunnos aktivoituu. Esim. 'Tagi t^j esiintyy 2-3 sanaa ennen korvattavaa tagia ja t^k korvattavaa tagia seuraavassa sanassa'. ks. kirjan taulukko 10.7.
- Heräteympäristöön voidaan ottaa myös sanoja tai näiden ominaisuuksia, ei pelkästään tageja:
  Tag-triggered: heräteympäristössä voi olla tageja
  Word-triggered: heräteympäristössä voi olla sanoja
  Morphology-triggered: heräteympäristössä voi olla morfologisia piirteitä

Esimerkkejä muunnossäännöistä ja herätteistä englannille:

Muunnos	Heräte
NN $\Rightarrow$ VB	edellinen tagi oli TO
VBP $\Rightarrow$ VB	jokin kolmesta edellisestä tagista oli MD
JJR $\Rightarrow$ RBR	seuraava tagi on JJ
VBP $\Rightarrow$ VB	toinen kahdesta edellisestä sanasta ei ole *n't*

Oppimisalgoritmi

Sovelletaan ahnetta optimointia, valitaan joka kierroksella se transformaatio joka eniten vähentää virheellisten taggausten lukumäärää.

Notaatio: Transformaatiot $u_i(\cdot)$, $v(\cdot)$, C_k on korpus k:n transformaation soveltamisen jälkeen. $E(\cdot)$ on virhemäärä ja ϵ virheen pieni kynnysarvo.

Algoritmi:

- Alustus: C_0, korpus jossa jokainen sana tagattuna yleisimmällä tagillaan.
- for (k=1; ; k++)
  - v:= valitse transformaatio u joka minimoi virheen $E(u_i(C_k))$
  - Jos v ei pienennä virhettä tämänhetkiseen verrattuna enempää kuin ϵ, lopeta.
  - Sovella transformaatiota korpukseen: $C_{k+1} = v(C_k)$

- Tulosta tagijono.

Päätettävä: muunnosten soveltamisjärjestys datassa (esim. vasemmalta oikealle) ja käytetäänkö välitöntä muuntamista vai viivästettyä. Jos välitöntä, muunnosten soveltamisjärjestyksellä on väliä.

Soveltaminen ohjaamattomassa oppimistilanteessa

Tilanne: ei tagattua dataa, mutta tunnetaan joka sanalle sallitut tagit (sanakirjasta).

Huom: Useimmilla sanoilla vain yksi sallittu tagi, eli useimpien sanojen tagi tiedetään ennalta. Vain osa tageista epäselviä.

Periaate: Käytetään samassa kontekstissa esiintyvien, tagiltaan yksiselitteisten sanojen tagijakaumaa epäselvän tagin ennustamiseen.

Transformaation hyvyys lasketaan siten, että kuvitellaan tunnetut, yksikäsitteis sanojen luokat tuntemattomiksi ja sovelletaan transformaatiota niiden luokitteluun. Pienimmän osuuden virheluokituksia aiheuttava transformaatio on

paras.

Esimerkki. 'The **can** is open' AT _ IS
Kontekstissa 'AT _ IS' olevat sanaluokaltaan yksikäsitteiset sanat ovat (lähes) aina substantiiveja, eivät verbejä. → 'can' tagataan verbiksi.

Tämänkaltainen 'ohjaamaton' soveltaminen: 95,6% (Brill, 1995)

Hyvä puoli: ei juuri ylioppimista, toisin kuin HMM:llä.

[Huomautus: HMM:illäkin voidaan periaatteessa välttää ylioppimista soveltamalla mallin rakenteen optimoimista (parametrien karsimista), jos käytetään täyttä Bayeslaista estimointia, esim. enseble-oppimista (variaatioanalyysiä).
]

Haaste: potentiaalisia transformaatioita (erityisesti herätekonteksteja) hyvin suuri joukko.

10.8 Yhteys muihin menetelmiin

Päätöspuut (Decision Trees)

Labeloidaan kaikki puun haaraan kytkeytyvä data k.o. haaran majority-luokalla.

Puuta haaroitetaan sen mukaan että alemman tason datan luokittelussa tehtäisiin mahdollisimman pieni osuus virheitä.

Huono puoli: potentiaalisia sääntöjä hyvin suuri joukko. Hakua sääntöjoukossa voidaan kuitenkin nopeuttaa.

Pääasiallinen ero muunnostaggaukseen: Päätöspuu jakaa datajoukon osiin, ja myöhemmät transformaatiot operoivat ainoastaan ko. haaran osadatalla.

Muunnostaggauksessa datan osajoukko, johon muunnosta sovelletaan, valitaan heräteympäristön perusteella koko datasta.

Eroja aitoon probabilistiseen mallinnukseen verrattuna

Ei käytettävissä prob. mallinnuksen kaikkea välineistöä.

Esim. laajentaminen tilanteeseen, jossa tuotetaan yhden parhaan tagin sijaan k parasta, ei ole yhtä suoraviivaista

Prioritiedon huomioiminen:

Muunnostaggaus pystyy helposti huomioimaan heräteympäristojen muodossa annetun prioritiedon.

Ei kykene hyödyntämään eri luokkien prioritn:iä, ainoastaan tiedon sanan todennäköisimmästä luokasta (initialisoinnissa).

Muita eroja:

Joustavuus: Muunnostaggauksessa hyödynnetään hyvin joustavaa kokoelmaa potentiaalisia vaikuttavia tekijöitä (piirteitä) kussakin vaiheessa ja kullekin transformaatiolle.

Ymmärrettävyys: Binääriset säännöt ovat yksinkertaisempia ihmiselle ymmärtä
Kuitenkin sekvenssistä jonkin yksittäisen säännön muuttamisen vaikutusta on
vaikea ennustaa johtuen sääntöjen välisestä interaktiosta.

Yhteys automaatteihin

Taggerin sääntöjen oppiminen tapahtuu kvantitatiivisesti.

Valmis muunnostaggeri voidaan kuitenkin muuttaa deterministiseksi FST:ksi
ja saada sille näin tehokas toteutus.

10.9 Taggauksen evaluoinnista

Taggausprosentit englannille tyypillisesti luokkaa 95-97%, kun raportoidaan kaikkien sanojen yli (ei vain moniselitteisten).

Tulokseen vaikuttavat mm. seuraavat tekijät:

- Datan määrä (isommalla datalla tulee parempia tuloksia)
- Tagijoukko: yleensä, mitä enemmän tageja, sen vaikeampi ongelma. Toisaalta, jos käytetän joillekin sanoille ihka omia tageja (esim. 'to'= TO) ei näitä voi tagata väärin.
- Erot opetusdatan, sanakirjan ja sovelluksen (testidatan) välillä
- Tuntemattomien sanojen tn: vaikuttaa suuresti onnistumisprosenttiin.

Eri (hyvien) menetelmien väliset erot ovat puolen prosentin luokkaa. Yllämainit mm. aineistokohtaisten ominaisuuksien aiheuttamat erot ovat usein paljon suurempia.

Kuitenkin pienikin sanakohtainen parannus menetelmässä aiheuttaa merkittävän lausekohtaisen parannuksen, mikä on taggausta hyödyntävien sovellusten kannalta relevanttia.

Eräs parhaista tunnetuista taggereista englannille: Helsingin Yliopistossa kehitetty EngCG (Voutilainen, Tapanainen et.al): ihmisen kehittämä sääntöpohja taggeri (asiantuntijajärjestelmä taggaukseen). 99% tarkkuus. Muistuttaa transformaatitaggausta paitsi että sääntöjen valinnan tekee ihmisasiantuntija.

Taggausten sovelluksista

- Yllättävän vähän julkaisuja taggauksen sovelluksista.
- Useissa sovelluksissa tarvitaan lisäksi *osittaisjäsennys*
- Information extraction: taggausta jossa syntaktisten kategorioiden sijaan pyritään tunnistamaan (tiettyjä) semanttisia kategorioita (esim. erisnimet). Syntaktisesta kategoriasta voi olla apua.

- Tiedonhaun indeksointitermien valinta tai painotus (sekä taggaus että osittaisjäsennys)

- Kysymyksiin vastaus: 'Who killed president Kennedy' , vastaus 'Oswald' edellyttää että ymmärretään mikä asiaan liittyvä tieto (esim. aika, paikka, vai henkilö) kysyjälle pitäisi kertoa. Lisäksi on pystyttävä eristämään oikea tieto dokumenteista jotka aiheeseen liittyvät. Molemmissa voi olla hyötyä taggauksesta.

- Negatiivinen tulos taggauksen hyödyllisyyden kannalta: parhaat sanatietoa hyödyntävät prob.jäsentimet toimivat paremmin lähtemällä taggaamattomasta tekstistä ja taggaamalla sitä itse kuin hyödyntämällä valmista taggausta.

11. Probabilistinen jäsentäminen ja PCFG:t

PCFG=Probabilistic Context Free Grammar
Todennäköisyyksiin perustuva yhteysvapaa kielioppi

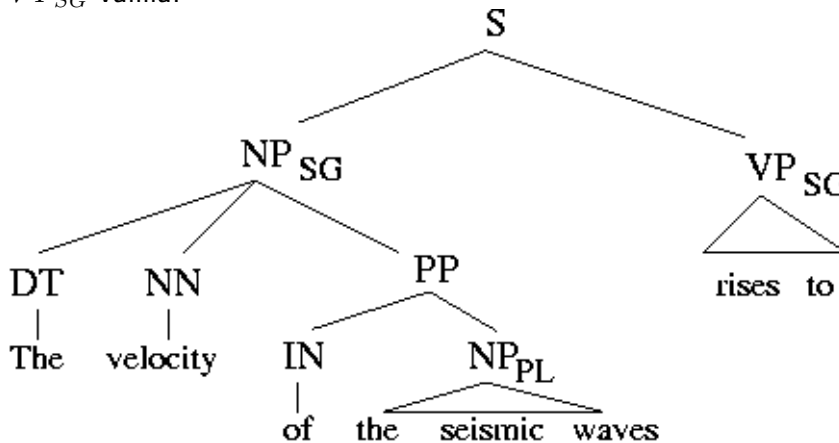
Motivaatio

- Tähän asti kielen säännönmukaisuuksia kuvattu vain sanatasolla (n-grammitmallit) 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').
- Keskeinen havainto: yksittäisten sanojen sijaan riipuvuudet voidaan

ehkä paremmin kuvata suurempien kokonaisuuksien välillä → oletetaan hierarkkinen rakenne myös lauseiden tai virkkeiden sisällä. (Vastaava hierarkkisuus yleisemmällä tasolla itsestäänselvä: dokumentti, kappale, lause, sana, kirjain, veto).

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

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



11.1 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.