Katsaus tilastolliseen puheentunnistukseen

Tero Tarvainen

12. huhtikuuta 2002

Älykkäiden järjestelmien tutkimusseminaari
Tietojenkäsittelytieteen laitos
Helsingin Yliopisto

1 Johdanto

Ensimmäinen puheentunnistuksen virstanpylväs oli lelukoira nimeltä Rex. Rex oli koirankopissa asustava koira, joka kuulleessaan nimensä "Rex" tuli kopista esiin. Lelu valmistettiin 1920-luvulla. Varsinainen puheentunnistukseen liittyvä tutkimus alkoi kuitenkin 1950-luvulla. Tutkimuksen alkutaipaleella kehitetyt järjestelmät pystyivät tunnistamaan yksittäisiä sanoja, kuten numerot nollasta yhdeksään, mutta olivat erittäin riippuvaisia sekä puhujasta, että ympäristöstä. Nykyisten järjestelmien sanastojen koko on kasvanut jo useisiin tuhansiin ja ne ovat tulleet puhuja riippumattomiksi.

Suurisanastoiset puheentunnistusjärjestelmät (Large Vocabulary Speech Recognition systems) perustuvat yleisesti tilastolliseen hahmontunnistukseen. Tilastollisen puheentunnistuksen pohja luotiin jo 1970-luvulla IBM:llä Bakerin, Jelinekin ja heidän kolleegoidensa toimesta. Heidän kehittämänsä menetelmät ovat muuttuneet vain vähäisesti sittemmin.

Tämä kirjoitelma tarkastelee yleisesti käytettyä tapaa toteuttaa suurisanastoista jatkuvaa puheentunnistusta (Large Vocabulary Continuos Speech Recognition, LVCSR). Esitetyt menetelmät ja mallit ovat HTK (Hidden Markov Toolkit) [You02] järjestelmästä, mutta vastaavat menetelmät ovat käytössä myös muissa alan huippujärjestelmissä. Kirjoitelman tarkoitus on tarjota lukijalle yleinen katsaus tilastolliseen puheentunnistukseen, sen ongelmiin ja siihen, miten osa siihen liittyvistä ongelmista on ratkaistu.

Kirjoitelma etenee seuraavasti: Toisessa luvussa esitetään yleiskatsaus tilastolliseen puheentunnistukseen tarkoituksena hahmottaa ongelmaa ja sitä, miten ongelma on jaoteltu eri osiin. Luvussa kolme esitetään lyhyesti signaalinesikäsittelyä. Aiheen vaativuuden ja suuruuden takia, signaalinkäsittely jätetään vaille tarkempaa tarkastelua. Luvussa neljä keskitytään Markovin piilomallien käyttöön akustisessa mallinnuksessa. Kielen mallintamista tarkastellaan luvussa viisi ja luvussa kuusi keskitytään itse puheentunnistamiseen, eli siihen miten eri osat mahdollistavat yhteistyöllä puheentunnistuksen. Lopuksi asiasta esitetään vielä yhteenveto.

2 Yleiskatsaus tilastolliseen puheentunnistukseen

Puheentunnistusjärjestelmä saa syötteenään puhutun lausahduksen digitaalisen esityksen eli signaalin. Tämä signaali muutetaan sekvenssiksi akustisia vektoreita, $Y = y_1, y_2, ..., y_T$. Jokainen näistä akustisista vektoreista sisältää kompaktin esityksen alkuperäisen aaltomuodon lyhyestä aikavälistä, yleisesti 10 ms. Esimerkiksi tyypillinen kymmenen sanan lausahdus kestää noin kolme sekuntia, joten akustisia vektoreita generoidaan $T = 300$ kappaletta.

Puheentunnistusjärjestelmän tehtävänä on löytää todennäköisin sanajono $W = w_1, w_2, ..., w_n$, jota annettu akustisien vektoreiden jono Y kuvaa. Täytyy siis etsiä sellainen sanajono W^* jolle pätee:

$$W^* = argmax_W P(W|Y) = argmax_W \frac{P(W)P(Y|W)}{P(Y)}, \tag{1}$$

jossa jälkimmäinen muoto saadaan suoraan Bayesin kaavasta. Todennäköisimmän sanajonon W löytäminen vaatii sanajonon, joka maksimoi tulon $P(W)P(Y|W)$, löytämisen. Miksi yhtälö halutaan sitten kirjoittaa tässä muodossa? Tämä siksi, että todennäköisyyttä $P(W|Y)$ ei voida laskea suoraan, mutta $P(W)$ ja $P(Y|W)$ voidaan. Ensimmäinen tulon termi $P(W)$ on *a priori* todennäköisyys sanajonon W esiintymiselle. Tämä todennäköisyys on täysin riippumaton havaitusta aaltomuodosta ja voidaan määritellä suoraan kielimallin avulla. Tulon toinen termi $P(Y|W)$ on todennäköisyys akustisien vektoreiden Y havaitsemiselle, kun on annettuna sanajono W. Tämä todennäköisyys voidaan määritellä akustisen mallin avulla.

3 Signaalin mallintaminen

Ennen kuin puhuttu lausahdus voidaan tunnistaa, se täytyy muuttaa piirrevektori jonoksi $Y = y_1, ..., y_t$. Ennen varsinaista piirrevektorien muodostamista, havaittu signaali puhdistetaan mahdollisista hälyäänistä. Signaalin puheelle kriittisiä taajuusalueita voidaan myös korostaa.

Piirrevektoreiden muodostaminen aloitetaan jakamalla havaittu signaali ikkunoihin, joiden kesto on yleensä noin 25 millisekuntia ja jotka ovat kymmenen millisekunnin päässä toisistaan, siis osittain päällekäisiä. Jokaiselle ikkunalle lasketaan sitä kuvaavia piirteitä, jotka sijoitetaan ikkunan piirrevektoriin. Laskettavia piirteitä ovat esimerkiksi ikkunan taajuusspektri, energian keskiarvo, suurimman amplitudin ja pienimmän suhde ja muita signaalia kuvaavia piirteitä.

Lopputuloksena ikkunalle saadaan piirrevektori, jonka dimensio on noin neljänkymmenen luokkaa (HTK järjestelmässä käytetty dimensio on 39). Käytetty ikkunointi perustuu siihen oletukseen, että lyhyellä aikavälillä tarkasteltuna puhesignaali on muuttumatonta.

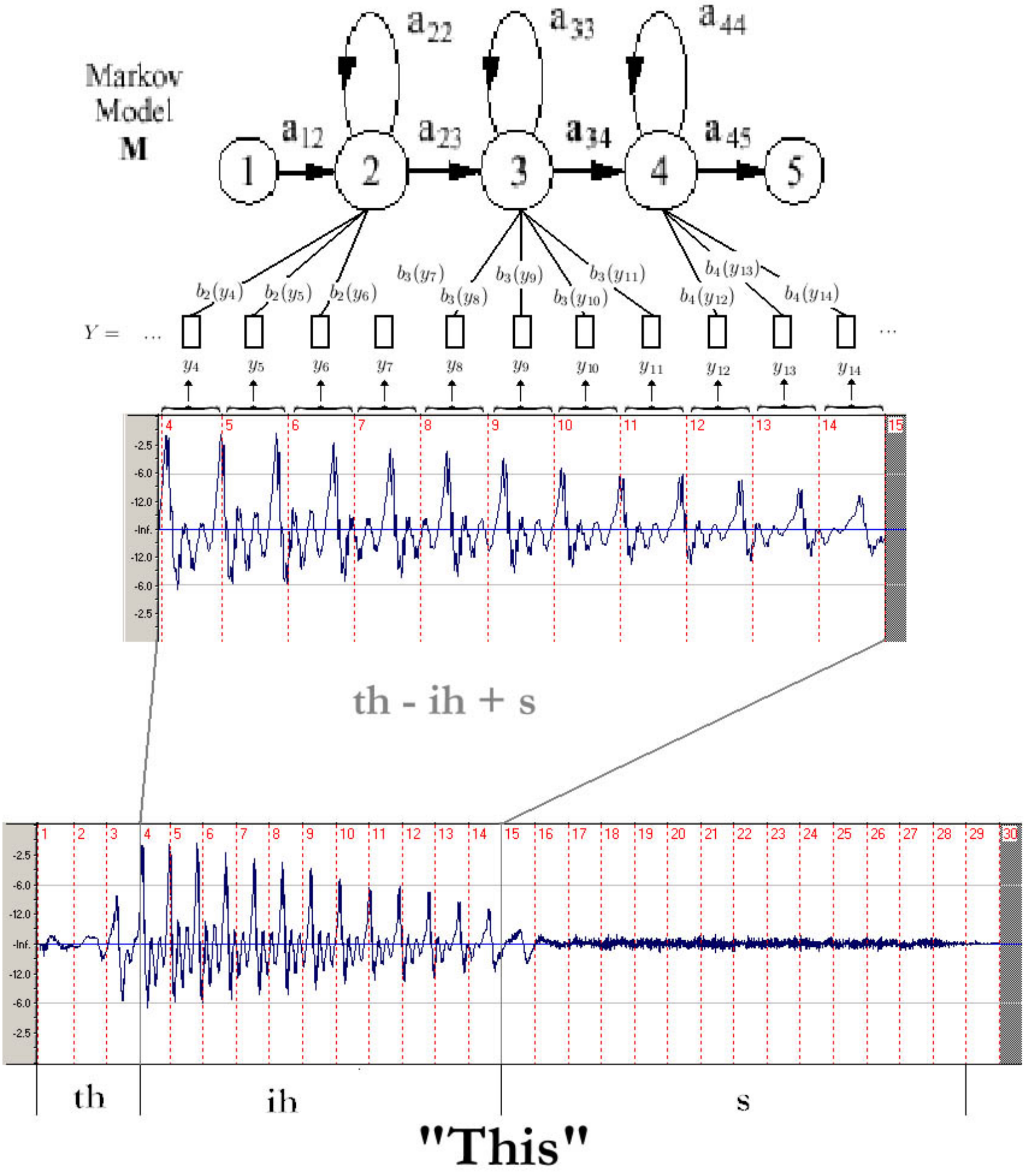
Piirrevektoreiden muodostaminen on puheentunnistuksen kannalta erittäin tärkeää. Aihetta ei kuitenkaan tässä kirjoitelmassa käsitellä tämän enempää, sillä aihe liittyy enemmän digitaaliseen signaalin käsittelyyn. Tarkempaa tietoa asiasta löytyy esimerkiksi Mantha et al. artikkelista *"Implementation and Analysis of Speech Recognition Front-Ends"* [Man99].

4 Akustinen mallinnus

Akustisen mallinnuksen tarkoituksena on tarjota tapa laskea todennäköisyys vektorijonolle Y, kun sanajono W on annettu, eli laskea yhtälön (1) termi $P(Y|W)$.

Puheentunnistusjärjestelmä, jonka sanasto on suuri, perustuu yleisesti äänteiden (phone) tunnistamiseen. Tämä sen takia, että äänteitä on luonnollisesti huomattavasti vähemmän kuin sanoja. Englanninkielessä vain n. 45 erilaista.

Yksittäisen äänteen tunnistamiseen käytetään Markovin piilomallia (Hidden Markov Model HMM) [Rab89]. Mallin topologiana käytetään yleisesti yksinkertaista vasen-oikea mallia (left-right model), jossa jokaisesta tilasta on tilasiirtymät vain itseensä ja seuraavaan tilaan. Kuvassa 1 olevalla HMM:llä on viisi tilaa, joista kolme on emittoivaa. Intuitio emittoiville tiloille on seuraava: Ensimmäinen emittoiva tila kuvastaa äänteen alkamista, toinen sen kestoa ja kolmas lopuketta. Tällaista kolmen emittoivan tilan mallia käytetään esim. HTK järjestelmässä [You02] ja se soveltuu hyvin kontekstista riippuvaisten äänteiden kuvaukseen.



Kuva 1: Esimerkkitopologia Markovin piilomallille

Malli sisältää tilasiirtymät $\{a_{ij}\}$ (todennäköisyys, että tilasta i siirrytään tilaan

j) ja emissiojakaumat $\{b_j(y_t)\}$ (todennäköisyys, että tila j emittoi akustisen vektorin y_t). Ensimmäinen ja viimeinen tila ovat siitä poikkeavia, ettei niihin liity tilasiirtymä- tai emittointitodennäköisyyksiä. Näitä tiloja käytetään yhdistelemään malleja keskenään, jolloin saadaan muodostettua kokonaisia sanoja ja lausahduksia.

HMM:ää voidaan pitää äärellisenä tilakoneena, joka jokaisena ajanhetkenä vaihtaa tilaa ja jokaisella ajan hetkellä t tilassa j emittoi akustisen vektorin y_t todennäköisyydellä $b_j(y_t)$. Todennäköisyys sille, että malli generoi vektorijonon Y, kun sen tilajono $X = x(1), x(2), ..., x(T)$ on tunnettu on:

$$P(Y, X|M) = a_{x(0)x(1)} \prod_{i}^{N} b_{x(t)}(y_t) a_{x(t)x(t+1)}, \tag{2}$$

missä M sisältää HMM:n tilasiirtymä- ja emittointitodennäköisyydet. Käytännössä tunnistamisessa käytetään yhtälön (2) logaritmistä muotoa

$$\log P(Y, X|M) = \sum_{t=0}^{T} \log a_{x(t)x(t+1)} + \sum_{t=1}^{T} \log b_{x(t)}(y_t), \tag{3}$$

jolloin logaritminen todennäköisyys saadaan helposti summaamalla tilajonon X mukaisten tilasiirtymien logaritmiset todennäköisyydet ja vektorijonon Y mukaisten emissioiden logaritmiset todennäköisyydet. Yleisesti tilasiirtymät ovat diskreettejä todennäköisyyksiä ja emissotodennäköisyydet normaalijakautuneita jatkuvia jakaumia.

4.1 Äänteiden kontekstiriippuvainen mallinnus

Oletus, että vain yksi HMM riittäisi kuvaamaan yhden äänteen, osoittautuu käytännössä vääräksi. Käytännössä äänteet ovat kontekstiriippuvaisia. Hyvän foneettisen erottelun saavuttamiseksi tarvitaankin yksi HMM jokaisen äänteen eri kontekstille. Yksinkertainen ja yleisin tapa tehdä tämä on käyttää ns. *triphoneja*. Tällöin jokainen äänne kuvataan kontekstissa, jossa on mainittu edellä oleva äänne ja jäljessä tuleva äänne. Tähän käytetty notaatio on seuraavanlainen: $a - b + c$ tarkoittaa, että äänne b tulee äänteen a jälkeen ja ennen c:tä.

Triphonejen käytössä ilmeneekin uusi ongelma. Esimerkiksi englanninkielessä erilaisia äänteitä on 45, joten mahdollisia triphoneja on tällöin $45^3 = 91125$ kappaletta (joista tosin ”vain” n. 60000 on kielellisesti mahdollista). Jotta kaikki mallit saataisiin opetettua, tarvittaisiin erittäin suuri määrä opetusaineistoa. Tälläisen opetusaineiston kokoaminen onkin käytännössä melko mahdotonta.

Yksi ratkaisu ongelmaan on etsiä eri malleista samankaltaisia tiloja ja sitoa niitä keskenään. Intuitio sitomiselle on seuraava: Oletetaan, että malli 1 kuvaa äännettä $t - \mathbf{iy} - n$ ja malli 2 kuvaa äännettä $t - \mathbf{iy} - s$. Molemmat äänteet seuraavat konsonanttia (äännettä t), joten voisi ajatella, että vaikuttaako aikaisempi äänne jollakin tietyllä tavalla seuraavan äänteen muodostumiseen. Ja käy ilmi, että tämän tyylisiä riippuvuuksia todellakin löytyy. Ongelmaksi muodostuu siten niiden kriteerien etsiminen, joiden mukaan tiloja voidaan sitoa.

Ratkaisuksi tähän on esitetty puupohjaista tilojen sitomista (*Tree-Based State Tying*) [YOW94]. Ideana on konstruoida foneettinen päätöspuu (binääripuu), jonka jokainen solmu sisältää erityisen kysymyksen. Yksi päätöspuu konstruoidaan jokaisen äänteen jokaiselle tilalle (äänteitä 45 jokaisella 3 emittoivaa tilaa = 135 puuta). Puun juureen kootaan kaikki ko. äänteen mahdollisten triphonejen tietyt tilat joukoksi (esimerkiksi jokaisen ensimmäinen emittoiva tila). Joukko jaetaan jokaisen solmun kohdalla kahtia riippuen solmun kysymyksen vastauksesta. Kaikki tilat, jotka vastasivat 'Kyllä', jaetaan vasemmanpuoleiseen haaraan ja kaikki 'Ei' vastanneet oikeanpuoleiseen haaraan. Tätä jatketaan niin kauan kunnes saavutaan lehtiin. Tilat, jotka ovat samassa lehdessä sidotaan keskenään. Kysymykset solmuissa ovat muotoa "Onko aikaisempi äänne nasaalinen?"ja "Seuraako äännettä vokaali?".

Se, mikä kysymys on missäkin solmussa valitaan seuraavasti: Solmuun ryhmiteltyjen tilojen emissiotodennäköisyysjakaumista muodostetaan keskiarvo. Jokainen tila asetetaan käyttämään tätä keskiarvojakaumaa. Opetustaineiston avulla näille malleille lasketaan logaritminen todennäköisyys. Joukko jaetaan kahtia. Lasketaan logaritminen todennäköisyys molemmille joukoille siten, että tilat käyttävät joukon sisäistä emissiotodennäköisyysjakauman keskiarvoa jakaumanaan. Kysymys, joka kasvattaa logaritmista todennäköisyyttä eniten, valitaan solmun kysymykseksi. Näin jatketaan kunnes logaritmisen todennäköisyyden arvo ei kasva ennalta määrättyä raja-arvoa enempää. Toisin sanoen, kaikki äänteeseen liittyvät tilat klusteroidaan pienemmiksi joukoiksi. Jokaiselle joukolle valitaan joukkoa kuvaava emissiojakauma. Joukon alkiot sidotaan käyttämään tätä yhteistä emissiojakaumaa omana jakaumanaan.

Näin generoidun päätöspuun avulla voidaan luokitella myös sellaiset triphonet, joita ei ollut opetusdatassa. Luokittelu tapahtuu siten, että ko. triphonen tilat käytetään vastaavien päätöspuiden läpi. Jokainen triphonen tila asettuu johonkin lehtisolmuun, jolloin tilalle voidaan käyttää ko. lehtisolmuun (klusteriin) liittyvään emissiojakaumaa.

5 Kielen mallinnus

Kielen mallinnuksen tarkoitus on mahdollistaa tapa laskea todennäköisyys sanajonolle W, eli laskea yhtälön (1) termi $P(W)$. Toisin sanoen haluaisimme tietää todennäköisyyden sille, että sana w_k esiintyy, kun tiedämme että sanat $W_1^{k-1} = w_1, w_2, ..., w_{k-1}$ ovat edeltäneet, eli

$$P(w_k|w_{k-1}, w_{k-2}, ..., w_1) \tag{4}$$

Tämän avulla voidaan koko sanajonon todennäköisyys laskea kaavalla:

$$P(W) = \prod_{k=1}^{K} P(w_k|w_{k-1}, w_{k-2}, ..., w_1) \tag{5}$$

Todennäköisyys (4) saadaan suoraan opetusdatasta laskemalla vain kuinka monta kertaa koko sanajono esiintyy datassa, ja jakamalla se sellaisten sanajonojen lukumäärällä, jotka muuten ovat samoja paitsi viimeisen sanan osalta.

Koko sanahistorian muistaminen on kuitenkin turhan työlästä ja käytännössä rajoitutaakin tarkastelemaan sanan lähempää ympäristöä, esimerkiksi vain kahta aikaisempaa sanaa. Kielissä, jossa sanojen järjestys on tärkeä ja joissa vahvimmat asiaa kuvaavat sanat ovat lähekkäin, tälläinen rajoitettu historian tarkasteleminen tuntuu jokseenkin mielekkäältä. Tällöin sanajonon todennäköisyys (5) supistuu muotoon,

$$P(W) = \prod_{k=1}^{K} P(w_k|w_{k-1}, w_{k-2}, ..., w_{k-n+1}), \tag{6}$$

jossa n kuvaa muistettavan historian kokoa. Esimerkiksi n:n arvolla 3 yksittäisen sanan esiintymistodennäköisyydeksi saadaan $P(w_5|w_4, w_3)$. Käytännössä n:lle käytetään arvoa väliltä $[2-4]$. Tätä mallia kutsutaan *N-Gram* kielimalliksi [Bro90]. Yksittäiselle *N-gramille* saadaan todennäköisyys laskemalla opetusdatasta sen esiintymiskerrat ja jakamalle se niiden *N-gramien* esiintymiskerroilla, jotka sisältävät saman ympäristön, mutta eri loppusanan.

Vaikka *N-Gram* kielimalli pienentääkin tarkasteltavien sanojonojen lukumäärää merkittävästi, niin esimerkiksi 10000 sanan sanastolla ja 3-*Gram* kielimallilla (n:n arvolla 3) mahdollisia tarkasteltavia sanajonoja (*trigrameja*) on 10000^3. Tämä on huomattavan suuri luku ja 10000 sanan sanasto on vielä melko pieni. Tälläisen mallin opettaminen vaatisi huomattavan määrän opetusaineistoa ja aineiston niukkuus onkin yleisesti ongelma. Myös mallin tallentaminen vaatisi kohtalaisen ison tietovaraston.

Yksi tapa ratkaista ongelma on luokitella samankaltaiset sanat ja laskea esiintymistodennäköisyydet käyttäen luokkia [Bro90]. Intuitio sille, miksi näin voi tehdä on se, että selvästikin jotkut sanat ovat jossain mielessä samankaltaisia. Esimerkiksi sanat *'hauskaa'* ja *'kivaa'* ovat merkityksellisesti samankaltaisia lauseessa *'puheentunnistus on hauskaa / kivaa'*. Luokittelulla voidaan laskea *N-Gram*:lle todennäköisyys:

$$P(w_k|w_{k-1}, w_{k-2}, ..., w_{k-n+1}) = P(w_k|c_k)P(c_k|c_{k-1}, c_{k-2}, ..., c_{k-n+1}), \tag{7}$$

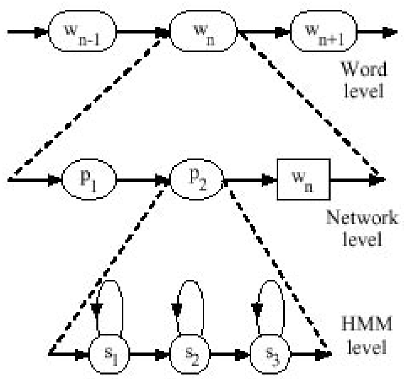
missä c_k:lla tarkoitetaan sitä luokkaa, johon k:s sana kuuluu. Todennäköisyys $P(c_k|c_{k-1}, c_{k-2}, ..., c_{k-n+1})$, määrittelee sen, millä todennäköisyydellä luokan c_k sana tulee, kun sitä edeltää luokkien $c_{k-1}, c_{k-2}, ..., c_{k-n+1}$ sanat. Painokerroin $P(w_k|c_k)$ kuvaa sanan w_k kuulumista luokkaan c_k.

Kiintoisaksi kysymykseksi nouseekin, miten opetusaineiston avulla voidaan saavuttaa hyvä klusterointi sanaston sanoille. Sanojen klusterointiin on kehitelty erilaisia menetelmiä [Bro90, DPL94].

6 Puheen tulkinta

Puheentunnistus (dekoodaus) tapahtuu siis etsimällä sellainen sanajono W^*, joka maksimoi yhtälön (1). Kuten jo aikaisemmin todettiin, akustisen mallin avulla voidaan laskea kuinka hyvin havaittu signaali sopii eri sanajonoille ja kielimallin avulla voidaan laskea sanajonon todennäköisyys.

Puheentunnistusjärjestelmä saa syötteenä tuntemattoman signaalin, josta se generoi T kappaletta akustisia vektoreita. Tällöin jokainen polku, joka kulkee T määrän emittoivien HMM tilojen läpi, on potentiaalien tunnistushypoteesi. Kaikki mahdolliset sanajonot saadaan esimerkiksi konstruoimalla puu, jonka juuren lapsina ovat kaikki mahdolliset sanat, jotka voivat aloittaa lausahduksen. Näiden sanojen lapsina ovat kaikki mahdolliset sanat, jotka voivat jatkaa lausahdusta ja näillä taas seuraavat sanat. Jokainen puun sana sisältää yhden tai useamman äännejonon (yksi äännejono jokaiselle eri ääntämiselle). Jokainen äänne sisältää sitä kuvaavan HMM:n. Tämä hierarkia on esitetty kuvassa 2. Kuvassa äännejonon perässä oleva laatikko tarkoittaa erityistä sananloppumissolmua.



Kuva 2: Tunnistuksen tasot [You02]

Jokaiselle poluista voidaan laskea logaritminen todennäköisyys (log likelihood).

Tämä tapahtuu siten, että HMM:n sisällä logaritmiset todennäköisyydet lasketaan summaamalla ko. tilasiirtymien- ja emissiotodennäköisyyksien logaritmit yhteen ja sanojen välillä summaamalla kielimallin määrittelemien todennäköisyyksien logaritmit yhteen. Syy logaritmisen todennäköisyyden käyttöön on se, että sen avulla voidaan laskea pistemäärä jokaiselle solmulle, solmuun kuljetun polun mukaan. Dekooderin työ on etsiä ne polut, joiden logaritminen todennäköisyys (pistemäärä) on suurin. Selvästikin, kaikkien mahdollisten sanajonojen läpikäyminen on käytännössä mahdotonta, joten etsintäavaruutta on pakko rajoittaa jollakin tavalla.

HTK järjestelmässä [You02] etsintä suoritetaan leveyssuuntaisella läpikäynnillä ja etsintäavaruutta rajoitetaan menetelmällä nimeltä *Beam-Search*. Etsinnän perusidea on seuraavanlainen: Käytetään erityisiä merkkejä (tokens), jotka kulkevat mahdollisten polkujen läpi. Merkit sisältävät tiedon kulkemastaan polusta ja sen hetkisen polun pistemäärän. Joka kerta kun polku haarautuu, ko. merkki kopioidaan jokaiseen haarautumaan. Alussa on vain yksi merkki. Merkki kopioidaan jokaiseen mahdolliseen aloitussanaan (ensimmäisen äänteen HMM:n ensimmäiseen tilaa). Merkin pistemäärään lisätään tilaan liittyvän tilasiirtymätodennäköisyyden logaritmi ja emissiotodennäköisyyden logaritmi (todennäköisyys sille, että ko. tila on emittoinnut akustisen vektorin y_1). Jokaisen aika-askeleen jälkeen kaikki ne merkit joiden pistemäärän erotus parhaimmasta pistemäärästä jää alle tietyn raja-arvon (*Beam-Width*) karsitaan pois. Näin jatketaan eteenpäin. Kun merkki saapuu sananloppu solmuun, sen pistemäärään lisätään kielimallin määrittelemä sanatodennäköisyyden logaritmi. Etsintä loppuu, kun syötevektorijono loppuu. Merkki, joka sisältää parhaimman pistemäärän, kuvaa todennäköisintä lausahdusta. Tälläistä etsintää kutsutaan myös *Viterbi*-dekoodaukseksi (*Beam-Search* karsinnalla).

Jos karsinnassa käytettävä raja-arvo on liian tiukka, voi oikea sanajono karsiutua etsinnän aikana. Liian löysän raja-arvon käyttö johtaa taas siihen, että etsittäviä polkuja on suuri määrä. Raja-arvon asetuksella voidaan siis vaikuttaa etsintänopeuden ja etsintätarkkuuden suhteeseen.

7 Yhteenveto

Esitelty tilastolliseen hahmontunnistukseen perustuva menetelmä on pohjana useissa nykyisissä puheentunnistusjärjestelmissä. Esimerkkinä IBM:n ViaVoice [1].

Esitetty malli perustui erillisten HMM:n yhdistämiseen, jolloin saatiin muodostettua sanoja ja lausahduksia. Tälläinen ns. *beads-on-a-string* malli toimii hyvin artikuloidulle puheelle, mutta huonommin luonnolliselle puheelle, joka sisältää enemmän variaatiota. Käytettyä *N-Gram* mallia voidaan myös kritisoida liian suppeaksi kielenkuvaukseksi. Kielelliset rakenteet, jotka voisivat auttaa tulevan sanan todennäköisyyden määrittämisen, jäävät huomioimatta, mikäli rakenne on hajautunut lauseessa pidemmälle välille [Che97]. Akustisessa mallinnuksessa tehty oletus vierekkäisten piirrevektorien riippumattomuudesta on myös jokseenkin väärä. Näiden heikkouksien eteen on tehty paljon tutkimusta ja joitain parannuksiakin on saavutettu.

Myös muita menetelmiä puheentunnistukseen on tutkittu paljon. Näistä mainittakoon neuroverkkoihin ja dynaamisiin Bayes-verkkoihin perustuvat järjestelmät. Myös erilaisia hybridi-järjestelmiä on kehitelty.

Puheentunnistus on kehittynyt viime vuosina sellaiselle tasolle, että sitä hyödyn-

[1] http://www-3.ibm.com/software/speech/ (tarkistettu 10.4.2002)

tävät tuotteet ovat yleistyneet. Parhaimmat sovellukset yltävät jo kohtalaisiin tuloksiin, mutteivat kykene vielä tulkitsemaan luonnollista puhuttua kieltä.

Viitteet

[You01] Young SJ, Statistical Modelling in Continuous Speech Recognition. *Proc. 17th Conference on Uncertainity in Artificial Intelligence*, Seattle, Yhdysvallat, 2001, 562-571

[Rab89] Rabiner L, A Tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2), 1989, 257-285

[YOW94] Young SJ, Odell JJ, Woodland PC, Tree-Based State Tying for High Accuracy Acoustic Modelling. *Proc. Human Language Technology Workshop*, Plainsboro NJ, Yhdysvallat, 1994, 307-312

[Bro90] Brown P et al., Class-based n-gram models of natural language, *Computational Linguistics*, 18(4), 1992, 467-479

[You02] Young SJ, Kershaw DJ, Odell JJ, Ollason D, Valtchev V, Woodland PC, The HTK Book Version 3.1, Cambridge University, Cambridge, Englanti, 2002.

[Man99] Mantha V, Duncan R, Wu Y, Zhao J, Ganapathiraju A, Picone J, Implementation and Analysis of Speech Recognition Front-Ends, *Proceedings of the IEEE Southeastcon*, Lexington, Kentucky, Yhdysvallat, 1999, 32-35

[DPL94] Dagan I, Pereira FCN, Lee L Similarity-Based Estimation of Word Cooccurrence Probabilities, *Meeting of the Association for Computational Linguistics*, 1994, 272-278

[Che97] Chelba C, A Structured Language Model, *Proceedings of the Thirty Fifth Annual Meeting of the Association for Computational Linguistics*, Somerset, New Jersey, 1997, 498-503