T-61.281 Luonnollisen kielen tilastollinen käsittely

Vastaukset 5, ti 24.2.2004, 8:30-10:00 N-grammikielimallit, Versio 1.1

1. Alla on erään henkilön ja tilaston estimaatit sille, miten todennäköistä on, että alla annetut sanat seuraavat sanoja "tuntumaan jo":

sana	tilasto trig	ihminen trig	ihminen lause
ja	0.00	0.00	0.00
hyvältä	1.00	0.18	0.40
kumisaapas	0.00	0.00	0.00
keväältä	0.00	0.23	0.50
ilman	0.00	0.05	0.05
päihtyneeltä	0.00	0.20	0.00
turhalta	0.00	0.23	0.05
koirineen	0.00	0.00	0.00
öljyiseltä	0.00	0.11	0.00
Turku	0.00	0.00	0.00

Taulukko 1: *Ihminen vs tilasto, trigrammiestimaatit*

Tarkemmalla tukimisilla huomataan, että taulukossa ihmisen antamat trigrammiestimaatit ovat jonkin verran pielessä ja tilastolliset pehmentämättömät trigrammiestimaatit aivan pielessä.

Tilastollisten estimaattien laskuun käytettiin n. 30 miljoonan sanan aineistoa. Tässä aineistossa ei yksikään annetuista taivutetuista trigrammeista esiintynyt kertaakaan. Trigrammit perusmuotoistamalla löydettiin 11 lausetta, joissa esiintyi "tuntua jo hyvä". Estimaatti kaipaa siis selvästi tasoittamista, eikä senkään jälkeen ole kovin luotettava.

Myös esimerkki-ihmisen antamaa estimaattia voi epäillä, aivan mahdollisille lauseille on annettu nollatodennäköisyys, esim. "Kyllä alkaa tuntumaan jo kumisaapas jalassa", lause joka voidaan tokaista vaikka pitkän vaelluksen päätteeksi. Toisaalta annetulla 2 desimaalin tarkkuudella estimaatit lienevät hyviä.

Kun testihenkilölle annettiin koko lause nähtäväksi, saatiin jo varsin laadukkaat estimaatit. Jotta tilastollisesti pystyttäisiin pääsemään samaan tulokseen, tarvitsisi mallin ymmärtää suomen kielen syntaksia (miten sanoja voidaan taivuttaa ja laittaa peräkkäin) sekä myös sanojen semanttista merkitystä ("helmikuu" on lopputalvea, melkein kevättä).

2. a) Suurimman uskottavuuden estimaatit voidaan laskea kaavasta

$$P(w_i|w_{i-1}, w_{i-2}, \dots) = \frac{C(w_i, w_{i-1}, w_{i-2}, \dots)}{C(w_{i-1}, w_{i-2}, \dots)},$$

missä funktio C kertoo opetusjoukossa nähtyjen näytteiden määrän.

Unigrammiestimaatissa unohdetaan riippuvuus kaikista edellisistä sanoista, bigrammiestimaatti riippuu vain edellisestä sanasta ja trigrammiestimaatissa käytetään historiana kahta edellistä sanaa.

Unigrammiestimaatit voidaan siis laskea

$$P(w_i) = \frac{C(w_i)}{C(0)},$$

missä $C(0)$ on opetusjoukon näytteiden lukumäärä. Estimaatit ovat siis samat kummallekin tehtävänannon historialle.

$$\begin{aligned} P(\text{'olla'}) &= \frac{5}{98} = 0.051 \\ P(\text{'leuto'}) &= \frac{1}{98} = 0.001 \\ P(\text{'gorilla'}) &= \frac{0}{98} = 0.000 \end{aligned}$$

Bigrammiestimaatit saadaan ottamalla yksi sana historiasta käyttöön:

$$\begin{aligned} P(w_i|w_{i-1}) &= \frac{C(w_i, w_{i-1})}{C(w_{i-1})} \\ P(\text{'olla'}|\text{'olla'}) &= \frac{0}{5} = 0.000 \\ P(\text{'leuto'}|\text{'olla'}) &= \frac{1}{5} = 0.200 \\ P(\text{'gorilla'}|\text{'olla'}) &= \frac{0}{5} = 0.000 \\ P(\text{'olla'}|\text{'vaikuttaa'}) &= \frac{0}{1} = 0.000 \\ P(\text{'leuto'}|\text{'vaikuttaa'}) &= \frac{0}{1} = 0.000 \\ P(\text{'gorilla'}|\text{'vaikuttaa'}) &= \frac{0}{1} = 0.000 \end{aligned}$$

Huomataan, että tämän mallin mielestä mikään sanayhdistelmä, mitä se ei ole nähnyt ei ole mahdollinen.

b) Laplacen estimaatissa aikaisemmin havaitsemattomille sanoille annetaan hieman todennäköisyyttä. Estimaatti vastaa prioria, että kaikki sanat ovat yhtä todennäköisiä. Käytännössä se lasketaan niin, että kuvitellaan että kaikkia sanoja on jo nähty kerran:

$$P(w_i|w_{i-1}, w_{i-2}, \dots) = \frac{C(w_i, w_{i-1}, w_{i-2}, \dots) + 1}{C(w_{i-1}, w_{i-2}, \dots) + N},$$

missä N on mallin sanaston koko.
Lasketaan siis estimaatit:

$$\begin{aligned} P(\text{'olla'}) &= \frac{5+1}{98+64000} = 9.3 \cdot 10^{-5} \\ P(\text{'leuto'}) &= \frac{1+1}{98+64000} = 3.1 \cdot 10^{-5} \\ P(\text{'gorilla'}) &= \frac{0+1}{98+64000} = 1.6 \cdot 10^{-5} \end{aligned}$$

Bigrammeille:

$$\begin{aligned} P(\text{'olla'}|\text{'olla'}) &= \frac{1}{5+64000} = 1.6 \cdot 10^{-5} \\ P(\text{'leuto'}|\text{'olla'}) &= \frac{1+1}{5+64000} = 3.1 \cdot 10^{-5} \\ P(\text{'gorilla'}|\text{'olla'}) &= \frac{1}{5+64000} = 1.6 \cdot 10^{-5} \\ P(\text{'olla'}|\text{'vaikuttaa'}) &= \frac{1}{1+64000} = 1.6 \cdot 10^{-5} \\ P(\text{'leuto'}|\text{'vaikuttaa'}) &= \frac{1}{1+64000} = 1.6 \cdot 10^{-5} \\ P(\text{'gorilla'}|\text{'vaikuttaa'}) &= \frac{1}{1+64000} = 1.6 \cdot 10^{-5} \end{aligned}$$

Huomataan, että tässä priorioletus, että kaikki sanat ovat yhtä todennäkisiä ohjaa estimaatteja vahvasti, kaikki sanat ovat toisaan mallien mielstä lähes yhtä todennäköisiä.

c) Lidstonen estimaatissa voidaan säätää sitä, kuinka paljon uskotaan siihen, että sanat ovat yhtä todennäköisiä. Siinä kuvitellaan, että sanat on jo nähty λ kertaa ennen opetusaineistoa:

$$P(w_i|w_{i-1}, w_{i-2}, \dots) = \frac{C(w_i, w_{i-1}, w_{i-2}, \dots) + \lambda}{C(w_{i-1}, w_{i-2}, \dots) + \lambda N},$$

Valitaan tässä $\lambda = 0.01$. Lasketaan estimaatit:

$$\begin{aligned} P(\text{'olla'}) &= \frac{5+0.01}{98+0.01 \cdot 64000} = 6.8 \cdot 10^{-3} \\ P(\text{'leuto'}) &= \frac{1+0.01}{738} = 1.4 \cdot 10^{-4} \\ P(\text{'gorilla'}) &= \frac{0.01}{738} = 1.4 \cdot 10^{-5} \end{aligned}$$

Bigrammeille:

$$\begin{aligned}
P(\text{'olla'}|\text{'olla'}) &= \frac{0.01}{645} = 1.6 \cdot 10^{-5} \\
P(\text{'leuto'}|\text{'olla'}) &= \frac{1 + 0.01}{645} = 1.6 \cdot 10^{-3} \\
P(\text{'gorilla'}|\text{'olla'}) &= \frac{0.01}{641} = 1.6 \cdot 10^{-5} \\
P(\text{'olla'}|\text{'vaikuttaa'}) &= \frac{0.01}{641} = 1.6 \cdot 10^{-5} \\
P(\text{'leuto'}|\text{'vaikuttaa'}) &= \frac{0.01}{641} = 1.6 \cdot 10^{-5} \\
P(\text{'gorilla'}|\text{'vaikuttaa'}) &= \frac{0.01}{641} = 1.6 \cdot 10^{-5}
\end{aligned}$$

Tässä tapauksessa opetusdata ohjaa selkeämmin estimaatteja. Sopiva λ voidaan valita laittamalla opetusjoukosta pieni osa sivuun ja testaamalla tällä sivuun laitetulla tekstillä, mikä λ toimii parhaiten.

3. Tarkoituksenamme siis on laskea todennäköisyyksiä nähdyille trigrammeille. Tavallinen suuriman uskottavuuden estimaattihan olisi

$$p(x) = \frac{r}{N} \tag{1}$$

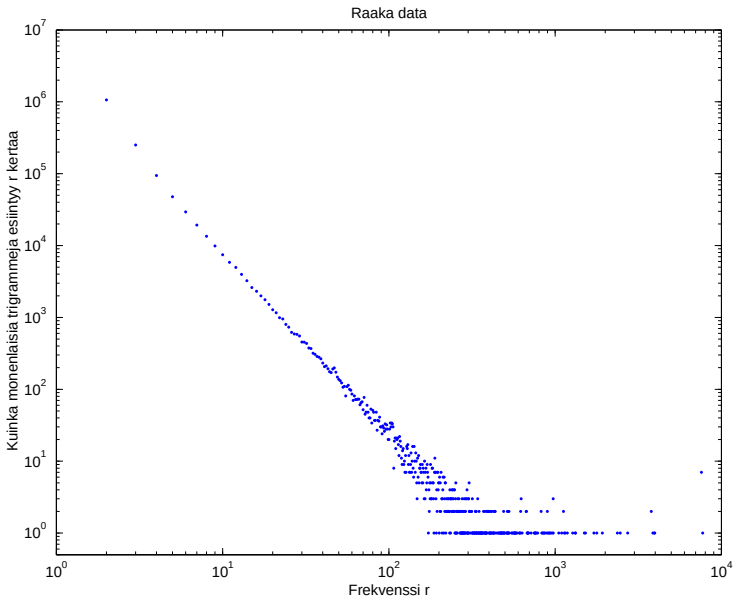
missä r kertoo, kuinka monta kertaa sana esiintyi ja N on kaikkien sanojen lukumäärä. Good-Turing estimaatissa käytetään korjattua estimaattia r^*:

$$r^* = (r + 1)\frac{E(N_{r+1})}{E(N_r)} \tag{2}$$

Good-Turing –tasoitusta voi intuitiivisesti ajatella vaikka niin, että kuvitellaan kaikkia yksiköitä nähdyksi hieman vähemmän kertoja kuin ne oikeasti nähtiin. Eli jos trigrammi nähtiin 10 kertaa, leikitään että se nähtiinkin vain 9.1 kerran. Jos trigrammi nähtiin kerran, leikitään että se nähtiin 0.5 kertaa. Oletetaan että on olemassa N_1 trigrammia, joita ei nähty ja leikitään, että ne nähtiin 0.3 kertaa. Tämä ei tietysti ole matemaattisiseti aivan eksakti määritelmä, mutta helpottaa ehkä tehtävän seuraamista.

Good-Turing –tasoituksen laskeminen aloitetaan taulukoimalla, kuinka monta kertaa eri trigrammia on nähty r kertaa (esim. aineistossa oli 7462 trigrammia, jotka kaikki esiintyivät 10 kertaa). Tästä taulukosta on piiretty kuvaaja 1.

Huomataan, että tähän käyrään olisi helppo sijoittaa suora viiva, paitsi että suuremmilla frekvensseillä tapahtuu jotain kummaa: trigrammeja, joita on esiintynyt vaikkapa 500 kertaa on joko 0 tai 1 kappale. Eli lopussa ei ole enää tasaista käyrää,



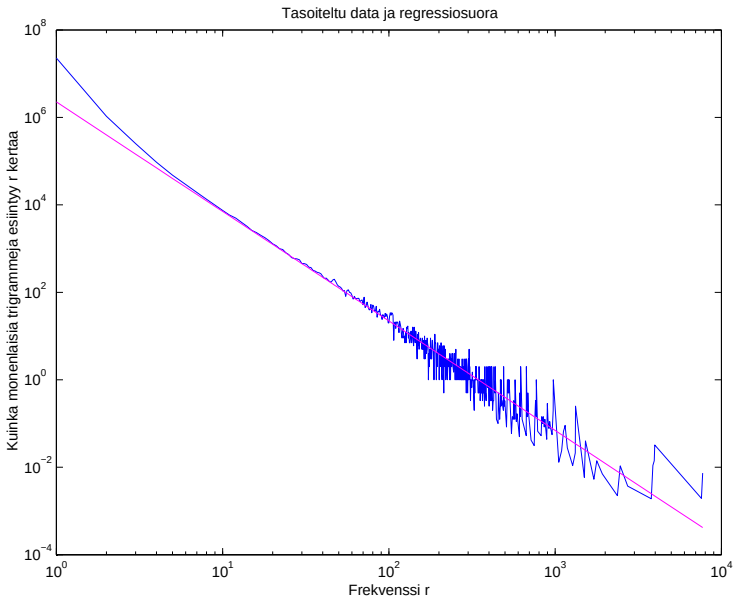
Kuva 1: *X-akseli: esiintymisfrekvenssi. Y-akseli: Kuinka monta trigrammia on esiintynyt r kertaa.*

vaan vain diskreettejä arvoja 0 ja 1. Kokeillaan tasoitella käyrän loppupäätä levittämällä todennäköisyysmassaa tasaisesti koko ympäristöön. Esim. jos trigrammi on esiintynyt 510 kertaa, mutta seuraavaksi yleisin trigrammi on esiintynyt 514 kertaa, jaetaan tuo 1 koko välille, eli kaikille frekvensseille 510-514 tulee arvoksi $\frac{1}{514-510}$. Katsotaan, miltä kuvaaja näyttää tämän jälkeen. Kuvassa 2 nähdään tasoitettu data ja siihen sovitettu suora. Suora sovitettiin kummankin muuttujan logaritmeihin, jolloin se saatiin kauniisti myötäilemään datan muotoa.

Matalat r:n arvot ovat paremmin arvioidut koska niihin meillä on ollu paljon dataa. Käytetään siis niiden arvioina suoraan taulukossa olleita arvoja ja katsotaan korkeat r:n arvot suoraan käyrältä. Tässä tehtävässä päätin käyttää suoralta luettuja arvoja, kun $r > 15$.

Vielä pitäisi antaa jonkinverran todennäköisyysmassa trigrammeille, joita ei ole vielä nähty. Good-Turing estimaatissa näille annetaan yhteensä $\frac{N_1}{N}$ todennäköisyyttä. Tämä todennäköisyys voitaisiin jakaa vaikkapa aineistosta opetetulle bigrammimallille. Nyt, jos trigrammimalli ei osaa antaa sanalle todennäköisyyttä, voidaan tätä todennäköisyyttä kysyä bigrammimallilta. Tuntemattomille bigrammeille jäävä todennäköisyysmassa voitaisiin taas puolestaan jakaa unigrammimallille. Tuntemattomille unigrammeille jäävästä todennäköisyydestä voidaan vain todeta, että tässä on todennäköisyys, että tulee vastaan sana, jota malli ei tunne. Tällainen perääntyvä (back-off) kielimalli on käytössä esim. lähes kaikissa suuren sanaston puheentunnistimissa. Tässä esitettiin vain perusidea perääntyvien kielimallejen estimoinnille. Käytännössä se ei ole aivan näin suoraviivaista.

Kun nyt laskemme korjatulla r^*:llä kaavan 1 mukaan todennäköisyydet saamme mel-



Kuva 2: *Tasoitettu kuva ja logaritmisella asteikolla sovitettu suora.*

r	todennäköisyys
1	$1.9 \cdot 10^{-10}$
2	$3.3 \cdot 10^{-9}$
3	$2.5 \cdot 10^{-8}$
10	$2.7 \cdot 10^{-7}$
50	$1.7 \cdot 10^{-6}$
100	$3.5 \cdot 10^{-6}$
573	$2.0 \cdot 10^{-5}$
1327	$4.7 \cdot 10^{-5}$

Taulukko 2: *Good-Turing todennäköisyysestimaatit*

ko hyvät estimaatit eri sanojen todennäköisyyksille. Taulukkoon 2 on merkitty muutamalle eri r:lle todennäköisyydet. Opetetun mallin mielestä 81% todennäköisyydestä on tuntemattomilla trigrammeilla. Tämä on suomen kielelle melko uskottavan kuuloinen tulos, sillä suomen kielen sanamäärä on niin suuri, että kielelle on käytännössä mahdotonta tehdä kattavaa trigrammimallia. Sivuhuomatuksena mainittakoon, että trigrammimallinnus voi soveltaa suomen kieleen myös hajoittamalla sanat vaikkapa morfeemeiksi ja opettamalla trigrammimalli näiden pienempien palojen yli.

Huomautettakoon vielä, että tässä estimoitiin todennäköisyys $P(w_i, w_{i-1}, w_{i-2})$ sille, että aineistossa tulee vastaan trigrammi $\{w_i, w_{i-1}, w_{i-2}\}$. Itse kielimallissahan tarvitaan todennäköisyyksiä $P(w_i|w_{i-1}, w_{i-2})$, jotka saadaan tässä estimoiduista arvoista laskettua.

4. Muutetaan hämmentyneisyyden (perplexity) kaavaa niin, että voidaan suoraan käyttää log-todennäköisyyksiä:

$$\begin{aligned} perp(w_1, w_2, \ldots, w_N) &= \prod_{i=0}^{N} P(w_i|w_{i-1}, \ldots, w_1)^{-\frac{1}{N}} \\ &= \prod_{i=0}^{N} 10^{-\frac{1}{N}\log(P(w_i|w_{i-1},\ldots,w_1))} \\ &= 10^{-\frac{1}{N}\sum_{i=0}^{N}\log(P(w_i|w_{i-1},\ldots,w_1))} \end{aligned}$$

Lasketaan summa erikseen:

$$\begin{aligned} &\sum_{i=0}^{N} \log(P(w_i|w_{i-1}, \ldots, w_1)) \\ = \; & \overbrace{-4.1763}^{\text{kielen}} \; \overbrace{-2.1276}^{\text{oppiminen}} \; \overbrace{-0.4656}^{\text{on}} \; \overbrace{-0.001 - 4.2492}^{\text{monimutkainen}} \; \overbrace{-0.8876}^{\text{ja}} \; \overbrace{+0.0495 - 4.1804}^{\text{huonosti}} \\ & \overbrace{-0.1415 - 0.1652 - 5.2195}^{\text{ymmärretty}} \\ = \; & -21.5644 \end{aligned}$$

Niille sanoille, joille ei löytynyt trigrammimallia, jouduttiin käyttämään sekä peräänty-miskerrointa että todennäköisyyttä. Jos myöskään bigrammimallia ei löytynyt, jouduttiin vielä kerran perääntymään.

Sijoitetaan vielä luvut hämmentyneisyyden lausekkeeseen

$$perp(w_1, w_2, \ldots, w_N) = 10^{\frac{-21.5644}{-7}} \approx 1200$$

Tulosta voi ajatella vaikka niin, että kielimalli vastaa sellaista kielimallia, joka joutuu valitsemaan 1 200 yhtä todennäköisen sanan väliltä (ei ihan eksaktisti paikkansapitävä väite).

Sana 'tapahtumaketju' ei ollut 64 000 yleisimmän sanan joukossa ja ei sisältynyt siis kielimalliin. Kielimallin ohi meni siis $\frac{1}{8} \approx 13\%$ sanoista.