Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2006

Luennot:
Timo Honkela ja
Teemu Hirsimäki

Laskuharjoitukset:
Sami Virpioja

Luentokalvot:

Krista Lagus,
Timo Honkela ja
Teemu Hirsimäki

Luentomateriaali

Contents

1 YLEISTÄ KURSSISTA

1.1 Kurssin suorittaminen

Kurssi suoritetaan tekemällä harjoitustyö ja läpäisemällä tentti.

1.2 Kirja

Kurssilla hyödynnetään kirjaa:

Christopher D. Manning, Hinrich Schütze:
Foundations of statistical natural language processing,
MIT Press, 1999.

Kirja löytyy TKK:n pääkirjastosta ja tietotekniikan kirjastosta.

Tutustumiskappale on nähtävillä laboratorion sihteerin Tarja Pihamaan huoneessa B326 olevassa harmaassa peltisessä vetolaatikostossa.

1.3 Tentin järjestelyt

Tentissä on 5 tehtävää *à* 6 pistettä, maksimi 30 pistettä.

Tentissä saa olla mukana matemaattinen kaavakokoelma ja tavallinen funktiolaskin.

Tenttiin ilmoittaudutaan normaalisti eli Topin kautta.

1.4 Tenttikysymyksistä

Tentissä pyritään mittaamaan sitä kuinka hyvin opiskelija on perehtynyt toisaalta tilastollisen kielenkäsittelyn sovellusongelmiin ja toisaalta alan keskeisiin menetelmiin.

Tehtävät tulevat painottumaan luennoilla ja laskareissa käsittelemiin aiheisiin. Kuitenkin kirjan lukeminen näiden aiheiden osalta on suositeltavaa.

Tehtävät voivat olla esseetehtäviä, pieniä sanallisia tehtäviä ja laskutehtäviä. Laskutehtävät ovat samantyyppisiä kuin laskareissa.

Tehtävinä voi olla esim. tietyn sovellusongelman selostaminen (mistä on kysymys), mitä menetelmiä ongelmaan on käytetty tai voidaan käyttää, jonkin (tietyn) menetelmän selostaminen yksityiskohtaisesti, tai eri menetelmien hyvien ja huonojen puolien vertaaminen.

Voidaan myös edellyttää kykyä tulkita mitä oletuksia jossain mallissa tehdään, ja arvioida kuinka paikkansapitäviä ne ovat ko. sovellusongelman kannalta.

1.5 Harjoitustyö

Kurssin suoritukseen kuuluu pakollinen harjoitustyö.

Jos haluaa kurssista suoritusmerkinnän toukokuun tenttitulosten yhteydessä, harjoitustehtävä on saatava hyväksytysti läpi toukokuun 1. päivään mennessä.

Lisätietoja harjoitustyöstä on kurssin www-sivuilla.

2 JOHDANTO

2.1 Tilastollinen luonnollisen kielen käsittely

- Kieliteknologian osa-alue
- Sovelletaan informaatiotekniikan, tilastomatematiikan, ja tietojenkäsittelytieteen menetelmiä kieliteknologisiin ongelmiin.
- Rakennetaan malleja luonnollisesta kielestä niin, että niiden sisältämät todennäköisyysarvot estimoidaan (hyvin) suurista aineistoista (nk. *korpuksista*).
- Menetelmäaloja: koneoppiminen, hahmontunnistus, tilastotiede, todennäköisyyslasku, signaalinkäsittely
- Lähialoja: kielitiede, korpuslingvistiikka, fonetiikka, keskusteluntutkimus, tekoälytutkimus, kognitiotiede

2.2 Luonnollisen kielen käsittelyn sovelluskohteita

Sovelluskohteita ovat mm.

- tiedonhaku
- tekstien järjestäminen ja luokittelu
- puheentunnistus
- luonnollisen kielen käyttöliittymät esimerkiksi tietokantoihin ja varauspalveluihin

2.3 Mallinnuksen peruskäsitteitä

- Malli — Jonkin ilmiön tai datajoukon kattava kuvaus.
  Esim: sääntökokoelma joka kuvaa suomen morfologian.

- Malliperhe, malliavaruus — joukko potentiaalisia malleja joita harkitaan ilmiön kuvaamiseen. Esim. niiden sääntöjen kokoelma jota voitaisiin periaatteessa käyttää kielen syntaksin kuvaamiseen.
- Mallin valinta — prosessi jonka kautta päädytään johonkin tiettyyn malliin. Algoritmit usein tämäntyyppisiä: vuorotellaan mallin evaluointia ja mallin muuttamista, pyrkien kohti parempaa mallia.
- Oppiminen — ks. mallin valinta.
- Probabilistinen malli(perhe) — esittää ilmiöiden todennäköisyyksiä.
- Iteratiivinen — vähän kerrassaan, toiston kautta tapahtuva

2.4 Yleisestä kielitieteestä

Tavoiteena kuvata ja selittää toisaalta kielen (kielten) säännönmukaisuudet, toisaalta kielen (kielten) monimuotoisuus.

Tavoitteena on *konstruoida malli kielestä*.

Kielen ilmenemismuotoja mm. keskustelut visuaalisella kontaktilla ja ilman, viittomalla, yksinpuhelut, kirjoitetut artikkelit, kirjat, luennot, ja muut kielelliset viestit eri viestinvälineitä ja -ympäristöjä käyttäen.

Laajemmin nähtynä kielen mallinnuksen tavoitteena on selvittää ja kuvata:

- Miten ihmiset käyttävät kieltä, mitä todella sanotaan?
- Mitä kielen käyttäjä tahtoo tai mihin pyrkii sanoessaan jotain?

2.5 Lähestymistapoja kieli-ilmiöihin

- Autonominen kielitiede:
  Selvitetään kielissä esiintyviä säännönmukaisuuksia ja variaatiota.
- Kognitiivinen kielitiede:
  Selvitetään kielen käsittelyyn liittyviä kognitiivisia mekanismeja, kuten sitä, miten kielikyky syntyy ja muotoutuu ihmisessä (ja muissa olennoissa), ja miten tuotamme ja ymmärrämme kieltä.
- Luonnollisen kielen käsittely tekoälyn osa-alueena:
  Kehitetään kielen ilmausten automaattisen tulkinnan ja tuottamisen mekanismeja. Selvitetään kielen ja maailman välisiä yhteyksiä ja kehitetään malleja niiden toiminnalliseen kuvaukseen.

2.6 Perinteinen lähestymistapa kielitieteessä

Ominaisuus 1: Perinteisen lähestymistavan mukaan kieli on kuvattavissa *joukkona 'kovia' sääntöjä*, esim. produktiosääntöjä.

Esimerkki: Englannin substantiivilauseke NP koostuu valinnaisesta artikkelista DET=[a, the, an], valinnaisesta määrästä adjektiiveja ADJ=[brown, beautiful,...] ja substantiivista N=[flower, building, thought...].

NP => (Det)? (ADJ)* N

Ominaisuus 2: Sääntöjen avulla pyritään kuvaamaan mitkä lauseet ovat hyvinmuodostettuja (sallittuja, kieliopin mukaisia) ja mitkä väärinmuodostettuja (kiellettyjä, kieliopin vastaisia).

Mallinnuksella on kaksi tavoitetta: *kattavuus* ja *tarkkuus*.

2.7 Kielen mallintamisen haasteita

- Monitulkintaisuudet
- Tulkinnan kontekstuaalisuus
- Kielen tulkinnan sumeus
- Jatkuvien ja diskreettien ilmiöiden välinen suhde
- Kielen muuttuminen
- Tulkinnassa tarvittavan tietämyksen määrä ja laatu
- Multimodaalinen kommunikaatio
- Tulkinnan subjektiivisuus ja intersubjektiivisuus

2.8 Probabilistinen esitystapa

- Probabilistisessa mallissa malliperheenä todennäköisyydet. (vertailukohta: kaksiarvoinen esitystapa jossa asiat ovat joko-tai, tosia tai epätosia)
- Esitystapa mahdollistaa tiedon esittämisen silloinkin, kun ei voida muodostaa kategorista sääntöä, mutta on olemassa preferenssi: Subjekti on ennen predikaattia 90% tapauksista P(A)=0.9.
- 'Kova' sääntö: P(A)=1 tai P(A)=0.
- Probabilistisessa representaatiossa tiedon kerääminen ja mallin päivittäminen voi tapahtua iteratiivisesti, vähitellen. Lisäesimerkit tarkentavat aiemmin muodostettua alustavaa kuvaa.

2.9 Probabilistisen esitystavan ja sumean esitystavan suhde

- Probabilistinen näkökulma: kuinka todennäköinen jokin tapahtuma on.
- Sumeus: missä määrin jokin alkio kuuluu johonkin joukkoon, tms.

2.10 Perusteluja datasta oppimiselle

Miksi kannattaa muodostaa malleja automaattisesti, datasta oppimalla tai estimoimalla (eli automaattisesti), eikä asiantuntijatietoa kirjaamalla?

- Data on halpaa ja sitä on paljon, myös sähköisesti.
- Voidaan saada mallit aikaan nopeammin / vähemmällä ihmistyövoimalla / pienemmin kustannuksin.
- Kielen muuttuessa mallit voidaan estimoida uudestaan helposti.
- Asiantuntijatietämys hankalaa tuottaa tai kerätä (mm. konsistenssiongelmat).
- Asiantuntijatietoa käytettäessä malliperhettä rajoittaa 'ihmisbias'.
- Koneiden 'kognitiiviset ominaisuudet' eroavat ihmisen vastaavista.
- Toteutettaessa kielikykyä koneille ei tarvitse rajoittua ihmiselle helposti ymmärrettäviin malleihin.
- Aineistolähtöinen keskittää resurssit niihin ilmiöihin jotka todella esiintyvät. Resurssien käyttö suhteessa ilmiön keskeisyyteen aineistossa.

2.10.1 Onnistuneen oppivan mallinnuksen seurauksia

- Resurssien käytön tehostuminen: Voidaan ulottaa mallinnus laajempaan kielijoukkoon, ja yksittäisen kielen sisällä eri osa-alueisiin.
- Laadullinen parannus, koska koneellisesti pystytään käymään läpi suuri joukko malleja ja koska mallin valinnassa ei ole inhimillistä biasta (ainakaan samassa määrin kuin käsin muodostetuissa malleissa).

2.10.2 Riskejä ja haasteita

- Datan valinta ja kattavuus,
- sopivien malliperheiden määrittely,
- optimointimenetelmien tehokkuus.

2.11 Ihmisen kielikyky ja kielen oppiminen

Miten kielikyky ihmisellä syntyy ja muotoutuu? Mikä osa on synnynnäistä, mitä opitaan?

2.11.1 Rationalistinen näkemys: Kielikyky on synnynnäinen, ja oma erillinen kielimodulinsa

Keskeisiltä osin ihmismielen ja kielen rakenne on kiinnitetty (oletettavasti geneettisesti määrätty). Perusteltu argumentti stimuluksen vähyydestä (mm. Chomsky 1986). Kannattajia mm: Chomsky, Pinker.

Vrt. tekoälytutkimus 1970-luvulla: tietämyksen koodaaminen käsin. Saatiin aikaan pienimuotoisia älykkään oloisesti käyttäytyviä systeemejä (mm. Newell & Simon: Blocks world). Systeemit usein käsin koodattuja sääntöpohjaisia järjestelmiä. Näiden laajentaminen on kuitenkin osoittautunut hyvin hankalaksi.

2.11.2 Empiristinen näkemys: Kieli opitaan, kielikyky toteutuu osana yleistä kognitiivista systeemiä

Amerikkalaiset strukturalistit. Zellig Harris (1951) jne: tavoitteena kielen rakenteen löytäminen automaattisesti analysoimalla suuria kieliaineistoja. Ajatus siitä että hyvä rakennekuvaus (grammatical structure) on sellainen joka kuvaa kielen kompaktisti.

Nykyisin melko yleisen näkemyksen mukaan mieli ei ole täysin tyhjä taulu, vaan oletetaan että tietyt (1) rakenteelliset preferenssit yhdessä (2) yleisten kognitiivisten oppimisperiaatteiden ja (3) sopivanlaisen havaintoaineiston kanssa johtavat kielen oppimiseen.

Vrt. adaptiivisten menetelmien tutkimus, havaintopsykologia ja laskennallinen neurotiede, ihmisen havaintomekanismien ja piirreirroittimien muotoutuminen aistisyötteen avulla (*plastisiteetti*).

Avoimia kysymyksiä:

- Tarvittavan prioritiedon määrä ja muoto?
- Mitä ovat tarvittavat oppimisperiaatteet?
- Minkälaista syötettä ja missä järjestyksessä tarvitaan?

2.11.3 Käytännöllinen lähestymistapa

Tavoite voi olla puhtaasti käytännöllinen: kehittää toimivia, tehokkaita kieliteknologisia menetelmiä ja järjestelmiä.

Eri menetelmiä sovellettaessa ei välttämättä oteta rationalismi-empirismi- vastakkainasetteluun lainkaan kantaa.

Aineistoihin (korpuksiin) pohjautuvat ja tietämysintensiiviset mallit ovat tällöin samalla viivalla.

Vertailukriteerit:

- lopputuloksen laatu
- lopullisen mallin tilankäytön tehokkuus ja riittävä nopeus (esim. reaaliaikaiset sovellukset)
- mallin konstruoinnin tai oppimisen tehokkuus (tarvittava ihmistyö, prosessointitila ja -aika)

Usein kohteena jokin spesifi kieliteknologinen sovellusongelma, jonka ratkaisemiseksi riittää vain osittainen kielen mallinnus.

Koko kielikyvyn implementointi luultavasti edellyttäisi koko kognitiivisen välineistön ja tekoälyn toteuttamista, mukaanlukien maailmantiedon kerääminen ja esittäminen.

3 MATEMAATTISIA PERUSTEITA

3.1 Todennäköisyyslasku

3.1.1 Peruskäsitteitä

Todennäköisyysavaruus (*probability space*):
Tapahtuma-avaruus Ω — diskreetti tai jatkuva
Todennäköisyysfunktio P
Kaikilla tapahtuma-avaruuden pisteillä A on todennäköisyys: $0 \leq P(A) \leq 1$

Todennäköisyysmassa koko avaruudessa on $\sum_A P(A) = 1$

3.2 Ehdollinen todennäköisyys

A= asiaintila jonka todennäköisyyden haluamme selvittää
B= meillä oleva ennakkotieto tilanteesta, ts. tähän asti tapahtunutta

Ehdollinen todennäköisyys, A:n todennäköisyys ehdolla B:

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{1}$$

Oletetaan että on jo heitetty kolikkoa kerran ja saatu kruuna. Mikä nyt on todennäköisyys että saadaan 2 kruunaa kolmen heiton sarjassa?

Alunperin mahdolliset heittosarjat: {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}

Prioritiedon B perusteella enää seuraavat sarjat mahdollisia: { HHH, HHT, HTH, HTT }

$P(A|B) = 1/2$

3.2.1 Ketjusääntö

$P(A_1, ...A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1, A_2)\dots P(A_n|A_1, \dots, A_{n-1})$

3.2.2 Riippumattomuus

Tilastollinen riippumattomuus:

$$P(A, B) = P(A)P(B) \tag{2}$$

Sama ilmaistuna toisin: se että saamme lisätiedon B ei vaikuta käsitykseemme A:n todennäköisyydestä, eli:

$P(A) = P(A|B)$

Huom: tilastollinen riippuvuus ≠ kausaalinen riippuvuus!

Esim. jäätelön syönnin ja hukkumiskuolemien välillä on tilastollinen riippuvuus. (Yhteinen kausaalinen tekijä ehkä lämmin kesäsää.)

3.3 Bayesin kaava

Paljon käytetty Bayesin kaava perustuu ajatukseen siitä, että koska kahden tapahtuman yhdessä esiintymisessä ei ole kyse kausaalisesta riippuvuudesta, tapahtumien järjestystä voidaan vaihtaa:

$$P(A, B) = P(B)P(A|B) = P(A)P(B|A) \tag{3}$$

Eli $P(B|A)$ voidaan laskea $P(A|B)$:n avulla.

$$P(B|A) = \frac{P(A, B)}{P(A)} = \frac{P(B)P(A|B)}{P(A)} \tag{4}$$

3.3.1 Todennäköisimmän tapahtuman määrittely

Jos A = lähtötilanne, joka ei muutu (esim. jo tapahtuneet asiat), ja haluamme ainoastaan tietää, mikä tulevista tapahtumista B on todennäköisin, $P(A)$ on normalisointitekijä joka voidaan jättää huomiotta: $arg\max_B P(B|A) = arg\max_B \frac{P(B)P(A|B)}{P(A)} = arg\max_B P(B)P(A|B)$

3.4 Satunnaismuuttuja

Periaate: satunnaismuuttuja on se asia, josta ollaan kiinnostuneita, ja joka kussakin kokeessa saa jonkin arvon.

- Jatkuva-arvoinen satunnaismuuttuja: $X : \Omega => \mathbb{R}^n$, jossa $\mathbb{R}$ on reaalilukujen joukko ja n on avaruuden dimensio. Jos $n > 1$ puhutaan myös satunnaisvektorista.
- Diskreetti satunnaismuutuja: $X : \Omega => S$, jossa S on numeroituva $\mathbb{R}$:n osajoukko.
- Indikaattorimuuttuja: $X : \Omega => 0, 1$.

Todennäköisyysjakauma *probability mass function pmf* $p(x)$ kertoo miten todennäköisyysmassa jakautuu satunnaismuuttujan eri arvojen kesken. Jakauman massa aina = 1 (muussa tapauksessa ei ole tn-jakauma).

3.5 P:n laskeminen

- Yleisesti P on tuntematon, ja estimoitava datasta, tyypillisesti erilaisten tapahtumien frekvenssejä laskemalla.
- Koko todennäköisyysjakauman estimoinnin sijaan on mahdollista käyttää *parametrisia malleja* todennäköisyysjakaumille: tällöin estimoidaan vain jakauman parametrit.
- Bayeslaisessa estimoinnissa datan lisäksi huomioidaan prioritieto.

3.6 Bayesläisestä tilastotieteestä

Tähän asti on tarkasteltu todennäköisyyttä *frekventistisestä* näkökulmasta.

Bayesläinen tulkinta: todennäköisyys kuvastaa *uskomuksen astetta*. Bayesläisessä mallinnuksessa myös prioritieto eli uskomukset *ennen* datan näkemistä ilmaistaan eksplisiittisesti.

3.6.1 Esimerkki 1: ainutkertaiset tapahtumat

Mikä on todennäköisyys sille että maailmankaikkeus loppuu huomenna?

Frekventisti: ei vastausta, koska koetta ei voi toistaa N kertaa.
Bayesläinen: subjektiivinen todennäköisyys (uskomus) on olemassa.

3.6.2 Esimerkki 2: taskussani olevien kolikoiden rahallinen arvo

Arvo on jokin täsmällinen luku, mutta tietoni siitä ovat vajavaiset. Uskomukseni: Arvo on varmasti positiivinen, ja lähes varmasti alle 20 euroa.

3.7 Bayesläinen päätösteoria

Optimaalinen tapa mallin (teorian) valintaan: valitaan malli (teoria), joka uskottavimmin selittää jonkin havaintojoukon.

Ts. maksimoidaan mallin todennäköisyys kun tunnetaan data ts. mallin *posterioritodennäköisyys*: $P(\text{Malli}|\text{data})$

3.7.1 Esimerkki 1: Mallin parametrien valinta

Kolikonheitto. Olkoon malli M_m joka sanoo $P(\text{kruuna}) = m, 0 \leq m \leq 1$. Olkoon s jokin heittojono jossa i kruunaa, j klaavaa.

$$P(s|M_m) = m^i(1-m)^j \tag{5}$$

Frekventistinen näkökulma: valitaan malli joka maksimoi datan todennäköisyyden (*MLE, maximum likelihood estimate*): $arg\max_m P(s|M_m)$

Havainnot: 10 heittoa joista 8 kruunaa.
Frekventistinen lähestymistapa (MLE): $m = \frac{i}{i+j} = 0.8$

Bayesläinen lähestymistapa: kolikkoa tarkastelemalla näyttäisi siltä että kolikko on tasapainoinen, siis dataa katsomatta vaikuttaisi todennäköiseltä että $m = 1/2$ tai niillä main. Tämä uskomus voidaan liittää malliin *priorijakaumana mallien yli*.

Valitaan prioriuskomuksiamme kuvastava priorijakauma

Eräs sopiva priorijakauma olisi gaussinen jakauma jonka keskipiste (ja siis maksimi) on 1/2:ssa. Valitaan kuitenkin prioriksi polynominen jakauma, jonka keskipiste (korkein kohta) 1/2 ja pinta-ala 0 ja 1 välillä on 1: $p(M_m) = 6m(1-m)$

Posterioritodennäköisyys Bayeslaisessa lähestymistavassa:

$$\begin{aligned} P(M_m|s) &= \frac{P(s|M_m)P(M_m)}{P(s)} \\ &= \frac{m^i(1-m)^j \times 6m(1-m)}{P(s)} \end{aligned}$$

jossa $P(s)$ on datan prioritodennäköisyys. Oletetaan, ettei se riipu mallista M_m joten voidaan jättää huomiotta mallia valittaessa.

Maksimoidaan osoittaja etsimällä derivaatan nollakohta m:n suhteen, kun $i = 8$ ja $j = 2$. Tämä on $arg\max_m P(M_m|s) = \frac{3}{4}$

Mallin estimointi on-line

Aloitetaan pelkällä priorimallilla, ja aina uuden havainnon tultua päivitetään malli posteriorimalliksi; ns. MAP (Maximum A Posteriori) -estimointi).

Taustaoletus: peräkkäiset havainnot ovat riippumattomia.

3.7.2 Esimerkki 2: Teorioiden tai malliperheiden vertailu

Havainnot: joukko aidan takaa kuultuja "kruuna" ja "klaava"- sanoja.

Malli/Teoria $M1(\theta)$: joku heittää yhtä kolikkoa, joka saattaa olla painotettu, ja mallin vapaa parametri θ on painotuksen voimakkuus.

Malli/teoria $M2$: joku heittää kahta tasapainoista kolikkoa, ja sanoo "kruuna" jos molemmat kolikot ovat kruunuja, ja "klaava" muuten. Mallin $M2$ mukaan heittojonon, jossa on i kruunaa ja j klaavaa todennäköisyys on siis:

$P(\text{data}|M2) = (\frac{3}{4})^i(\frac{1}{4})^j$

Tehdään oletus: molemmat teoriat/mallit yhtä todennäköisiä *a priori* (ts. ennen kuin on saatu yhtään havaintoa): $P(M1) = P(M2) = 0.5$

Bayesin kaavasta:

$$P(M1|data) = \frac{P(data|M1)P(M1)}{P(data)} P(M2|data) = \frac{P(data|M2)P(M2)}{P(data)} \quad (6)$$

Halutaan selvittää kumpi malleista on uskottavampi. Lasketaan niiden uskottavuuksien välinen suhde:

$$\frac{P(M1|data)}{P(M2|data)} = \frac{P(data|M1)P(M1)}{P(data|M2)P(M2)} \quad (7)$$

Jos suhdeluku on > 1, valitaan malli $M1$, jos < 1, malli $M2$

(Vastaukset eri heittosarjoilla: taulukko 2.1 kirjan sivulla 58)

3.8 Shannonin informaatioteoria

- Claude Shannon, 1948 ("A Mathematical Theory of Communication")
- Tavoitteena maksimoida informaation siirtonopeus kohinaisella kommunikaatikanavalla
- Teoreettinen maksimi datan pakkaamiselle = Entropia H
- Kanavan kapasiteetti C: jos kapasiteettia ei ylitetä, virheiden todennäköisyys saadaan niin alhaiseksi kuin halutaan.
- Nykyiset tiedonpakkausmenetelmät hyödyntävät näitä teoreettisia tuloksia.

3.8.1 Entropia

Olkoon $p(x)$ satunnaismuuttujan X jakauma diskreetin symbolijoukon (aakkoston) A yli:

$$p(x) = P(X = x), x \in A \tag{8}$$

$$H(p) = H(X) = -\sum_{x \in A} p(x) \log_2 p(x) \tag{9}$$

(Määritellään $0 \log 0 = 0$).

Entropia ilmaistaan tavallisesti biteissä (kaksikantainen logaritmi), mutta muunkantaiset logaritmit yhtä lailla ok.

Jos symbolijoukko on tasajakautunut, entropia on maksimissaan.

Esimerkki: 8-sivuisen nopan heittäminen, kommunikoitava yksittäisen heiton tulos.

$$\begin{aligned} H(X) &= -\sum_{i=1}^{8} p(i) \log p(i) = -\sum_{i=1}^{8} \tfrac{1}{8} \log \tfrac{1}{8} \\ &= -\log \tfrac{1}{8} = log 8 = 3\text{bittiä} \end{aligned}$$

Pätee yleisesti: jos viestin todennäköisyys on $p(i)$, sen optimaalinen koodinpituus on $-\log p(i)$ bittiä.

Vaihtoehtoinen kirjoitustapa entropian kaavalle:

$$\begin{aligned} H(X) &= -\sum_{x \in A} p(x) \log p(x) = \sum_{x \in A} p(x) \log \tfrac{1}{p(x)} \\ &= E(\log \tfrac{1}{p(x)}) \end{aligned}$$

ts. entropia = optimaalisen koodinpituuden odotusarvo, eli montako bittiä keskimäärin on käytettävä yhden viestin välittämiseen.

3.8.2 Yhteisinformaatio (Mutual Information, MI)

Yhteisinformaatio I muuttujien X ja Y välillä on

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{10}$$

3.8.3 Kohinainen kanava-malli

Binäärinen kommunikointikanava, lähetetään 1 tai 0.
p = todennäköisyys jolla kanavalla lähetetty bitti kääntyy päinvastaiseksi.

Kanavan kapasiteetti C on tällöin:

$$C = \max_{p(X)} I(X;Y) = 1 - H(p) \tag{11}$$

4 Tekstikokoelmien analysointi ja tiedonhaku

Tiedonhaussa tehtävänä on hakea käyttäjän tiedontarvetta vastaavaa tietoa suurista dokumenttikokoelmista.

Ongelmaa tutkittu vuosikymmenet erillään NLP-tutkimuksesta, johtuen erilaisista käytetyistä menetelmistä. Nykyisin lähentymistä, koska myös NLP:ssä tilastolliset menetelmät valtaavat alaa.

Ad hoc retrieval - käyttäjä kirjoittaa hakulausekkeen ja systeemi vastaa palauttamalla joukon dokumentteja, joiden on tarkoitus vastata tiedontarpeeseen.

4.1 Exact match retrieval – täsmälliset osumat

Hakukriteerit määrittelevät täsmällisiä haettavia ominaisuuksia, ja vastauksena annetaan dokumentit jotka täyttävät nämä kriteerit täsmälleen.

Tämä hakutyyppi on käytössä monissa vanhemmissa tietokannoissa [esim. kirjastojen cdrom-tietokannat]

Tunnetuin alalaji: Boolen haut, joissa haettavan dokumentin kriteerit yhdistetään Boolen logiikan avulla.

Lähestymistapa toimii kohtuullisesti pienillä ja homogeenisilla kokoelmilla, kokeneen hakijan käytössä.

Ongelmia etenkin suurilla ja heterogeenisilla kokoelmilla:

- Tuloksena voi olla tyhjä joukko tai valtava määrä osumia – ei voi tietää ennalta.
- Käyttäjän on hyvin vaikea rajata haku siten että saisi juuri haluamansa dokumentit, mutta mahdollisimman vähän roskaa.

- Saman sisällön voi ilmaista monella eri tavalla — täsmällisen haun systeemeissä pitäisi nämä kaikki tavat ilmaista täsmällisesti, ja toisilleen vaihtoehtoina.
- Haun tulokset eivät ilmesty paremmuusjärjestyksessä (koska kaikki ovat yhtä hyviä).
- Ei tiedetä paljonko ja minkälaisia 'lähes yhtä hyviä' dokumentteja oli.

4.2 Ranking – Järjestetyt osumat

Täsmällisen osumajoukon palauttamisen sijaan järjestetään kaikki dokumentit paremmuusjärjestykseen sen mukaan, miten hyvin ne vastaavat hakulauseketta tai muulla perusteella (vrt. esim. Google / PageRank).

Lähestymistapoja esim. probabilistinen haku ja johonkin samankaltaisuusmittaan perustuva haku.

Nykyään täsmähakua yleisempi hakujärjestelmätyyppi.

4.3 Sanojen selityksiä

haku (query): hakusana, hakulause, hakulauseke. Se millä haetaan.

termi, indeksointitermi: Sanastoon kuuluva sana, siis osa dokumenttien representaatiota. Kaikki sanat eivät ole termejä. Termien ei edes tarvitse välttämättä olla sanoja: ne olla myös sanojen alkuosia (esim. 5 ensimmäistä kirjainta) tai sanojen perusmuotoistettuja muotoja. Termeihin voi kuulua myös geneerisiä koodeja.

Relevanssi (relevance): vastaavuus hakulauseen (tai sen tarkoituksen) kanssa.

Relevanssipalaute (Relevance feedback): tapa jolla käyttäjä voi interaktiivisesti tarkentaa ja uudelleenkohdentaa hakuaan, antamalla palautetta siitä kuinka hyviä systeemin antamat dokumentit olivat. Ad-hoc-hauissa eräs tutkimuskohde.

suodatus (filtering), reititys (routing): tekstinkategorisoinnin erikoistapaus; kategorisoidaan dokumentit relevantteihin ja ei-relevantteihin.

4.4 Tiedonhakujärjestelmien perusosia

4.4.1 Käänteisindeksi (inverted index)

Osoittimet sanoista dokumentteihin, sekä frekvenssit dokumenteissa. Joskus myös osoittimet tekstipositioihin dokumentissa.

4.4.2 Sulkusanalista (stop word list)

Lista sanoista joiden indeksointi estetään, yleensä aineistoriippumaton.

Listaan valitaan sanoja joita pidetään hakujen kannalta hyödyttöminä tai häiritsevinä. Esim. kieliopilliset tai funktiosanat, mm. suljettujen sanaluokkien sanat kuten pronominit.

Voi sisältää myös muita yleisiä, indeksoinnin kannalta melko tyhjiä sanoja (esim. apuverbit ja muut yleisimmät verbit kuten 'mennä', 'tulla')

Osuu jossain määrin päällekkäin yleisimpien sanojen listan kanssa.

Sulkusanalista vähentää merkittävästi indeksin kokoa, koska monet estettävistä sanoista yleisiä.

Huono puoli on että sulkusanalistalla olevat sanoilla ei voi hakea, esim. 'milloin ja missä' sisältää pelkästään sulkusanalistasanoja. Vrt. myös 'it magazine'.

4.4.3 Stemming (juureksi palautus) tai perusmuotoistaminen

Stemming on approksimaatio morfologiselle analyysille. Siinä poistetaan sanoista päätteiksi katsotut pätkät, tarkoituksena saada pelkkä sananvartalo. Vartaloita käytetään indeksointitermeinä.

Esimerkkejä mahdollisista vartaloista ja sananmuodoista:

Vartalo	Vartalon sananmuotoja
laugh-	laughing, laugh, laughs, laughed
gall-	gallery, galleries (ongelma: gall)
etsi-	etsiskellä, etsittiin, etsin
yö-	yöllinen, yötön, yöllä
öi-	öisin, öinen
aika-	aikana, aikaan, aikaa
aj-	ajallaan, ajaton, ajat, ajoissa
ajat-	ajatella, ajatus (ongelma: vrt. ed.)

Kuten esimerkeistä näkyy, stemming on rankasti yksinkertaistava ratkaisu, ja sopii huonosti esim. suomen kielelle.

Yhdellä perusmuodolla voi olla useita eri hakuvartaloita.

Vartalon katkaisukohdan valinta on jossain määrin mielivaltainen kompromissi spesifiyden ja kattavuuden välillä.

Suomen sanojen perusmuotoistus on mahdollista tehdä esimerkiksi TWOLilla (Koskenniemen 2-tasomalli).

4.5 Hakumenetelmien evaluointimittoja

N = dokumenttimäärä, joka hakujärjestelmää pyydettiin palauttamaan
REL = tälle haulle relevanttien kokonaismäärä dokumenttikokoelmassa
rel = tälle haulle relevanttien lukumäärä palautetussa dokumenttijoukossa

4.5.1 Tarkkuus ja saanti

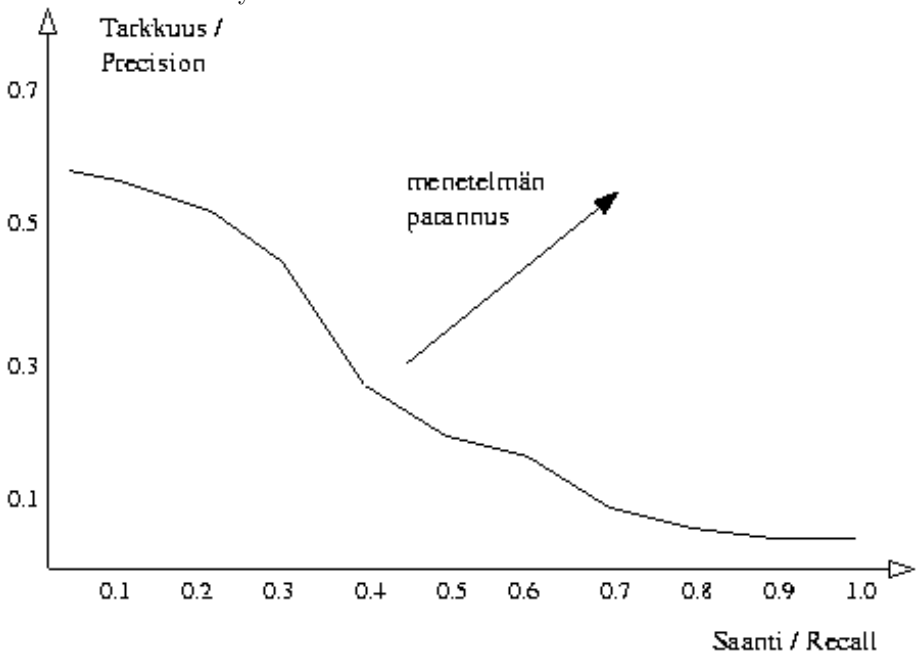
Perusmitat hakujärjestelmien evaluoinnissa.

Tarkkuus l. *precision* P: Relevanttien osuus vastaukseksi saaduista dokumenteista, $P = rel/N$

Saanti l. *recall* R: Vastaukseksi saatujen relevanttien osuus kaikista relevanteista, $R = rel/REL$.

Kun palautettavien lukumäärä nousee, tarkkuus tyypillisesti laskee ja saanti kasvaa.

Tarkkuus-saanti -käyrä:



Esimerkki soveltamisesta menetelmien vertailuun
(%= relevantti, x=epärelevantti dokumentti):

Mitta	Menetelmä 1	Menetelmä 2	Menetelmä 3
	d1: %	d10: x	d6: x
	d2: %	d9: x	d1: %
	d3: %	d8: x	d2: %
	d4: %	d7: x	d10: x
	d5: %	d6: x	d9: x
	d6: x	d5: %	d3: %
	d7: x	d4: %	d5: %
	d8: x	d3: %	d4: %
	d9: x	d2: %	d7: x
	d10: x	d1: %	d8: x
Tarkkuus kun n=5	1.0	0.0	0.4
Tarkkuus kun n=10	0.5	0.5	0.5
Interpoloimaton tarkk.	1.0	0.3544	0.5726
Interpoloitu (11-pist.)	1.0	0.5	0.6440

Jos ajattelee lukevansa hakukoneen palauttamaa listaa ylhäältä alkaen, menetelmä 1 on näistä selvästi paras. Kuitenkin tarkkuus 10 dokumentin kohdalla on niille sama.

Huonoa: tarkkuus ja saanti eivät huomioi tulevatko oikeat osumat alku- vai loppupäässä. Siksi on olemassa myös muita mittoja.

4.5.2 F-mitta

Toinen tapa mitata tarkkuutta ja saantia yhtäaikaa, yhdellä mitalla:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} \tag{12}$$

jossa R on recall (saanti) ja P precision(tarkkuus).

Voidaan käyttää evaluoimaan menetelmiä kun palautettavien dokumenttien lukumäärä on kiinnitetty ja halutaan huomioida sekä tarkkuus että saanti.

4.5.3 Menetelmien vertailu

Yleensä luvut keskiarvoistetaan useiden hakujen (esim. 50) yli, ja verrataan menetelmien saamia keskiarvoja.

Lisäksi pitäisi tehdä tilastollinen testi (esim. t-testi) jolla varmistetaan havaittujen erojen tilastollinen merkitsevyys.

TREC: Text retrieval competition

TREC on kansainvälinen vuosittainen tiedonhaun kilpailu jossa eri sarjoja (esim. monikielinen tiedonhaku).

Aluksi jaetaan aineisto, jolla menetelmänsä hyvyyttä voi tutkia ja optimoida menetelmää.

Testiaineisto annetaan sokkona eli kilpailijat eivät saa tietää oikeita vastauksia (ts. mitkä dokumentit ovat relevantteja millekin haulle).

Lopuksi julkaistaan relevanssitiedot kullekin dokumentille sekä lasketaan kunkin menetelmän hyvyydet keskitetysti samoilla mittareilla.

4.5.4 Ongelmanasettelun ja evaluoinnin ongelmallisuudesta

Edellä esitetyn evaluoinnin taustalla on periaate:

Probability ranking principle (PRP): On optimaalista järjestää dokumentit niiden relevanssin todennäköisyyden mukaan, ts. relevanteimmiksi estimoidut ensin.

Poikkeuksia/ongelmia:

PRP olettaa että dokumentit ovat riippumattomia, mutta todellisuudessa näin ei ole.

Esim. duplikaatit, tai dokumentit jotka muuten toistavat päällekkäistä informaatiota jonkin edellisen kanssa: Käyttäjä ei ehkä halua lukea samaa asiaa monesta lähteestä, vaan pikemminkin saada kattavan kuvan hyvistä hakua vastaavista dokumenteista.

PRP olettaa että peräkkäisten hakujen sarjassa haut ovat toisistaan riippumattomia, ts. että kyse on yksittäisistä toisiinsa liittymättömistä kysymys-vastauspareista.

Kuitenkin parhaimmillaan kyse on pikemminkin dialogista, jonka aikana kyselijän tiedon tarve tarkentuu, laajentuu tai uudelleenkohdistuu ymmärryksen kasvaessa. Peräkkäiset haut ovat siis toisistaan riippuvia.

4.6 Vektoriavaruusmalli

Vector space model, VSM (G. Salton et al, 1975)

- Yleisesti käytetty standardimenetelmä.
- Dokumentti esitetään vektorina, jonka dimensioita ovat sanaston sanat (indeksointitermit), ja dimension arvona jokin funktio termin frekvenssistä dokumentissa ja sen painosta, joka ei riipu tästä dokumentista
- Dokumentit ja hakulause esitetään samassa vektoriavaruudessa. Hakua lähimpänä olevat dokumentit palautetaan.
- Etäisyydet lasketaan tyypillisesti nk. kosinietäisyyksinä, joskus myös Euklidisena etäisyytenä.

4.6.1 Yleisesti käytetty termien painotusmenetelmä: tf.idf

tf: term frequency
idf: inverse document frequency

IDF yhdistää termin lokaalin merkittävyyden eli esiintymistiheyden tietyssä dokumentissa sekä termin globaalin merkittävyyden eli esiintymistiheyden koko aineistossa tai dokumenteissa.

Notaatio:
$tf_{t,d}$ = termin w_t lukumäärä dokumentissa d
df_t = niiden dokumenttien lukumäärä joissa termi w_t esiintyy
cf_t = termin frekvenssi koko kokoelmassa

Näiden huomioiminen voidaan tehdä monella eri tavalla. Eräs vaihtoehto:

$$w(i,j) = (1 + \log(tf_{t,d})) \log \frac{N}{df_t} \tag{13}$$

jossa N on dokumenttien lukumäärä kokoelmassa.

Eri tf.idf-komponentteja taulukossa:

termifrekvenssi	dokumenttifrekvenssi	normalisointi
n (natural) $tf_{t,d}$	n (natural) df_t	n (none)
l (logarithm) $1 + \log tf_{t,d}$	t $\log \frac{N}{df_t}$	c (cosine)
a (augmented) $0.5 + \frac{0.5tft,d}{max_t(tf_{t,d})}$		

4.7 Latenttien muuttujien menetelmät

- Aiemmin hyödynnetty dokumentin representoinnissa vain tietoja yksittäisten sanojen esiintymistä
- Ongelma: Ei käytä minkäänlaista tietoa sanojen semanttisesta samankaltaisuudesta (kahden sanan keskinäinen etäisyys oletetaan samaksi, sanoista riippumatta)
- Ratkaisu: Jos voidaan projisoida sanat ja dokumentit jonkinlaiseen *latenttien semanttisten piirteiden avaruuteen* ja suorittaa etäisyyslaskenta siellä.
- Hyödynnetään semanttisen avaruuden muodostamisessa sanojen yhteisesiintymätietoja. Esim. jos sanat 'HCI', 'vuorovaikutus', 'käyttäjä' ja 'käyttöliittymä' esiintyvät poikkeuksellisen usein yhdessä (tässä: samoissa dokumenteissa), voidaan olettaa että ne liittyvät semanttisesti toisiinsa.

4.7.1 Latent Semantic Indexing-menetelmä (LSI)

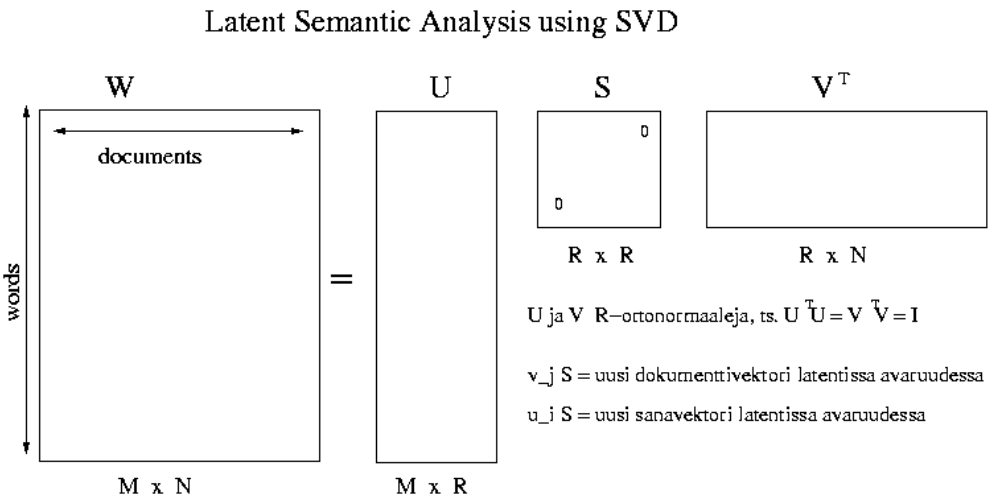
Perusajatus: tehdään sana-dokumenttimatriisille singulaariarvohajotelma eli SVD (Singular Value Decomposition), ja otetaan lopputulokseen mukaan vain avaruuden R merkitsevintä dimensiota.

Lähtökohta: W eli dokumentti-sana-yhteisesiintymämatriisi, jonka alkiot ovat jokin funktio sanan lukumäärästä dokumentissa. Esim.

$w_{i,j} = (1-\epsilon_i \frac{c_{i,j}}{n_j})$, jossa $c_{i,j}$ on sanan i määrä dokumentissa j, n_j on dokumentin j sanojen kokonaismäärä, ja $epsilon_i$ on sanan i normalisoitu entropia koko korpuksessa. Myös tf.idf-painotuksia voidaan käyttää.

Lasketaan R:n asteen SVD(W): $(\hat{W}) = USV^T$, jossa
S on diagonaalimatriisi jonka diagonaalissa singulaariarvot ja U ja V tarvitaan sanojen ja dokumenttien projisointiin latenttiin avaruuteen. (T tarkoittaa matriisin transpoosia).

SVD laskee optimaalisen R-ulotteisen approksimaation W:lle.



R:n arvoksi suositellaan 100-200.

Tulkinta

LSI esittää dokumentin sisällön semanttisten piilomuuttujien ('abstraktien käsitteiden') lineaarikombinaationa (summana). Piilomuuttujia on R kappaletta.

Samankaltaisissa dokumenttiympäristöissä esiintyneet sanat saavat samankaltaisen latentin representaation.

Projektioita latenttiin, semanttiseen avaruuteen voidaan soveltaa mm. tiedonhakuun, dokumenttien klusterointiin ja sanojen klusterointiin.

4.7.2 Riippumattomien komponenttien analyysi

Vastaavalla tavalla kuin LSA voidaan sana-dokumenttimatriisille laskea toinen muunnos, nimittäin ICA, Independent component analysis eli riippumattomien komponenttien analyysi.

Erona edelliseen on, että nyt etsitään latentit muuttujat (projektiosuunnat) jotka ovat mahdollisimman *riippumattomia* toisistaan jonkin tietyn jakaumien riippumattomuutta mittaavan mitan mielessä (esim. kurtoosi).

ICA:n teoriaa ja mm. kieliteknologisia sovelluksia tutkitaan intensiivisesti Adaptiivisen informatiikan tutkimusyksikössä.

4.8 Dimension pienennys

Vektoriavaruusmallissa vektorien dimensio on sanaston koko eli valtava.

Vielä satojen tuhansien dokumenttien aineistossa sanasto voi olla yhtä suuri kuin dokumenttien määräkin—uudet dokumentit tuovat yhä uutta sanastoa.

Sanaston määrää kuitenkin pienentävät seuraavat esikäsittelyn toimet:

- sulkusanalistan käyttö (pieni vaikutus)
- harvinaisten sanojen karsiminen (huomattava vaikutus, koska suuri osa sanaston sanoista on harvinaisia, ks. Zipf'in laki)
- sanojen perusmuotoistaminen (mutta suuri osa kohdatuista sanoista on lingvististä tietoa käyttävälle perusmuotoistavalle mallille aina tuntemattomia, joten tämä auttaa vain osaksi), tai
- sanojen katkaisu 'juurimuotoihin'

Edellämainittujen jälkeenkin sanasto voi kuitenkin helposti sisältää kymmeniä tuhansia sanoja (indeksointitermejä).

Mikäli vektoreita halutaan lisäksi ryhmitellä tai luokitella, monet oppivat menetelmät joiden kompleksisuus on vahvasti sidoksissa datan dimensioon, ovat vaikeuksissa.

Seuraavilla menetelmillä voi dimensiota pienentää edelleen:

- LSI (on käytetty dokumenttien dimension pienennykseen)
- SOM (sanakartta, varhainen versio WEBSOM-menetelmästä)
- ICA (dimension pienennys on sivuvaikutus, mutta siis ainakin periaatteessa sovellettavissa)
- Satunnaisprojektio (on käytetty dokumenttien dimension pienennykseen)

Näin ollen esim. LSI:n käyttöä voi perustella pelkästään dimension pienennysnäkökulmasta, välittämättä siitä parantuuko dokumenttien semanttinen kuvaus vai ei.

4.8.1 Satunnaisprojektio

Satunnaisprojektiossa otetaan tietyllä tavalla muodostettu satunnaismatriisi jota käytetään datavektorien projisointiin pienempiulotteiseen avaruuteen.

$\mathbf{n}_i$ - alkuperäinen dokumenttivektori dokumentille i
$\mathbf{R}$ - satunnaismatriisi jonka kolumnit ovat normaalijakautuneita yksikkövektoreita. Dimensionaalisuus on $rdim \times ddim$, $ddim$ on alkuperäinen dimensio ja $rdim$ uusi, $rdim << ddim$
$\mathbf{x}_i$ - uusi, satunnaisprojisoitu dokumenttivektori dokumentille i, vektorin dimensio $rdim$.

Tällöin projisoidut dokumenttivektorit saadaan seuraavasti:

$$\mathbf{x}_i = \mathbf{R}\mathbf{n}_i \ . \tag{14}$$

Dimension pienennyksessä on oleellista että projektion yksikkövektorit ovat mahdollisimman ortogonaalisia (ts. korrelaatiot vektorien välillä ovat mahdollisimman pieniä). $\mathbf{R}$:n kohdalla vektorit eivät ole täysin ortogonaalisia, mutta mikäli $rdim$ on riittävän suuri, ja vektorit on poimittu satunnaisesti hyperyksikköympyrän tasajakaumasta, keskimääräiset korrelaatiot ovat hyvin pieniä.

$rdim$:lle tyypillisesti käytetyt arvot ovat luokkaa 100-1000.

4.9 Dokumenttikartat

Dokumenttikarttoja tuotetaan Kohosen itseorganisoiva kartta (Self-Organizing Map, SOM) -menetelmää käyttäen. Kustakin dokumenttikokoelmaan kuuluvasta dokumentista muodostetaan vektoriesitys (ks. vektoriavaruusmalli) ja usein vektorien dimensiota jollakin em. menetelmistä pienentäen.

TKK:lla dokumenttikarttojen tuottamisen menetelmiä on tutkittu WEBSOM-tutkimuksen puitteissa erityisen aktiivisesti 1995-99. Nykyisin menetelmästä on runsaasti erilaisia variantteja ja sovelluksia eri aloilla.

Käyttötapoja ovat mm. samoilu, assosiatiivinen haku, avainsanahaku ja suodatus (demonstraatio).

4.9.1 Ohjelmaesimerkki

Dokumenttikarttojen tekemiseen liittyvä esimerkki löytyy www-sivulta *http* : $//www.cis.hut.fi/Opinnot/T-61.5020/Luennot06/luento06\_6/$ Tämä esimerkki kannattaa käydä varsin yksityiskohtaisesti läpi.

4.9.2 Tyylipiirteisiin perustuva dokumenttikartta

Esimerkki tyylipiirteisiin perustuvan dokumenttikartan tuottamisesta löytyy sivulta $http://www.cis.hut.fi/Opinnot/T-61.5020/Luennot06/luento06\_7/$

Tämäkin esimerkki kannattaa käydä varsin yksityiskohtaisesti läpi.

Karlgrenin käyttämiä tyylipiirteitä ovat esimerkiksi:

- merkkien lukumäärä, sanojen lukumäärä
- eri sanaluokkien määrä (edellyttää tagattua korpusta tai tiettyjen sanaluokkien osalta sanalistojen käyttöä; esim. persoonapronominit: "I", "you", "he", "she", jne.
- joidenkin erityisten sanojen esiintyvyys (esim. "me", "I", "it", that", "which", jne.)
- sanojen ja merkkien lukumäärä lauseessa
- tekstissä esiintyvien (token) ja eri sanojen (type) välinen suhde

5 N-grammi-kielimallit

5.1 Tilastollinen mallinnus

1. Otetaan dataa (generoitu tuntemattomasta tn-jakaumasta)
2. Tehdään estimaatti jakaumasta datan perusteella
3. Tehdään päätelmiä uudesta datasta jakaumaestimaatin perusteella

Mallinnuksen osatehtävät voidaan hahmottaa seuraavasti:

- Datan jakaminen ekvivalenssiluokkiin
- Hyvän tilastollisen estimaattorin löytäminen kullekin luokalle
- Useiden estimaattorien yhdistäminen

Tyypillinen oletus: **stationaarisuus**, eli että datan tn-jakauma ei muutu oleellisesti ajan myötä.

5.1.1 Tilastollisen kielimallin tehtävistä

Klassinen tehtävä: seuraavan sanan (tai kirjaimen) ennustaminen jo nähtyjen sanojen (tai kirjainten) perusteella ('Shannon game'). Esim. seuraavissa sovelluksissa:

- puheentunnistus
- optinen merkkientunnistus, käsinkirjoitettujen merkkien tunnistus
- kirjoitusvirheiden korjaus
- tilastollinen konekääntäminen

Estimointimenetelmät yleisiä, soveltuvat myös muihin tehtäviin (esim. WSD, word sense disambiguation, jäsentäminen)

5.2 N-grammimallit

N-grammimalli: ennustetaan sanaa w_n edellisten $n-1$ sanan perusteella:

$$P(w_n|w_1 w_2 \cdots w_{n-1}) \tag{15}$$

Kaava esiintyy myös muodossa $P(w_t|w_{t-(n-1)} w_{t-(n-2)} \cdots w_{t-1})$ jossa t viittaa sanan järjestysnumeroon (ajanhetkeen) koko aineistossa.

Esimerkki: aineistona tämän luennon kalvot, n=4:

		w_{t-3}	w_{t-2}	w_{t-1}	w_t		
...	sitä	enemmän	dataa	tarvitaan	**mallin**	estimointiin	...

Malleille käytettäviä nimiä

n=1	unigram
n=2	bigram
n=3	trigram
n=4	4-gram, fourgram

Yhteys ekvivalenssiluokkiin: n-grammimallissa jokainen $n-1$:n sanan pituinen historia saa oman ekvivalenssiluokkansa. Tämä tarkoittaa että historiat joissa viimeiset 3 sanaa samoja käsitellään keskenään identtisinä tilanteina seuraavan sanan ennustamisen kannalta, eli niillä on yhteinen estimaatti.

Sama n-grammien ominaisuus toisesta näkökulmasta: malli olettaa että sana riippuu ainoastaan $(n-1)$ edeltävästä sanasta, mutta ei tätä kauempana olevista sanoista (ns. Markov-oletus).

Markov-malli: k:n asteen Markov-malli on malli joka asettaa kaikki k:n pituiset historiat samaan ekvivalenssiluokkaan. Ts. n-grammimalli on $n-1$:n asteen Markov-malli.

Esimerkkejä:
Sue swallowed the large green ____

Samppa Lajunen voitti kultaa ____

Parametrien määrän kasvu

	Malli	**Parametreja jos sanasto 20,000**
n=1	unigram	20000
n=2	bigram	$20000^2 = 400$ milj.
n=3	trigram	$20000^3 = 8$ miljardia
n=4	4-gram, fourgram	1.6×10^{17}

5.3 Piirteiden jakaminen ekvivalenssiluokkiin

- Piirteet (sekä jatkuva-arvoiset että diskreetit) voidaan jakaa ekvivalenssiluokkiin 'bins'
- Esim. jatkuva-arvoisen muuttujan 'ikä' jakaminen luokkiin 0-2; 3-5; 7-10; 11-15; 16-25; 26-35 jne
- Mitä useampia ekv.luokkia, sitä enemmän dataa tarvitaan mallin estimointiin, jotta tulokset *luotettavia* kullekin luokalle
- Toisaalta, jos luokkia on kovin vähän, ennustettavan kohdemuuttujan (esim. 'pituus') arvoa ei voida ennustaa kovin *tarkasti*.

Esimerkki: ennustetaan seuraavaa sanaa
1. kolmen edellisen sanan sanaluokan (subst, verbi, adj, num jne) TAI
2. kolmen edellisen sanan perusteella

1. tapauksessa vähemmälläkin datalla jonkinlaiset estimaatit, kun taas
2. tapauksessa tarkempia estimaatteja mutta dataa tarvitaan paljon enemmän.

5.3.1 Joitain tapoja muodostaa ekvivalenssiluokkia

- Isojen ja pienten kirjainten käsittely samalla tavalla (esim. kaiken muuntaminen pieniksi kirjaimiksi)
- Sanojen muuntaminen perusmuotoon (saman sanan eri taivutusmuodot käsitellään ekvivalentteina)
- Ryhmittely sanaluokkatiedon mukaan (syntaktiselta rooliltaan samankaltaiset muodostavat ekv. luokan)
- Sanojen semanttinen ryhmittely (merkitykseltään samankaltaiset muodostavat ekv.luokan)

Kussakin vaihtoehdossa tarvitaan kuitenkin menetelmä jolla sanan ekvivalenssiluokka voidaan luotettavasti päätellä.

Lisäksi ekvivalenssiluokkien olisi hyvä olla sellaisia että niiden sisällä sanat todella käyttäytyvät samankaltaisesti, ts. tarkkuus säilytetään.

5.3.2 Historian huomioimisen eri tapoja

Edellä kuvattiin yksittäisten piirteiden ekvivalenssiluokkien laskemista. Eri tapoja ekvivalenssiluokkien muodostamiseen historian suhteen:

- Poimitaan historiasta tiettyjä piirteitä, mutta niiden sijainnilla ei ole väliä esim. malli: $P(w_t|lauseenpredikaatti), w_{t-1})$
- Käsitellään sanajonon sijaan sanajoukkoa ('sanasäkkiä', bag-of-words), eli ei välitetä sanojen järjestyksestä:
  $P(w_n|w_1, w_2, \ldots, w_{n-1})$

5.4 N-grammimallin tilastollinen estimointi

Annettuna: joukko näytteitä jotka osuvat kuhunkin ekvivalenssiluokkaan (biniin). Bayesin kaavoista:

$$P(w_n|w_1 \cdots w_{n-1}) = \frac{P(w_1 \cdots w_n)}{P(w_1 \cdots w_{n-1})} \tag{16}$$

Mallin optimointi: maksimoidaan datan todennäköisyys (eli sanojen tn:ien tulo).

Notaatio:

N	Opetusnäytteiden lukumäärä
B	Ekv.luokkien (binien) lukumäärä
w_{1n}	n-grammi $w_1 \cdots w_n$
$C(w_1 \cdots w_n)$	ngrammin $w_1 \cdots w_n$ lukumäärä opetusdatassa
r	n-grammin lukumäärä
N_r	Niiden binien lukumäärä joissa on r näytettä
h	historia (edeltävä sanajono)

5.4.1 Maximum likelihood-estimaatti (MLE)

$$P_{\text{MLE}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n)}{N} \tag{17}$$

$$P_{\text{MLE}}(w_n|w_1 \cdots w_{n-1}) = \frac{C(w_1 \cdots w_n)}{C(w_1 \cdots w_{n-1})} \tag{18}$$

- MLE-estimointi johtaa parametrien valintaan siten että opetusdatan todennäköisyys maksimoituu.
  (Huom: tämä pätee vain tietyin oletuksin, kuten että näytteet, esim. tri-grammien sanakolmikot, oletetaan riippumattomiksi toisistaan. Tämä taas ei pidä paikkaansa mm. overlapin takia.)
- Koko tn-massa jaetaan opetusdatassa esiintyneiden tapausten kesken, niiden frekvenssien suhteessa.
- Antaa siis tn=0 tapaukselle jota ei nähty opetusdatassa, eli ei jätä lainkaan tn-massaa aiemmin näkemättömille sanoille.
- Koska yleisesti sanajonon tn lasketaan kertomalla kunkin sanan tn, yksikin nolla saa koko sanajonon tn:n nollaksi.
- Esimerkki datan harvuudesta: ensimmäisten 1.5 miljoonan sanan jälkeen (IBM laser patent text corpus) 23% myöhemmistä trigrammeista oli ennennäkemättömiä.
- MLE ei kovin hyödyllinen estimaatti harvalle datalle, kuten n-grammeille.
- Tarvitaan siis systemaattinen tapa jolla huomioidaan ennennäkemättömien sanojen ja ennennäkemättömien n-grammien tn:t. Tätä kutsutaan mm. nimellä *tasoitus* eli *smoothing*

Taulukko 6.3: MLE-estimaatteja Austenin kirjoista eräälle lauseelle eri n-grammeilla.

In person	*she*		*was*		*inferior*		*to*		*both*		*sis*
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P($
1	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034	the
2	to	0.032	to	0.032	to	0.032	**to**	**0.032**	to	0.032	to
3	and	0.030	and	0.030	and	0.030			and	0.030	and
4	of	0.029	of	0.029	of	0.029			of	0.029	of
...											
8	was	0.015	**was**	**0.015**	was	0.015			was	0.015	wa
...											
13	**she**	**0.011**			she	0.011			she	0.011	she
...											
254					both	0.0005			**both**	**0.0005**	bot
...											
435					sisters	0.0003					**sis**
...											
1701					**inferior**	**0.00005**					
2-gram	$P(\cdot \mid person)$		$P(\cdot \mid she)$		$P(\cdot \mid was)$		$P(\cdot \mid inferior)$		$P(\cdot \mid to)$		$P($
1	and	0.099	had	0.141	not	0.065	**to**	**0.212**	be	0.111	of
2	who	0.099	**was**	**0.122**	a	0.052			the	0.057	to
3	to	0.076			the	0.033			her	0.048	in
4	in	0.045			to	0.031			have	0.027	and
...											
23	**she**	**0.009**							Mrs	0.006	she
...											
41									what	0.004	**sist**
...											
293									**both**	**0.0004**	
...											
∞					**inferior**	**0**					
3-gram	$P(\cdot \mid In,person)$		$P(\cdot \mid person,she)$		$P(\cdot \mid she,was)$		$P(\cdot \mid was,inf.)$		$P(\cdot \mid inferior,to)$		$P($
1	UNSEEN		did	0.5	not	0.057	UNSEEN		the	0.286	to
2			**was**	**0.5**	very	0.038			Maria	0.143	Cha
3					in	0.030			cherries	0.143	Hou
4					to	0.026			her	0.143	Twi
...											
∞					**inferior**	**0**			**both**	**0**	**sist**
4-gram	$P(\cdot \mid u,I,p)$		$P(\cdot \mid I,p,s)$		$P(\cdot \mid p,s,w)$		$P(\cdot \mid s,w,i)$		$P(\cdot \mid w,i,t)$		$P(\cdot$
1	UNSEEN		UNSEEN		in	1.0	UNSEEN		UNSEEN		
...											
∞					**inferior**	**0**					

5.4.2 Laplacen laki eli 'yhden lisäys'

Annetaan hiukan tn-massaa näkemättömille tapauksille lisäämällä jokaiseen lukuun 1:

$$P_{\text{LAP}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + 1}{N + B} \tag{19}$$

- Vastaa Bayesin estimaattia priorilla, että kaikki tapahtumat ovat yhtä todennäköisiä, ja tähän prioriin uskotaan aivan kuin olisi nähty yksi näyte joka lajia.
- Esim. 44 milj. sanan AP newswire-korpus, sanaston koko 400,653 sanaa, jolloin bigrammeja 1.6×10^{11}, eli N =44 milj., $B = 1.6 \times 10^{11}$
- Jos data on hyvin harvaa, antaa liiaksi tn-massaa ennen näkemättömille tapauksille (tässä 46.5% tn-massasta).
- Ts. uskotaan tasajakauma-prioriin liian vahvasti verrattuna datan määrään
- Kannattaisiko 1:n sijaan uskoa että ollaan nähty esim. 0.0001 jokaista näytettä?

Odotetun frekvenssin estimaatteja seuraavassa taulukossa:

$r = f_{\text{MLE}}$	$f_{\text{empirical}}$	f_{Lap}	f_{del}	f_{GT}	N_r	T_r
0	0.000027	0.000137	0.000037	0.000027	74 671 100 000	2 019 187
1	0.448	0.000274	0.396	0.446	2 018 046	903 206
2	1.25	0.000411	1.24	1.26	449 721	564 153
3	2.24	0.000548	2.23	2.24	188 933	424 015
4	3.23	0.000685	3.22	3.24	105 668	341 099
5	4.21	0.000822	4.22	4.22	68 379	287 776
6	5.23	0.000959	5.20	5.19	48 190	251 951
7	6.21	0.00109	6.21	6.21	35 709	221 693
8	7.21	0.00123	7.18	7.24	27 710	199 779
9	8.26	0.00137	8.18	8.25	22 280	183 971

Table 6.4 Estimated frequencies for the AP data from Church and Gale (1991a). The first five columns show the estimated frequency calculated for a bigram that actually appeared r times in the training data according to different estimators: r is the maximum likelihood estimate, $f_{\text{empirical}}$ uses validation on the test set, f_{Lap} is the 'add one' method, f_{del} is deleted interpolation (two-way cross validation, using the training data), and f_{GT} is the Good-Turing estimate. The last two columns give the frequencies of frequencies and how often bigrams of a certain frequency occurred in further text.

5.4.3 Lidstonen laki, Jeffreys-Perksin laki

$$P_{\text{Lid}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + \lambda}{N + B\lambda} \tag{20}$$

Voidaan osoittaa että ylläoleva tarkoittaa lineaarista interpolointia tasajakauma-priorin ja MLE-estimaatin välillä. Asetetaan $\mu = N/(N + B\lambda)$:

$$P_{\text{Lid}}(w_1 \cdots w_n) = \mu \frac{C(w_1 \cdots w_n)}{N} + (1 - \mu)\frac{1}{B} \tag{21}$$

- Jeffreysin priori: $\lambda = 1/2$, eli lisätään jokaiseen lukumäärään 1/2 (vastaa sitä että olisi nähty puolikas näyte jokaista lajia). Käytetään myös nimeä *Expected Likelihood Estimation* (ELE)
- On valittava λ:n arvo tavalla tai toisella
- Alhaisilla frekvensseillä tämäkään ei kovin hyvin vastaa todellista jakaumaa

5.4.4 Good-Turing -estimaattori

Ks. frekvenssien frekvenssi-histogrammeja taulukossa 6.7.

$$P_{GT}(w_1 \cdots w_n) = \frac{r^*}{N}, jossa r^* = \frac{(r+1)S(r+1)}{S(r)} \tag{22}$$

ja $S(r)$ on odotusarvo N_r:lle, tai vaihtoehtoisesti, arvo joka on saatu sovittamalla jokin tasainen käyrä frekvenssien frekvensseille: $N_r = S(r)$

Simple Good-Turing -estimaattori: Valitaan käyräksi potenssifunktio: $S(r) = ar^b$ jossa parametrit a ja b sovitetaan frekvenssien frekvenssi-histogrammin mukaan.

Melko hyvä estimaattori, yleisesti käytössä.

Bigrams				Trigrams			
r	N_r	r	N_r	r	N_r	r	N_r
1	138741	28	90	1	404211	28	35
2	25413	29	120	2	32514	29	32
3	10531	30	86	3	10056	30	25
4	5997	31	98	4	4780	31	18
5	3565	32	99	5	2491	32	19
6	2486		...	6	1571		...
7	1754	1264	1	7	1088	189	1
8	1342	1366	1	8	749	202	1
9	1106	1917	1	9	582	214	1
10	896	2233	1	10	432	366	1
	...	2507	1		...	378	1

Table 6.7 Extracts from the frequencies of frequencies distribution for bigrams and trigrams in the Austen corpus.

5.4.5 Muita tasoitusmenetelmiä

Termi 'discounting' viittaa siihen, että nähtyjen n-grammien tn:iä alennetaan ja tätä massaa jaetaan ennen näkemättömille.

- Absoluuttinen alennus (absolute discounting): Kaikista nähdyistä n-grammeista vähennetään vakio-tnmassa σ joka jaetaan tasan näkemättömien n-grammien kesken.
- Lineaarinen alennus (linear discounting): Skaalataan nähtyjen n-grammien tn:ät vakiolla joka on hiukan pienempi kuin 1, ja saatu tnmassa jaetaan tasan ei-nähtyjen kesken. Ei kovin hyvä, koska 'rankaisee' frekventtejä enemmän—kuitenkin niiden estimaatit ovat parempia.
- Witten-Bell discounting: Arvioidaan yllättävien asioiden näkemisen tn-massa sen perusteella kuinka tavallista yllättävien asioiden näkeminen on ollut tähän mennessä: $\sum_{i:C(i)=0} p_i = \frac{T}{N+T}$ jossa T on tähän mennessä nähtyjen binien määrä.

Pohjimmiltaan menetelmien eroissa on kyse on siitä minkälaisia oletuksia tehdään tapauksista, joita ei ole nähty, ja niiden suhteesta tapauksiin, joita on nähty.

Huom: Esim. CMU Statistical Language Toolkit toteuttaa useita eri discounting- ja tasoitusmenetelmiä n-grammeille.

5.5 Estimaattorien yhdistäminen

- Tähän asti tarkasteltu tilannetta jossa pyritään estimoimaan identtinen tn esim. kaikille 3-grammeille joita ei ole nähty.
- Kuitenkin jos 3-grammin osat (esim. 2-grammit) ovat frekventtejä, eikö niistä kerättyä tietoa kannattaisi käyttää 3-grammin tn:n estimoinnissa?
- Motivaationa estimaattien tasoitus (smoothing) tai yleisemmin eri informaationlähteiden yhdistäminen.

5.5.1 Lineaarinen interpolointi

(yleisemmin nimellä äärelliset mikstuurimallit tai sum of experts)

Lasketaan painotettu keskiarvo eri pituisten kontekstien antamista estimaateista:

$$P_{li}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n|w_{n-1}) + \lambda_3 P_3(w_n|w_{n-2}w_{n-1}) \quad (23)$$

($0 \leq \lambda_1 \leq 1$ ja $\sum_i \lambda_1 = 1$)

Parametrit λ voidaan asettaa käsin tai optimoida datan avulla.

5.5.2 Yleinen lineaarinen interpolointi

Edellä parametrit λ eivät riippuneet sanoista joiden kohdalla niitä sovelletaan, eli parametri on vakio vaikkapa kaikille bi-grammeille.

Yleisemmin ne voidaan kuitenkin asettaa riippumaan historiasta:

$$P_{li}(w|h) = \sum_i \lambda_i(h) P_i(w|h) \quad (24)$$

($0 \leq \lambda_1 \leq 1$ ja $\sum_i \lambda_1 = 1$) ja optimoida esim. EM-algoritmilla. Kuitenkin, jos jokaiselle historialle on oma λ ollaan taas datan harvuusongelmassa, ja joudutaan soveltamaan jotain tasoitusta, historioiden ekvivalenssiluokkia tms.

5.5.3 Perääntyminen (backing off)

- Periaate: Katsotaan aina spesifeintä mallia joka antaa 'riittävän luotettavaa' informaatiota tämänhetkisestä kontekstista.
- Eli peräännytään pitkien kontekstien käytöstä yhä lyhempiin: Päätetään uskoa estimaattia jos se perustuu vähintään k näytteeseen (k esim. 1 tai 2)
- Kritiikkiä: Uuden opetusdatan lisääminen voi vaikuttaa voimakkaasti tn:iin kun se aiheuttaa muutoksia useiden sanojen kohdalla niille sovellettavissa n-grammipituuksissa

- Kuitenkin mallit yksinkertaisia ja toimivat melko hyvin, joten yleisesti käytössä.
- back-off -malli on erikoistapaus yleisestä lineaarisesta interpoloinnista: $\lambda_i(h) = 1$ kun k:n arvo riittävän suuri, 0 muulloin.
- Lähestymistapa muistuttaa Kohosen Dynamically Expanding Context (DEC) -algoritmia.

Back-off-mallien käyttöesimerkki:

	$P(she\|h)$	$P(was\|h)$	$P(inferior\|h)$	$P(to\|h)$	$P(both\|h)$	$P(sisters\|h)$	Product
Unigram	0.011	0.015	0.00005	0.032	0.0005	0.0003	3.96×10^{-17}
Bigram	0.00329	0.1219	0.0000159	0.183	0.000449	0.00372	3.14×10^{-15}
n used	2	2	1	2	2	2	
Trigram	0.00529	0.0741	0.0000162	0.183	0.000384	0.00323	1.44×10^{-15}
n used	2	3	1	2	2	2	

Table 6.11 Probability estimates of the test clause according to various language models. The unigram estimate is our previous MLE unigram estimate. The other two estimates are back-off language models. The last column gives the overall probability estimate given to the clause by the model.

5.6 Mallien estimoinnista yleisesti

Seuraava koskee mitä tahansa menetelmien vertailua, ei pelkästään n-grammeja tai kielimalleja.

5.6.1 Held-out estimation

Tavallisesti data jaetaan ennen menetelmien kehittämistä kolmeen osaan

- **Opetusjoukko:** data jolla malli opetetaan
- **Validointijoukko:** opetusjoukosta riippumaton data, jonka avulla valitaan mallin opetuksessa käytettävät parametrit (esim. edellisen kalvon λ)
- **Testijoukko:** edellisistä riippumaton, satunnaisesti valittu datajoukko (kooltaan esim. 10% opetusdatasta), jolla lopullisen mallin hyvyys mitataan.

Testijoukko on pidettävä kokonaan syrjässä menetelmien kehittämisen aikana! Jos testijoukko pääsee vaikuttamaan menetelmänkehitykseen (vaikka vain alitajuisesti), se ei ole enää soveltuva menetelmän testaamiseen.

Kuitenkin usein menetelmänkehitys on syklinen prosessi jossa välillä muutetaan menetelmää ja sitten taas testataan. Siksi voi olla erikseen:
1. **kehittely-testijoukko**, jolla vertaillaan menetelmän eri variantteja
2. **lopullinen testijoukko** jolla tuotetaan julkaistavat tulokset, ja jota ei ole käytetty mihinkään ennen tätä.

Vaihtoehdot testijoukon (ja validointijoukon) valintaan:
1. täysin satunnainen valinta (satunnaisia lyhyitä tekstinpätkiä)
2. pitkiä yhtenäisiä pätkiä (esim. ajallisesti myöhempiä osia datasta)

2-tapa vastaa paremmin mallin käyttötilannetta: se myös antaa realistisemmat, yleensä hiukan huonommat tulokset johtuen siitä, että harvat ilmiöt ovat täysin stationaarisia.

5.6.2 Eri menetelmien vertailusta

Pelkkiä keskiarvotuloksia vertaamalla ei voi tietää ovatko havaitut erot menetelmissä merkitseviä.

Eräs ratkaisu: Mitataan lisäksi tulosten varianssi eri datajoukoilla, ja testataan erojen tilastollinen merkitsevyys esim. t-testillä.

	System 1	System 2
scores	71, 61, 55, 60, 68, 49, 42, 72, 76, 55, 64	42, 55, 75, 45, 54, 51 55, 36, 58, 55, 67
total	609	526
n	11	11
mean $\bar{x}_i$	55.4	47.8
$s_i^2 = \sum(x_{ij} - \bar{x}_i)^2$	1,375.4	1,228.8
df	10	10

$$\text{Pooled } s^2 = \frac{1375.4+1228.8}{10+10} \approx 130.2$$

$$t = \frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{2s^2}{n}}} = \frac{55.4-47.8}{\sqrt{\frac{2\cdot 130.2}{11}}} \approx 1.56$$

Table 6.6 Using the t test for comparing the performance of two systems. Since we calculate the mean for each data set, the denominator in the calculation of variance and the number of degrees of freedom is $(11 - 1) + (11 - 1) = 20$. The data do not provide clear support for the superiority of system 1. Despite the clear difference in mean scores, the sample variance is too high to draw any definitive conclusions.

5.6.3 Ristiinvalidointi (cross-validation)

- Jaetaan data K:hon osajoukkoon, joista 1 kerrallaan on testidata, muut opetusdataa. Toistetaan siten että kukin osajoukko on vuorollaan testidata. K välillä 2...N, jossa N datan määrä.
- Hyöty: Kaikki datat vaikuttavat sekä mallin opetukseen että sen testaukseen, data siis hyödynnetään mahdollisimman tarkasti (tärkeää etenkin kun dataa on vähän).
- useita eri variantteja (deleted estimation, leave-one-out-estimation)

Sekä ristiinvalidoinnin että held-out-estimoinnin avulla voidaan valita mallien parametrejä, ja siis esim. tasoittaa tn-estimaatteja.

5.7 N-grammimallin kritiikkiä

N-grammien ongelmia kielimallina:

- Eivät huomioi pidemmän tähtäimen riippuvuuksia sanojen välillä
- Sanajono yhdessä järjestyksessä ei kontribuoi saman sanajoukon tn:ään jossain toisessa järjestyksessä
- Tasoitusongelmat voi myös nähdä mallin rakenteellisena ongelmana
- Riippuvuudet estimoidaan sanojen välillä suoraan. Intuitiivisesti järkevämmältä tuntuisi että olisivat osaksi joidenkin latenttien muuttujien, kuten käsitteiden ja/tai sanaluokkien tms välillä.
- Kuitenkin: n-grammimalli yhdistää syntaktiset ja semanttiset ja kollokationaaliset lyhyen kontekstin riippuvuudet käytännössä yllättävänkin hyvin toimivalla tavalla, etenkin/ainakin englannille.
- Mallin optimointiin ja tasoitusmenetelmien parantamiseen on käytetty hyvin paljon resursseja. On mahdollista, että on juututtu lokaaliin minimiin malliperheiden suhteen.

6 Puheentunnistus

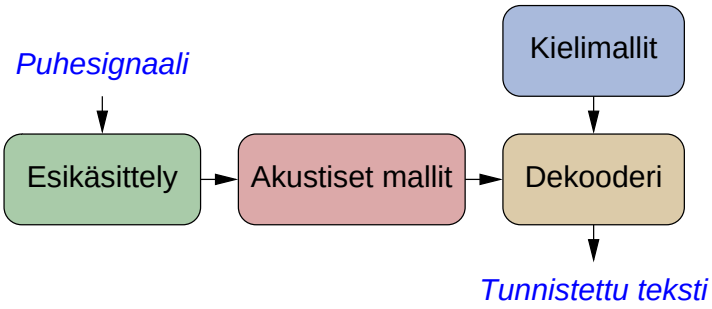
6.1 Mitä puheentunnistus on?

- Puheentunnistin on järjestelmä, joka pyrkii tulkitsemaan puhetta jollain tavalla.

- Käyttökohteita: käyttöliittymän ohjaus, sanelu, kokousten automaattinen kirjaus, palveluautomaatit, haku suurista puheaineistoista
- Tunnistustehtävän vaikeus voi vaihdella todella helposta täysin mahdottomaan riippuen seuraavista tekijöistä:
  - Sanaston laajuus (muutamia sanoja ... rajoittamaton)
  - Puhujien määrä (puhujariippuva ... rajoittamaton)
  - Tehtävä (irrallisia sanoja ... jatkuvaa puhetta)
  - Äänen laatu (hiljainen huone ... ruuhkabussi)

6.2 Tutkimuskenttää

- Tunnistusjärjestelmän eri osiin liittyy paljon tutkimusta:



- Lisäksi käytännön sovelluksissa saatetaan tarvita myös muita menetelmiä:
  - puheen erottelua musiikista ja muista äänistä
  - dialogin hallintaa
  - puhujan tunnistusta
  - hakumenetelmiä puheenhaussa

6.3 Puheentunnistuksen periaate

- Tunnistusjärjestelmää varten opetetaan kaksi mallia:
  1. Akustinen malli, jolla voidaan laskea minkä tahansa foneemisekvenssin sopivuus annettuun puhesignaaliin.
  2. Kielimalli, jolla vaihdaan laskea minkä tahansa foneemisekvenssin järkevyys.
- Itse tunnistus on periaatteessa yksinkertaista: etsitään vain foneemisekvenssi, jolle mallit antavat parhaan vasteen.
- *Demo yksittäisten sanojen tunnistuksesta.*

Huomioita demosta:

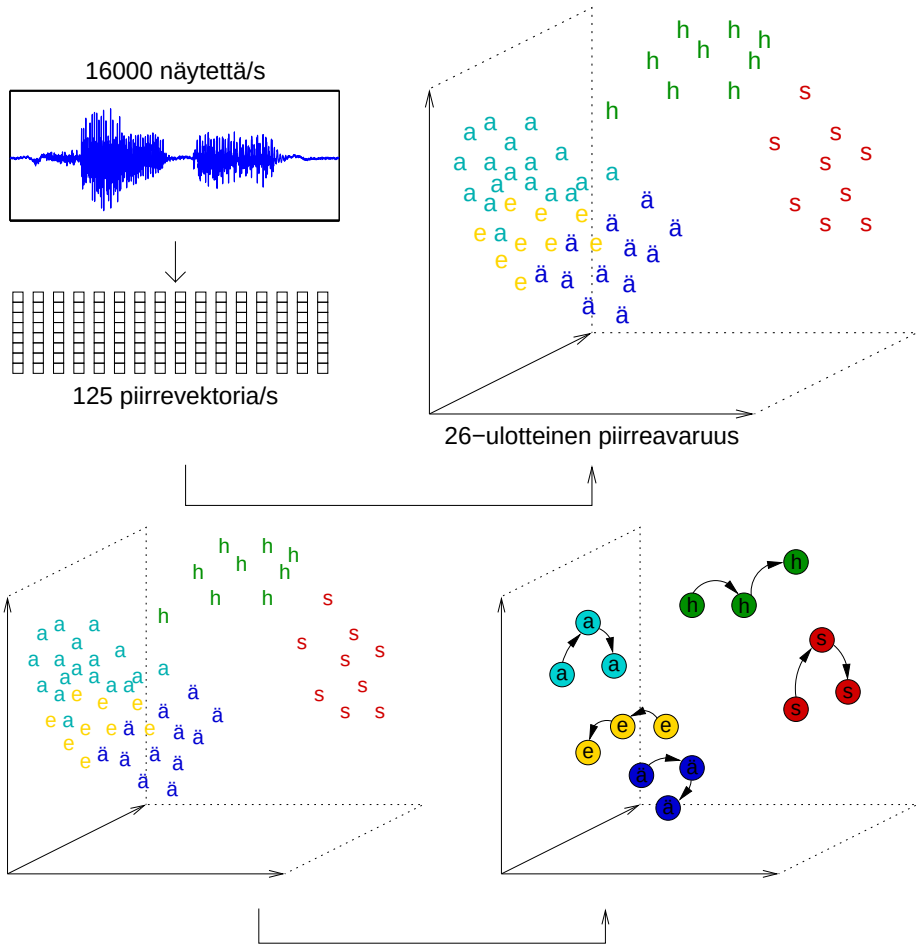
- Hyvin rajatulla sanastolla tunnistus on helppoa — etenkin jos sanat ovat akustisesti hyvin erilaisia.
- Sanastopohjainen tunnistus on ongelmallista suomenkielisen puheen tunnistuksessa, jos sanasto ei ole tarkkaan rajattu.
  - Taivutusmuotojen ja yhdyssanojen vuoksi sanoja on yksinkertaisesti liikaa.
  - *Tästä lisää huhtikuun puheentunnistusluennolla.*

6.4 Puhesignaalin esikäsittely

- Kaivetaan signaalista esiin tunnistustehtävän kannalta olennaisia piirteitä.
  - Alkuperäinen signaali: 16 000 näytettä sekunnissa
  - Esikäsitelty signaali: 125 näytevektoria sekunnissa (26 piirrettä / vektori)
- Menetelmät aikojen saatossa hyviksi havaittuja ja hyvin kieliriippumattomia.
- Yleisimmin käytetään puhesignaalin spektrogrammista laskettuja Melkepstrikertoimia.

6.5 Akustisen mallinnuksen perusidea

- Tarvitaan kymmeniä tai jopa satoja tunteja puhetta, josta tiedetään missä kohdassa lausutaan mikäkin foneemi.
- Foneemimallin opetus:
  - Kerätään kaikki näytteet, joiden kohdalla kyseinen foneemi lausutaan (pilvi pisteitä 26-ulotteisessa avaruudessa)
  - Mallinnetaan pilvi muutamalla Gaussin jakaumalla.
- Parannuksia:
  - Jaetaan kukin foneemi muutamaan tilaan ja mallinnetaan tilat erikseen.
  - Kullekin foneemille monta mallia: “a joka seuraa k:ta”, “a joka seuraa t:tä ja edeltää i:tä”



6.6 Kielimallit

- Puheentunnistuksen yhteydessä kielimallilla tarkoitetaan tietoa, jonka tunnistin olettaa puhutusta kielestä ennen puhenäytteen havaitsemista. Sovelluksesta riippuen se voi olla esimerkiksi:
  - Pelkkä sanasto.
  - Tilakone mahdollisista lauserakenteista.
  - Tilastollinen N-gram -malli.
- Kielimallin avulla tunnistimen hakuavaruutta voidaan rajoittaa. Hyvä kielimalli on välttämätön etenkin jatkuvan puheen tunnistuksessa.
- *Kielimalleista enemmän huhtikuun luennolla 4.4.*

6.7 Tunnistustuloksia

- Viime vuoden testituloksia suomenkielisestä puheentunnistuksesta:

Aineisto	Puhujariippuva	Sanavirhe (%)
Puhekirja	kyllä	7
Radiouutiset	ei	22
TV-uutiset	ei	35
Radioluennot	ei	35
TV-väittelyt	ei	70

6.8 Tunnistustuloksia maailmalta

- Evaluaatioita järjestetään vuosittain. Esimerkiksi Cambridgen yliopiston HTK-järjestelmän tuloksia viime vuosilta englannin tunnistuksesta:

Testiaineisto	Sanavirhe (%)	Nopeus (xRT)	Vuosi
Uutiset (NIST)	11	10	2004
Uutiset (NIST)	15	1	2004
Puhelut (NIST)	16	10	2004
Puhelut (SWB)	24	> 100	2005
Puhelut (SWB)	27	< 10	2005

6.9 Haku puheaineistoista

- Tehtävänä hakea käyttäjää kiinnostavia pätkiä suuresta puheaineistosta.
- Esimerkiksi TV-yhtiöillä on isoja puhe- ja videoaineistoja.
- Menetelmät sietävät hyvin virheitä: hakutulokset eivät käytännössä parane, vaikka tunnistuksen sanavirhe laskisi 20 prosentista.
- *Demo puhehausta.*

6.10 Tunnistustuloksen jälkikäsittely

- Usein on hyödyllistä saada tunnistimelta vaihtoehtoisia hypoteeseja parhaan hypoteesin lisäksi.
- Vaihtoehtoiset hypoteesit voidaan esittää tiiviisti suunnatulla verkolla
  - Verkon avulla saadaan tietoa tunnistustuloksen luotettavuudesta eri kohdissa.
  - Verkkoon voidaan soveltaa parempaa kielimallia, jonka käyttö tunnistuksen aikana olisi liian raskasta.

6.11 Piilo-Markov-Mallit

Hidden Markov Models, HMMs, kätketty Markov-malli

Mallin rakennuspalat:

- Tilat: $S = \{s_1, \ldots s_N\}$
- Tilasiirtymätodennäköisyydet: $A = \{a_{ij}\}, 1 \leq i, j \leq N$
- Havainnot: $\{o_1 \ldots o_t\}$
- Havaintotodennäköisyydet kussakin tilasiirtymässä: $B = \{b_{ijk}\}$

Ero näkyvään Markov-malliin: Vaikka havaintojono tunnetaan, tilasekvenssi ei yksikäsitteisesti tunnettu (kuitenkin tunnetaan tilasekvenssien tn-jakauma).

Esimerkki:

Tilat: Hyvä tuuli / Ärsyyntynyt
Havainnot: 'Upeaa!', 'Lähdetäänkö leffaan?', 'Miksei kukaan ole tyhjentänyt roskista?'

6.11.1 HMM:n variantteja

- mukana nollatransitioita: jotkin tilasiirtymät eivät emittoi mitään (merkitään esim. epsilonilla ϵ)
- arc-emission-HMM: havainnot emittoidaan kaarista: b_{ijk}
- state-emission-HMM: havainnot emittoidaan tiloista: b_{ik}

6.12 HMM-mallin käyttäminen

6.12.1 Output-sekvenssin tuottaminen

Jos HMM-malli on annettu, havaintojonojen tuottaminen siitä on triviaalia, seuraavasti:

t:=1
Arvotaan alkutila X_1 tilojen aloitustn:ien perusteella: $P(s_i) = \pi_i$
while (1)
. . . . . Jos ollaan tilassa i, arvotaan tilasiirtymä $s_i \rightarrow s_j$ tn:llä a_{ij}
. . . . . Arvotaan siirtymässä ij tulostettava symboli o_t=k tn:llä b_{ijk}
end

6.12.2 Havaintojonon tn:n laskeminen

Jos on annettuna tietty havaintojono $O = o_1 \dots o_t$ ja malli $\mu = (A, B, \Pi)$, miten saadaan tehokkaasti laskettua havaintojonon O todennäköisyys mallissa, $P(O|\mu)$?

Suoraviivainen, tehoton ratkaisu: summataan havaintojonon tn kaikkien tilasekvenssien yli (edellyttää $(2T+1)N^T$ kertolaskua, jossa T havaintojonon pituus)

$$P(O|X,\mu) = \prod_{t=1}^{T} P(o_t|X_t, X_{t+1}, \mu) \quad (25)$$

$$P(O|\mu) = \sum_X P(O,X|\mu) = \sum_X P(O|X,\mu)P(X|\mu) \quad (26)$$

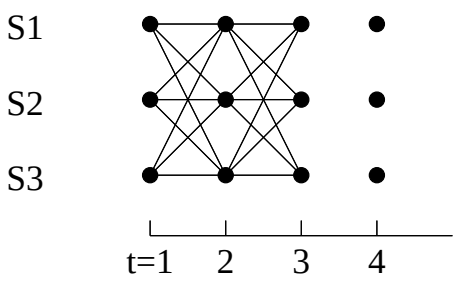
$$= \sum_{X_1 \dots X_{T+1}} \pi_{X_1} \prod_{t=1}^{T} a_{X_t X_t+1} \quad (27)$$

Huomattavasti tehokkaampi ratkaisu saadaan *dynaamisella ohjelmoinnilla* kun huomataan että sekvensseillä on yhteisiä alisekvenssejä, ja lasketaan tn:t kunkin alisekvenssin osalta vain kerran, eli

Forward-algoritmi: (eteenpäinlaskenta)

Kulkien hilaa askeleittain eteenpäin, kerätään tilakohtaisiin apuparametreihin $\alpha_i(t)$ **todennäköisyys olla tilassa** i **ensimmäisen** t**:n havainnon jälkeen**: $\alpha_i(t) = P(o_1 o_2 \dots o_t - 1, X_t = i|\mu)$

- Initialisoi tilojen alkutodennäköisyyksillä: $\alpha_i(1) = \pi_i$
- Induktioaskel: summaa i:n tulokaarista saapuva tn-masssa: $\alpha_i(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ijo_t}$
- Lopuksi: $P(O|\mu) = \sum_{i=1}^{N} \alpha_i(T+1)$



Apuparametri $\alpha_i(t)$ kussakin hilapisteessä.

Huom: **Backward-algoritmi** (taaksepäinlaskenta) samoin mutta ajassa päinvastaiseen suuntaan. Eteenpäinlaskenta $\alpha_i(t)$ ja taaksepäinlaskenta $\beta_i(t)$ voidaan myös yhdistää missä kohdassa aikajanaa tahansa: $P(O|\mu) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t), jossa 1 \leq t \leq T+1$

6.12.3 Dekoodaus eli todennäköisimmän tilajonon etsiminen

Tehtävä: etsi tilajono joka parhaiten selittää havaintojonon O.

Eräs mahdollinen tulkinta: maksimoidaan oikeinarvattujen tilojen *lukumäärä*. Kuitenkin tämä saattaa johtaa hyvin epätodennäköisiin tilasekvensseihin.

Siksi toinen tulkinta: etsitään *todennäköisin tilasekvenssi* joka on tuottanut havaintojonon.

6.12.4 Viterbi-algoritmi

(Tunnetaan myös nimillä DP alignment, Dynamic Time Warping, one-pass decoding)

- etsii havainnoille todennäköisimmän kokonaisen tilasekvenssin tehokkaasti
- Tallettaa jokaiseen hilapisteeseen siihenastisen todennäköisimmän polun:

$$\delta_j(t) = \max_{X_1 \ldots Xt-1} P(X_i \ldots X_{t-1}, o_1 \ldots o_{t-1}, X_t = j|\mu) \tag{28}$$

- Initialisoi tilojen alkutodennäköisyyksillä: $\delta_j(1) = \pi_j$
- Induktioaskel: valitse edellinen tila josta tulee suurin sekvenssin tn:
  talleta tn: $\delta_j(t+1) = \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{ijo_t}$
  talleta pointteri ko. tilaan: $\phi_j(t+1) = arg \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{ijo_t}$
- Lopuksi: Lue hilaa lopusta taaksepäin seuraten ϕ_j:n talletamia kaaria ja kerää todennäköisin tilajono.
- Tasatilanteet voidaan ratkaista arpomalla.
- Keskeinen ero forward-algoritmiin: *todennäköisyyksien summan* sijaan talletetaan *maksimi-tn*.
- Joskus yhden parhaan sekvenssin sijaan halutaan n parasta sekvenssiä (n-best-list). Tällöin talletetaan jokaisessa hilapisteessä $m < n$ parasta tn:ää ja tilaa.
- Viterbi tietyssä mielessä approksimaatio forward-algoritmille. Se ei laske *havainnon* tn:ää, vaan selvittää todennäköisimmän *tilasekvenssin* ja sen tn:n.

6.13 Parametrien estimointi

Jos annettuna tietty havaintojono O (opetusdata), etsitään HMM-mallin parametreille $\mu = A, B, \pi$ arvot jotka uskottavimmin selittävät havainnot. Sovelletaan MLE-estimointia:

$$arg \max_{\mu} P(O_{opetusdata}|\mu) \tag{29}$$

Ei analyyttistä ratkaisua.

6.13.1 Baum-Welch eli forward-backward-algoritmi

- Iteratiivinen optimointimenetelmä, EM-algoritmin sovellus. Periaate: vuorotellaan seuraavia kahta askelta:
  1. Pidä mallin parametrit vakiona, laske $P(O|\mu)$ ja kerää tiedot siitä miten O:n tn-massa jakautui mallin hilaan (eli miten monta kertaa kuljettiin mitäkin reittiä)
  2. Uudelleenestimoi mallin parametrit $\mu \rightarrow \hat{\mu}$ siten että O olisi mahdollisimman todennäköinen
- Taatusti joka kierroksella $P(O|\hat{\mu}) \geq P(O|\mu)$ (kuten EM yleensäkin)
- Malli voidaan initialisoida satunnaisesti tai jollain nopealla raa-alla menetelmällä
- Iteroidaan kunnes datan tn ei enää muutu merkittävästi.
- Löytää lokaalin maksimin, ei välttämättä globaalia (kustannusfunktio on suuren joukon parametrejä luultavasti epälineaarinen funktio).

Parametrien estimaatit sanallisesti:

$$\hat{\pi}_i = \text{ odotettu frekvenssi tilassa } i \text{ hetkellä } t = 1$$

$$\hat{a}_{ij} = \frac{\text{odotettu transitioiden lkm tilasta } i \text{ tilaan } j}{\text{odotettu transitioiden lkm tilasta } i}$$

$$\hat{b}_{ijk} = \frac{\text{odotettu transitioiden lkm tilasta } i \text{ tilaan } j \text{ kun havainto oli } k}{\text{odotettu transitioiden lkm tilasta } i \text{ tilaan } j}$$

6.13.2 Käytännön implementointi sekä joitain ongelmia

Ongelma: Kerrotaan paljon hyvin pieniä tn:iä keskenään → ylivuoto-ongelmia.

Viterbissä vain kertolaskua ja max-operaatiota → voidaan toteuttaa kokonaan logaritmoituna, jolloin kertolaskut summia.

Baum-Welchissä: voidaan käyttää skaalauskertomia joita kasvatetaan ajan t funktiona.

Ongelma: Parametrien suuri määrä

- tarvitaan paljon dataa TAI
- oletetaan että osa parametreistä jaetaan verkon eri osien kesken ('parameter tying': sidottuja tiloja tai sidottuja tilasiirtymiä)
- oletetaan että verkossa on 'rakenteisia nollia', ts. rakenteellisesti mahdottomia tilasiirtymiä

Estimointialgoritmi löytää lokaalin maksimin, ei globaalia

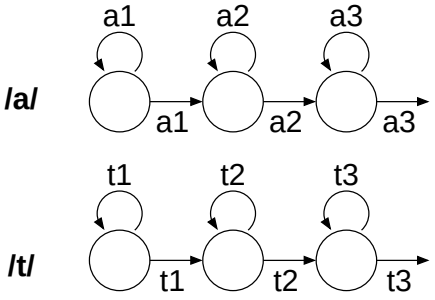
- Parametrien fiksu initialisointi
- Erityisen tärkeää initialisoida havaintotn:t $B = b_{ijk}$

Puheentunnistuksessa Viterbiä käytetään usein approksimaationa myös mallien opetuksessa. Etsitään mallin antama paras polku opetusnäytteisiin, ja päivitetään siirtymä- ja emissio-todennäköisyydet pelkästään parhaan polun mukaan. Tätä toistetaan useita kertoja.

6.14 Soveltaminen puheentunnistukseen

6.14.1 Äännemallit

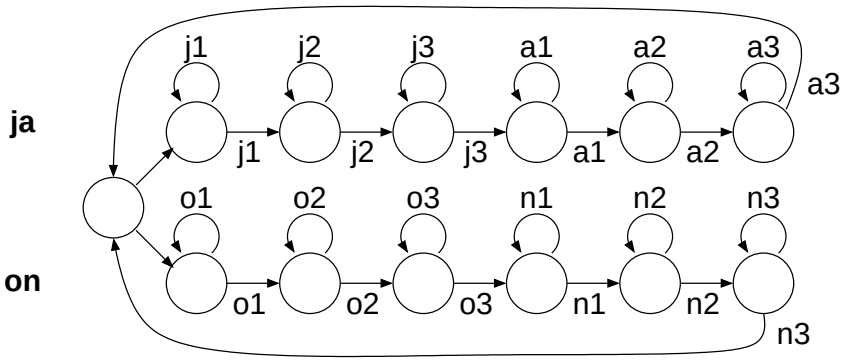
Tyypillisesti jokaiselle ääntelle tehdään oma 3-tilainen HMM. Havainnot ovat puhesignaalin spektristä laskettuja piirteitä.



Mallien parametrien opettamiseen tarvitaan käytännössä useita tunteja puhetta ja transkriptio puheesta.

Jos opetusdataa on kymmeniä tai satoja tunteja, voidaan kullekin ääntelle opettaa useampia malleja. Opetetaan erimerkiksi erillisiä malleja äänteelle /a/ riippuen edellisestä ja seuraavasta äänteestä: "/a/ jota edeltää /k/" tai "/a/ jota edeltää /t/ ja seuraa /u/".

Äännemalleja yhdistelemällä voidaan rakentaa isompi HMM, joka vastaa kokonaista sanaa. Vastaavasti voidaan koota yhdeksi isoksi HMM:ksi kaikki sanat, jotka halutaan tunnistaa. Yhdellä sanalla voi olla monta ääntämisvaihtoehtoa.



Yksinkertainen Viterbi-tunnistus

Kun tunnistimen HMM-verkko on rakennettu, voidaan Viterbi-algoritmilla voidaan etsiä tilasekvenssi, joka selittää parhaiten tunnistettavan puhesignaalin. Tilasekvenssi kertoo sekä parhaan sanasekvenssin, että signaalin segmentoinnin sanoiksi.

Huomioita:

- Etsii todennäköisimmän tilajonon, mikä ei ole sama asia kuin joka ajanhetken todennäköisin tila. Toimii käytännössä hyvin.
- Kukin tilajono vastaa jotain sanojen lausuntavaihtoehtoa. Löytää siis todennäköisimmän *lausuntavaihtoehdon*.
- Isoilla sanastoilla verkossa on tiloja niin paljon, ettei täydellistä Viterbihakua voida tehdä. Joka ajanhetkellä epätodennäköisimmät polut karsitaan ja toivotaan ettei tästä aiheudu virheitä.

Tunnistus N-grammimallin kanssa

Viterbi-algoritmin tehokkuus perustuu oletukseen, että siirtymätodennäköisyydet eivät riipu edellisistä tiloista. Siksi jokaiseen tilaan tarvitsi tallentaan tieto ainoastaan parhaasta polusta siihen asti.

N-grammimallissa sanan todennäköisyys riippuu edellisistä sanoista, joten tilojen siirtymätodennäköisyydet riippuvat pitkälle menneisyyteen. Ongelmaan on kaksi ratkaisua:

- Laajennetaan HMM-verkkoa niin, että tila määrää yksikäsitteisesti sanahistorian. Tällöin voidaan käyttää perus Viterbi-algoritmia, mutta verkoista tulee helposti todella suuria. Viime vuosina tämä lähestymistapa on yleistynyt, kun on kehitetty menetelmiä, joilla isoja verkkoja voidaan rakentaa ja minimoida tehokkaasti.
- Ei laajenneta HMM-verkkoa, mutta tunnistuksen aikana tiloihin talletetaan paras polku jokaista aktiivista sanahistoriaa kohden. Näin itse HMM-verkon koko pysyy kohtuullisena, mutta hakualgoritmi on raskaampi.

N-grammimallin yksiköiden valinta

Puheentunnistimien N-grammikielimallit rakennetaan usein kokonaisten sanojen yli. Suomen kielessä tästä aiheutuu kuitenkin ongelmia, jos tunnistustehtävän sanastoa ei haluta rajoittaa (vapaa sanelu, radiouutisten tunnistus):

- Tarvittava sanasto kasvaa helposti kohtuuttoman suureksi, jotta kaikki yleiset yhdyssanat ja taivutusmuodot saataisiin mukaan.
- Suurella sanastolla mallin todennäköisyyksiä ei saada estimoitua luotettavasti, kun monista sanayhdistelmistä on vähän näytteitä.

Luonnollinen ratkaisu on pilkkoa sanat lyhyempiin paloihin ja rakentaa N-grammimalli palojen yli. Mitä pienempiin paloihin sanat jaetaan, sitä vähemmällä määrällä paloja selvitään ja sitä luotettavammat estimaatit todennäköisyyksille saadaan. Toisaalta lyhyemmät palat vaativat korkeamman asteen mallin, jotta historia yltää yhtä pitkälle kuin sanoilla.

Sanat voidaan pilkkoa monella tavalla:

- **Kirjaimet**: Yksiköitä todella vähän. Tarvittaisiin kuitenkin todella korkea asteluku, jotta malli antaisi hyviä tuloksia. Käytännössä niin isoa mallia ei voida opettaa.
- **Tavut**: Yksiköitä vähän. Toimivat puheentunnistuksessa hyvin. Vieraskieliset sanat vaativat erikoiskäsittelyn. Kieliriippuva.
- **Kielioppiin perustuvat morfeemit**: Suomen kielelle on olemassa valmiita tietokoneohjelmia. Yksiköitä tulee aika paljon, mutta toimivat puheentunnistuksessa hyvin. Vieraskieliset sanat vaativat erikoiskäsittelyn. Kieliriippuva.
- **Tilastolliset morfit (Morfessor)**: Yksiköiden määrään voidaan vaikuttaa. Toimivat puheentunnistuksessa erittäin hyvin. Käsittelee vieraskieliset sanat siinä kuin muutkin. Ohjaamaton menetelmä, jota on sovellettu suomen lisäksi menestyksekkäästi myös turkin- ja vironkieliseen tunnistukseen.

Morfessor-algoritmi

Morfessor-algoritmi (Creutz ja Lagus, 2002) etsii ohjaamattomasti tehokkaan pilkonnan sanoille. Algoritmille syötetään iso tekstiaineisto, ja algoritmi etsii morfit, joilla saadaan minimoitua itse morfiston että morfeilla koodatun tekstiaineiston yhteinen koodauspituus. Pitkillä morfeilla aineisto voidaan koodata tiiviimmin, mutta toisaalta pitkiä morfeja tarvitaan enemmän ja niiden koodaamiseen menee enemmän tilaa.

Projektin kotisivulla on demo, jolla voi kokeilla miten eri sanat pilkkoutuvat: http://www.cis.hut.fi/projects/morpho/

Muita HMM:ien sovelluksia

- sanaluokkien taggaus
- geenisekvenssien analyysi

7 Tilastollinen konekääntäminen

- Automaattinen kielenkääntäminen on eräs pitkäaikaisimmista kieliteknologian tavoitteista.
- Konekääntäminen (machine translation) on kuitenkin hyvin vaikea ongelma.
- Nykyisten konekäännösohjelmien tulos toimii lähinnä raakakäännöksenä, joka voi nopeuttaa aidon kielenkääntäjän työtä, muta ei sellaisenaan kelpaa ihmislukijalle.
- Hyvin rajallisissa sovellusalueissa (kuten säätiedotukset) voidaan päästä kohtuulliseen lopputulokseen täysin automaattisesti; Kanadassa englanti-ranska -käännös ja Suomessa suomi-ruotsi -käännös.
- HAMT = Human Aided Machine Translation, MAHT = Machine Aided Human Translation, L10N = localisation

7.1 Klassinen tilastollinen konekäännösjärjestelmä

- Taustamateriaalina on syytä käydä läpi klassikkoartikkeli:

  Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin: A statistical approach to machine translation

  Se löytyy sivulta $http://www.cs.mu.oz.au/acl/J/J90/J90-2002.pdf$

7.2 Kääntämisen eri tasoja

- Yksinkertaisin lähestymistapa on *sanasta sanaan käännös* korvaa lähtökielen sanoja kohdekielen sanoilla. Lopputuloksen sanajärjestys on usein väärä.
- *Muunnosmenetelmät* (syntaktinen ja semanttinen) rakentavat rakenteisen välirepresentaation lähtökielen sanajonosta muuntavat sen kohdekielen välirepresentaatioksi (jonkinlaisia sääntöjä käyttäen) ja generoivat tästä kohdekielen sanajonon.
- *Syntaktinen muunnosmenetelmä* rakentaa lähtökielen sanajonosta syntaktisen rakennekuvauksen. Lähestymistapa edellyttää toimivaa syntaktista disambiguointia.

Tällä tavoin voidaan ratkaista sanajärjestysongelmat, mutta usein lopputulos ei ole semanttisesti oikein. Esim. saksan 'Ich esse gern' (Syön mielelläni) kääntyisi syntaktisella menetelmällä 'I eat readily' (tai 'willingly', 'gladly'). Englannissa saksan ilmausta vastaavaa verbi-adverbiparia ei kuitenkaan ole, vaan oikea käännös olisi 'I like to eat'.

- *Semanttisissa muunnosmenetelmissä* tehdään syntaktista jäsennystä täydellisempi kuvaus, semanttinen jäsennys, jonka tarkoituksena on saada aikaan käännös, joka on myös semanttisesti oikein.

  Kuitenkin semanttisesti 'sanatarkka' käännös voi olla kohdekielessä kömpelö, vaikka onkin periaatteessa ymmärrettävissä. Esim. espanjan lauseen 'La botella entró la cueva flotando' tarkka käännös olisi 'the bottle entered the cave floating' (pullo tuli luolaan kelluen) mutta luontevampaa olisi sanoa 'the bottle floated into the cave' (pullo kellui luolaan).

  Useiden kömpelöiden ja epäluontevien käännösten käyttö hidastaa ymmärtämistä, vaikka ymmärtäminen olisikin periaatteessa mahdollista. Monitulkintaisuuden mahdollisuudesta johtuen epäluonteva käännös voidaan myös helpommin tulkita väärin.

- *Interlingua* – keinotekoinen yleinen (kieliriippumaton) välikieli tai tietämysrepresentaatio. Käännetään lähtökielestä interlingualle ja interlinguasta mille tahansa kohdekielelle.

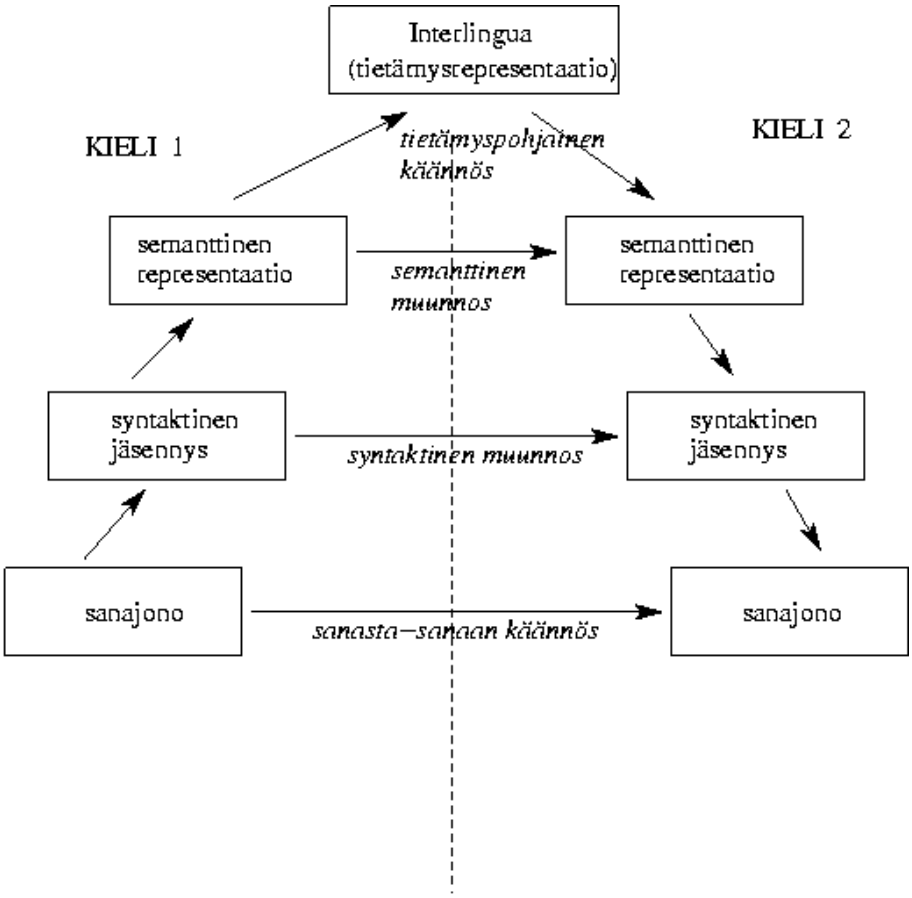
  Kääntimiä n kielen välille tarvitaan tällöin n^2 kpl sijaan vain $2n$ kpl. Lisäksi ne voidaan toteuttaa mahdollisimman suurelta osin yleiskäyttöisillä kielenkäsittelymenetelmillä. Kuitenkin riittävän välikielen määrittely on itsessään hankala ongelma, jota ei ainakaan toistaiseksi ole ratkaistu riittävässä laajuudessa.

Seuraavan sivun kuvassa on näytetty konekäännösjärjestelmän vaihtoehtoiset toteutustavat.

Tilastollisen kielenkäsittelyn menetelmiä voidaan käyttää järjestelmän komponentteina minkä tahansa nuolen kohdalla (esim. jäsentäminen, disambiguointi jne).

Konekääntimet voivat myös olla kombinaatioita symbolisista ja tilastollisista komponenteista.

Pelkästään kielenkääntämiselle erityinen ongelma on *tekstinlinjaus*.



7.3 Tekstinlinjaus

- Tekstinlinjauksella (text alignment) tarkoitetaan kahden erikielisen *rinnakkaistekstin* asettamista kohdakkain siten, että osoitetaan toisiaan vastaavat tekstijonot.
- Rinnakkaisteksteillä tarkoitetaan saman dokumentin erikielisiä käännöksiä.
- Useimmin käytetyt rinnakkaistekstit ovat hallinnollisia tekstejä peräisin maista tai valtioliitoista, joissa on useita virallisia kieliä (esim. EU, Kanada, Sveitsi, Hong Kong).
- Helpon saatavuuden lisäksi hallinnolliset rinnakkaistekstit ovat yleensä konsistentisti ja mahdollisimman tarkasti käännettyjä. Tällainen aineiston korkea laatu on tärkeää sekä tilastollisten menetelmien kehittämiselle että menetelmien evaluoinnille.

- Myös sanoma- ja aikakauslehtiä joskus käytetään, ja myös uskonnollisia tekstejä olisi helposti saatavilla. Kuitenkin tulokset ovat yleensä selvästi heikompia, oletettavasti johtuen vähemmän sanatarkoista ja konsistenteista käännöksistä, ja vähemmän stationaarisesta tekstilajista (esim. ajankohtaiset uutisaiheet muuttuvat nopeasti).
- Tekstinlinjauksessa on yleensä kaksi vaihetta:

    1. Lauseiden ja kappaleiden linjaus: tekstin raakalinjaus, jossa toisiaan vastaavat kappaleet, lauseet ja lauseparit asetetaan suunnilleen kohdakkain.
    2. Sanojen linjaus ja kaksikielisen sanakirjan indusointi, jossa raakalinjatun aineiston perusteella etsitään lähdekielisiä sanojen (ja fraasien) kohdekieliset vastineet.

7.4 Lauseiden ja kappaleiden linjaus

Yleensä lauseiden linjaus on välttämätön ensimmäinen askel monikielisen korpuksen tuottamisessa.

Konekäännöksen ja kaksikielisten sanakirjojen tuottamisen lisäksi linjaus voi hyödyttää myös muita sovelluksia kuten

- Sananmerkitysten disambiguointi: sanan eri merkityksiä voidaan ryhmitellä sen saamien eri käännösvastineiden perusteella.
- Monikielinen tiedonhaku: Tiedonlähde voi olla eri kielellä kuin millä kysymys esitetään.
- Kääntäjän apuväline: Kun dokumenttien tiedot muuttuvat, voidaan automaattisesti osoittaa toisenkielisen dokumentin kohta, jota täytyy myös päivittää, ja ehkä ehdottaa päivitystä.

7.4.1 Jyvitys

Jyvä (bead) on lause tai muutaman lauseen jono ja sitä vastaavat (linjatun tekstin) toisenkielinen lausejono. Kumpi tahansa jono voi olla myös tyhjä. Jokainen lause kuuluu täsmälleen yhteen jyvään.

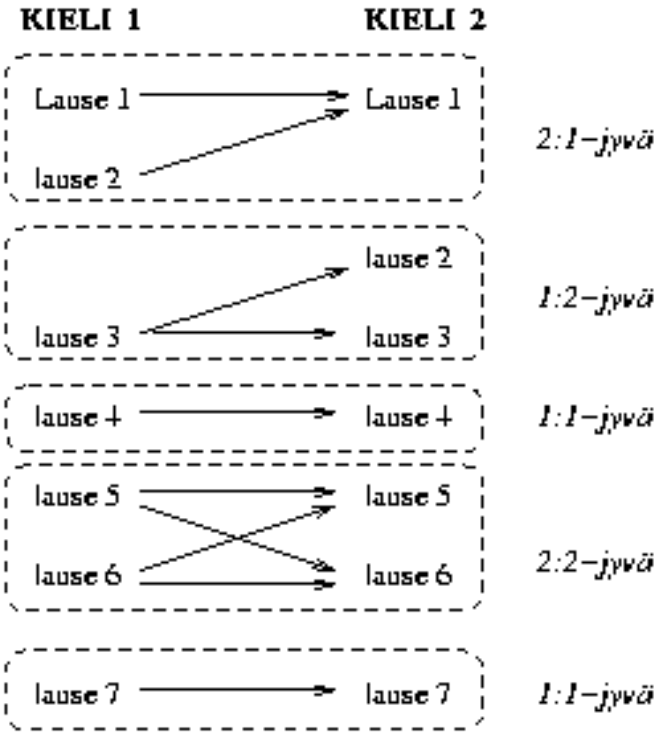
Jyvitys on kuvaus, jossa tekstit on jaettu osiin ja kerrottu, mitä kielen 1 osaa mikäkin kielen 2 osa vastaa.

Lauseiden linjaus ei ole triviaali ongelma, koska yhtä lähtökielen lausetta ei läheskään aina vastaa yksi kohdekielen lause (1:1-jyvä).

1:2 ja 2:2-jyvät (myös 1:3 ja 3:1): Lauseita pilkotaan eri tavoilla. Ihmiskääntäjä käyttää eri järjestyksiä tehdäkseen lopputuloksesta luontevan.

2:2-vastaavuudessa lähtökielen kahden peräkkäisen lauseen osia esitetään kohdekielen kahdessa peräkkäisessä lauseessa (riittävä päällekkäisyys).

Milloin päällekkäisyys on riittävä? Yleensä muutaman sanan siirtyminen ei riitä, vaan edellytetään kokonaisen lausekkeen päällekkäisyyttä.



7.4.2 Poistot ja lisäykset eli 1:0 ja 0:1-jyvät:

Joitain asioita voidaan sanoa eksplisiittisesti toisella kielellä mutta jättää pois toisella kielellä, koska ne oletetaan implisiittisesti tulkittaviksi (ehkä asioiden erilaisen järjestyksen ansiosta, ehkä sanojen erilaisten sivumerkitysten takia, ehkä kulttuurisista syistä).

Eri tutkimusten perusteella voidaan arvioida, että n. 90% vastaavuuksista on tyyppiä 1:1 (tosin osuus on luultavasti kielipari- ja tekstilajiriippuva).

On myös melko tavallista, että kääntäjät järjestävät lauseita eri järjestyksiin. Tässä esitetyt mallit eivät kuitenkaan kykene representoimaan tätä mahdollisuutta vaan tulkitsevat tapaukset mm. poistoiksi ja lisäyksiksi.

7.4.3 Tekstinlinjauksen tilastollisia menetelmiä

Osa tilastollisista menetelmistä perustuu ainoastaan tekstinpätkien pituuksien tarkasteluun, osa taas huomioi lauseissa käytetyn sanaston (merkkijonot).

- Tekstinpätkien pituuksiin perustuvat menetelmät
- Identtisiin merkkijonoihin perustuva menetelmä
- Leksikaaliset menetelmät

Jatkossa: olkoon kielen 1 teksti S jono lauseita $S = (s_1, \ldots, s_I)$ ja kielen 2 teksti T samoin $T = (t_1, \ldots, t_J)$ (S = source, T = target)

7.4.4 Tekstinpätkien pituuksiin perustuvat menetelmät

Useat varhaiset tekstinlinjausmenetelmät ovat tätä tyyppiä.

Etsitään linjaus A, jolla on suurin tn: $arg\max_A P(A|S,T) = arg\max_A P(A,S,T)$ (todennäköisin linjaus voidaan etsiä mm. dynaamisella ohjelmoinnilla).

Useat menetelmät jakavat linjatun tekstin jonoksi jyviä $(B_1, \ldots, B_K)$ ja approksimoivat koko linjatun tekstin tn:ää olettamalla, että jyvän tn ei riipu sen ympäristöstä, vaan ainoastaan jyvän sisältämistä lauseista:

$$P(A,S,T) = \prod_{k=1}^{K} P(B_k) \tag{30}$$

7.4.5 Jyvän todennäköisyyden laskenta

Gale & Church, 1991, 1993:

Jyvän tn. riippuu jyvässä olevien lauseiden pituuksista merkkeinä mitattuna. Perustuu oletukseen, että yhden kielen pitkiä pätkiä todennäköisesti vastaavat pitkät pätkät myös toisessa kielessä.

Oletetaan, että aineistot on jo linjattu kappaletasolla (laskennallisen tehokkuuden vuoksi).

Sallitaan vain linjaustyypit $\{1:1, 1:0, 0:1, 2:1, 1:2, 2:2\}$

Olkoon $D(i,j)$ etsitty pienimmän kustannuksen linjaus lauseiden $s_1, \ldots, s_i$ ja $t_1, \ldots, t_j$ välillä.

Lasketaan $D(i,j)$ rekursiivisesti. Perustapaus, määritellään: $D(0,0) = 0$. Rekur-

sio:

$$
\begin{aligned}
D(i,j) = \min \quad & D(i,j-1) && + cost(0:1 linjaus 0, t_j) && (31)\\
& D(i-1,j) && + cost(1:0 linjauss_i, 0) && (32)\\
& D(i-1,j-1) && + cost(1:1 linjauss_i, t_j) && (33)\\
& D(i-1,j-2) && + cost(1:2 linjauss_i, t_{j-1}, t_j) && (34)\\
& D(i-2,j-1) && + cost(2:1 linjauss_{i-1}, s_i, t_j) && (35)\\
& D(i-2,j-2) && + cost(2:2 linjauss_{i-1}, s_i, t_{j-1}, t_j) && (36)
\end{aligned}
$$

Kunkin tyyppisen linjauksen (jyvän) kustannus lasketaan seuraavasti:

Oletetaan malli: yksi kielen L_1 merkki generoi satunnaisen määrän merkkejä kieleen L_2. Oletetaan generoinneille gaussinen tn-jakauma, jonka keskiarvo μ ja varianssi σ^2 estimoidaan suurista rinnakkaiskorpuksista (saksa/englanti-parille estimoitiin $\mu = 1.1$ koko korpuksesta, ranska/englanti-parille 1.06. Varianssi voidaan estimoida kappaletason linjausta hyväksikäyttäen.)

Kustannuksena voidaan käyttää tekstinpätkien etäisyyden negatiivista log-likelihoodia mallissa:

$$cost(l_1, l_2) = -\log P(\alpha linjaus | \delta(l_1, l_2, \mu, \sigma^2) \quad (37)$$

jossa α on jokin sallituista linjaustyypeistä ja $\delta(l_1, l_2, \mu, \sigma^2) = (l_2 - l_1\mu)/\sqrt{l_1\sigma^2}$.

Tarvittavat todennäköisyydet estimoidaan soveltamalla Bayesin kaavaa

$$P(\alpha|\delta) = P(\alpha)P(\delta|\alpha) \quad (38)$$

Tällöin siis 1:1-linjauksen suuri prioritodennäköisyys (90 %) aiheuttaa preferenssiä sen valintaan.

Rekursiivinen kustannusten laskenta-algoritmi on hidas, jos tekstinpätkät ovat pitkiä. Yksittäisillä kappaleilla kuitenkin suhteellisen nopea.

Menetelmä toimii melko hyvin sukukielillä: raportoitu 4% virhemäärä. Kun lisäksi pyrittiin erikseen tunnistamaan epäilyttävät linjaukset, ja linjaamaan vain parhaat 80% päästiin virhetasoon 0.7

Menetelmä toimii parhaiten 1:1-linjauksilla (2%), mutta hankalammille linjauksille virheprosentit ovat suuria.

7.4.6 Church, 1993: Identtisiin merkkijonoihin perustuva menetelmä

Edelliset menetelmät eivät sovellu kohinaiseen tekstiin (esim. optisen tekstintunnistuksen tuottamaan), jossa saattaa olla roskaa välissä tai kokonaan kadonneita kappaleita. Myös kappale- ja lauserajat ovat vaikeita havaita mm. kadonneiden välimerkkien tai roskan takia.

Tämän menetelmän perustana oleva huomio:

Teksteissä, jotka on kirjoitettu jokseenkin samalla aakkostolla (esim. roomalaiset aakkoset), esiintyy samaatarkoittavia, identtisiä kirjainsekvenssejä kuten erisnimiä tai numeroita.

Sukulaiskielillä, tai läheisessä vuorovaikutuksessa olevilla kielillä voi lisäksi esiintyä muitakin yhteisiä sekvenssejä johtuen yhteisestä kantamuodosta (esim. englannin 'superior' ja ranskan 'supérieur') tai lainasanoista.

Lasketaan identtisiä merkki-n-grammeja (n esim. 4). Etsitään n-grammien linjaus joka sisältää mahdollisimman paljon identtisiä n-grammipareja. Lisäksi n-grammeja voidaan painottaa frekvenssin mukaan.

Menetelmä ei tuota varsinaista lauseiden jyvitystä.

Voi epäonnistua täydellisesti mikäli kielissä ei ole riittävästi yhteisiä merkkijonoja.

Leksikaaliset menetelmät

Tavoitteena on tuottaa aito lausetason 'jyvitys'.

Vaikuttaa selvältä, että tieto sanojen todennäköisistä käännöspareista auttaisi linjausta huomattavasti.

Puhtaasti tilastollisten menetelmien keskeinen ajatus:
vuorotellaan todennäköisen osittaislinjauksen tekemistä sanatasolla ja todennäköisimmän lausetason linjauksen tekemistä.

Apuna käytetään lisäksi oletusta, että toisiaan vastaavat lausejonot eivät luultavasti ole kovin kaukana toisistaan (esim. ristiinmenoja ei ole tai ne eivät ole pitkiä).

Iteraatioita ei yleensä tarvita kovin monta (johtuen yo. rajoituksesta).

Variantteja

Chen, 1993: Sovelletaan yksinkertaista sana-sana-käännösmallia estimoimaan sanaparien käännöstn:t. Lasketaan tällä mallilla maksimaalisen todennäköinen linjaus.

Hyviä puolia: aidon käännösmallin käyttö oletettavasti parantaa tarkkuutta puoliheuristisiin menetelmiin verrattuna. Yksinkertaisen käännösmallin käyttö tekee menetelmästä laskennallisesti tehokkaan.

Huono puoli: mallin yksinkertaisuuden aiheuttamat approksimaatiot voivat aiheuttaa ongelmia kun pitävät huonosti paikkaansa jollekin kieliparille tai tekstiparille.

Menetelmää on sovellettu suurten korpusten (useita miljoonia lauseita) linjaamiseen. Sen virheprosentiksi on estimoitu 0.4% mikä on yhtä hyvä tai parempi kuin muilla menetelmillä saman korpuksen osajoukolla.

Fung ja McKeown, 1994

Estimoidaan pieni kaksikielinen sanakirja, joka antaa todennäköisesti vastaavia sana-käännös-pareja. Käytetään näitä 'ankkureina' linjauksessa, kuten aiemmin käytettiin yhteisiä n-grammeja.

Ei tarvita yhteisiä sanoja tai n-grammeja.

7.4.7 Sanojen linjaus ja kaksikielisten sanakirjojen estimointi

Sanatason linjauksen peruslähestymistapa: vuorotellaan seuraavia askeleita:

1. muodostetaan jokin sanatason linjaus
2. estimoidaan sen perusteella sanaparien käännöstodennäköisyydet

Sovelletaan siis EM-tyyppistä algoritmia.

Kaksikieliseen sanakirjaan hyväksytään (lopulta) vain sanaparit, joista on saatu riittävästi evidenssiä eli esim. riittävän monta näytettä kyseisten sanojen vastaavuudesta.

Voidaan olettaa, että jatkossa sanojen (ja lauseiden) linjauksessa käytetään myös kaksikielisten sanakirjojen sisältämää tietoa.

Suoraviivainen sovellustapa olisi käyttää tällaista sanakirjaa initialisoimaan edellä esitetyn algoritmin sanaparien käännöstodennäköisyydet.

8 Kontekstitieto ja yhteisesiintyminen

- Kontekstin tärkeys kielen tulkinnassa: esimerkiksi monitulkintaisuudet ("Aloitin alusta", "Alusta kovalevy!", "Näin monta alusta", "Minä näin monta alusta")
- Chomskyn hierarkia kielille
- Kontekstin pituus
- Yhteisesiintymismatriisi
- Kollokaatiot

8.1 Chomskyn hierarkia kielille

Chomsky jakaa kielet seuraavanlaiseen kompleksisuushierarkiaan.

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 $A \rightarrow x$

8.1.1 Chomskyn hierarkia kielille, selitykset

Tyyppi 0 vastaavat niitä kieliä, joiden symbolijonot voidaan tuottaa (listata) Turing-koneella.

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.

8.2 Kontekstin pituus ja yhteisesiintymismatriisi

- n-grammit, dynaaminen konteksti
- yksittäiset sanat kontekstissa, sana-dokumenttimatriisi
- bag of words -malli versus sanapositioiden huomioiminen
- etäriippuvuudet ja lauserakenteen huomioiminen

8.3 Kollokaatiot

- Kollokaatio on kahdesta tai useammasta sanasta koostuva konventionaalistunut ilmaus
- Esimerkkejä:
  - 'weapons of mass destruction', 'disk drive', 'part of speech' (suomessa yhdyssanoina 'joukkotuhoaseet', 'levyasema', 'sanaluokkatieto')
  - 'bacon and eggs'
  - verbin valinta: 'make a decision' ei 'take a decision'.

  - adjektiivin valinta: 'strong tea' mutta ei 'powerful tea'; 'vahvaa teetä', harvemmin 'voimakasta teetä' (valinnat voivat heijastaa kulttuurin asenteita: strong → tea, coffee, cigarettes powerful → drugs, antidote)
  - 'kick the bucket', 'heittää veivinsä' (kiertoilmaus, sanonta, idiomi)

- Olentoja, yhteisöjä, paikkoja tai tapahtumia yksilöivät nimet: 'White House' Valkoinen talo, 'Tarja Halonen'

- Kollokaation kanssa osittain päällekkäisiä käsitteitä: termi, tekninen termi, terminologinen fraasi. Huom: tiedonhaussa sanalla 'termi' laajempi merkitys: 'sana tai kollokaatio'.

8.3.1 Sanan frekvenssi ja sanaluokkasuodatus

Pelkän frekvenssin käyttö:

Esimerkki: Onko luontevampaa sanoa 'strong tea' vai 'powerful tea'?
Ratkaisu: Etsitään Googlella: 'strong tea' 9270, 'powerful tea' 201

Joihinkin täsmällisiin kysymyksiin riittävä tapa. Kuitenkin järjestettäess bigrammeja frekvenssin mukaan, parhaita ovat 'of the', 'in the', 'to the', ...

Frekvenssi + sanaluokka:

Jos tunnetaan kunkin sanan sanaluokka, sekä osataan kuvailla kollokaatioiden 'sallitut' sanaluokkahahmot:

- Järjestetään sanaparit tai -kolmikot yleisyyden (lukumäärä) mukaan

- Hyväksytään vain tietyt sanaluokkahahmot:
  AN, NN, AAN, ANN, NAN, NNN, NPN (Justeson & Katz's POS filter)

$C(w^1\ w^2)$	w^1	w^2	Tag Pattern
11487	New	York	A N
7261	United	States	A N
5412	Los	Angeles	N N
3301	last	year	A N
3191	Saudi	Arabia	N N
2699	last	week	A N
2514	vice	president	A N
2378	Persian	Gulf	A N
2161	San	Francisco	N N
2106	President	Bush	N N
2001	Middle	East	A N
1942	Saddam	Hussein	N N
1867	Soviet	Union	A N
1850	White	House	A N
1633	United	Nations	A N
1337	York	City	N N
1328	oil	prices	N N
1210	next	year	A N
1074	chief	executive	A N
1073	real	estate	A N

Table 5.3 Finding Collocations: Justeson and Katz' part-of-speech filter.

8.3.2 Sanojen etäisyyden keskiarvo ja varianssi

Entä joustavammat kollokaatiot, joiden keskellä on kollokaatioon kuulumattomia sanoja?

Lasketaan etäisyyden keskiarvo ja varianssi. Jos keskiarvo nollasta poikkeava ja varianssi pieni, potentiaalinen kollokaatio (Huom: oletetaan siis etäisyyden jakautuvan gaussisesti).

Esim. '*knock ... door*' (ei 'hit', 'beat', tai 'rap'):
a) 'She *knocked* on his *door*'
b) 'They *knocked* at the *door*'
c) '100 women *knocked* on Donaldson's *door*'
d) 'a man *knocked* on the metal front *door*'

8.3.3 Algoritmi

- Liu'uta kiinteän kokoista ikkunaa tekstin yli (leveys esim. 9) ja kerää kaikki sanaparin esiintymät koko tekstissä

- Laske sanojen etäisyyksien keskiarvo:
  $\bar{d} = 1/n \sum_{i=1}^{n} d_i = 1/4(3+3+5+5) = 4.0$
  (jos heittomerkki ja 's' lasketaan sanoiksi)

- Estimoi varianssi s^2 (pienillä näytemäärillä):
  $s^2 = \frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1} = 1/3((3-4.0)^2 + (3-4.0)^2 + (5-4.0)^2 + (5-4.0)^2)$
  $s = 1.15$

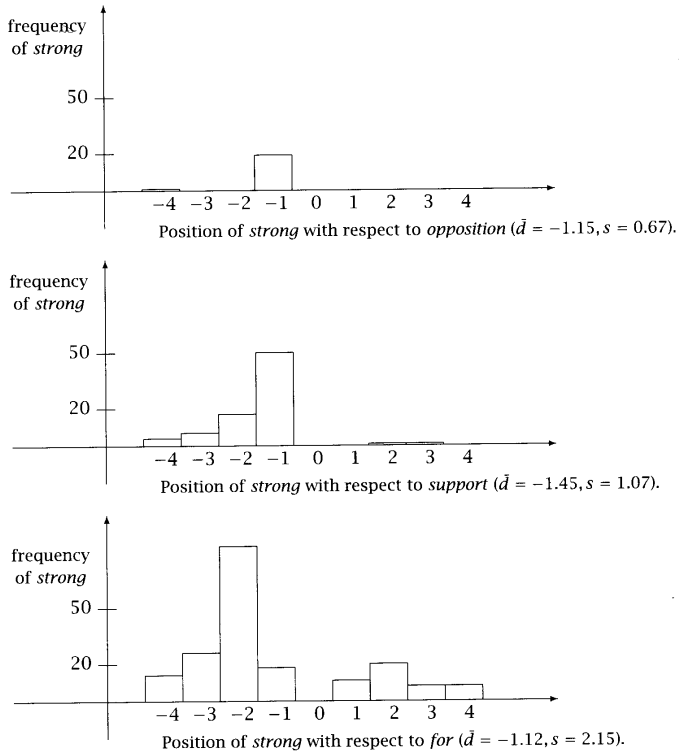


Figure 5.2 Histograms of the position of *strong* relative to three words.

s	$\bar{d}$	Count	Word 1	Word 2
0.43	0.97	11657	New	York
0.48	1.83	24	previous	games
0.15	2.98	46	minus	points
0.49	3.87	131	hundreds	dollars
4.03	0.44	36	editorial	Atlanta
4.03	0.00	78	ring	New
3.96	0.19	119	point	hundredth
3.96	0.29	106	subscribers	by
1.07	1.45	80	strong	support
1.13	2.57	7	powerful	organizations
1.01	2.00	112	Richard	Nixon
1.05	0.00	10	Garrison	said

Table 5.5 Finding collocations based on mean and variance. Sample deviation s and sample mean $\bar{d}$ of the distances between 12 word pairs.

Pohdittavaksi:

1. Mitä tapahtuu jos sanoilla on kaksi tai useampia tyypillisiä positioita suhteessa toisiinsa?
2. Mikä merkitys on ikkunan leveydellä?

8.3.4 Hypoteesin testaus

Onko suuri osumamäärä yhteensattumaa (esim. johtuen siitä että jommankumman perusfrekvenssi on suuri)? Osuvatko kaksi sanaa yhteen useammin kuin sattuma antaisi olettaa?

1. Formuloi *nollahypoteesi* H_0: assosiaatio on sattumaa
2. Laske tn p että sanat esiintyvät yhdessä jos H_0 on tosi
3. Hylkää H_0 jos p liian alhainen, alle merkitsevyystason, esim $p < 0.05$ tai $p < 0.01$.

Nollahypoteesia varten sovelletaan riippumattomuuden määritelmää.

Oletetaan että sanaparin todennäköisyys, jos H_0 on tosi, on kummankin sanan oman todennäköisyyden tulo:
$P(w^1 w^2) = P(w^1)P(w^2)$

8.3.5 T-testi

Tilastollinen testi sille eroaako havaintojoukon odotusarvo oletetun, datan generoineen jakauman odotusarvosta. Olettaa, että todennäköisyydet ovat suun-

nilleen normaalijakautuneita.

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}, jossa \tag{39}$$

$\bar{x}, s^2$: näytejoukon keskiarvo ja varianssi, N = näytteiden lukumäärä, ja μ = jakauman keskiarvo. Valitaan haluttu p-taso (0.05 tai pienempi). Luetaan tätä vastaava t:n yläraja taulukosta. Jos t suurempi, H_0 hylätään.

8.3.6 Soveltaminen kollokaatioihin:

Nollahypoteesina että sanojen yhteisosumat ovat satunnaisia: Esimerkki: H_0 : $P(new\ companies) = P(new)P(companies)$

$\mu = P(new)P(companies)$
$\bar{x} = \frac{c(new\ companies)}{c(\cdot,\cdot)} = \hat{p}$
$s^2 = p(1-p) = \hat{p}(1-\hat{p}) \approx \hat{p}$ (pätee Bernoulli-jakaumalle)
$N = c(\cdot,\cdot)$

- Järjestetään sanat paremmuusjärjestykseen mitan mielessä TAI
- Hypoteesin testaus: valitaan merkittävyystaso (p=0.05 tai p=0.01) ja katsotaan t-testin taulukosta arvo, jonka ylittäminen tarkoittaa nollahypoteesin hylkäystä.

Vertaillaan yhtä suuren frekvenssin omaavia bigrammeja keskenään t-testillä:

t	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
4.4721	42	20	20	Ayatollah	Ruhollah
4.4721	41	27	20	Bette	Midler
4.4720	30	117	20	Agatha	Christie
4.4720	77	59	20	videocassette	recorder
4.4720	24	320	20	unsalted	butter
2.3714	14907	9017	20	first	made
2.2446	13484	10570	20	over	many
1.3685	14734	13478	20	into	them
1.2176	14093	14776	20	like	people
0.8036	15019	15629	20	time	last

Table 5.6 Finding collocations: The t test applied to 10 bigrams that occur with frequency 20.

Esimerkki soveltamisesta muuhun ongelmaan: Vertailu mitkä lähikontekstin sanat parhaiten erottelevat sanoja 'strong' ja 'powerful'

t	$C(w)$	$C(\textit{strong}\ w)$	$C(\textit{powerful}\ w)$	Word
3.1622	933	0	10	computers
2.8284	2337	0	8	computer
2.4494	289	0	6	symbol
2.4494	588	0	6	machines
2.2360	2266	0	5	Germany
2.2360	3745	0	5	nation
2.2360	395	0	5	chip
2.1828	3418	4	13	force
2.0000	1403	0	4	friends
2.0000	267	0	4	neighbor
7.0710	3685	50	0	support
6.3257	3616	58	7	enough
4.6904	986	22	0	safety
4.5825	3741	21	0	sales
4.0249	1093	19	1	opposition
3.9000	802	18	1	showing
3.9000	1641	18	1	sense
3.7416	2501	14	0	defense
3.6055	851	13	0	gains
3.6055	832	13	0	criticism

Table 5.7 Words that occur significantly more often with *powerful* (the first ten words) and *strong* (the last ten words).

9 Sananmerkitysten yksikäsitteistäminen (word sense disambiguation, WSD)

Ongelman määrittely: Oletetaan sana w, jolle olemassa k erillistä merkitystä $s_1 \ldots s_k$. Tehtävänä on päätellä yksittäisen esiintymän osalta mikä merkitys on kyseessä. Kyse on siis *hahmontunnistuksesta* tai *luokittelusta*.

9.1 Hyödyllisiä aineistotyyppejä

- Sense-tagged -korpus: aineisto, johon on jokaisen sanan w esiintymän kohdalle tagattu kyseisen esiintymän merkitys s_i.
  Esim. Senseval (englanninkielinen).
- Sanakirjat ja tesaurukset, ks. esim. sana 'shake' (http://www.britannica.com/)
- Kaksikielinen 'kohdistettu' (linjattu) aineisto: sama aineisto molemmilla kielillä, johon on merkitty toisiaan vastaavat kohdat, esim. sana tai lause kerrallaan.

- Yksikielinen aineisto, jossa on 'siemeneksi' sense-tagattu (merkitysmerkitty) pieni osa sanan esiintymistä
- Yksikielinen aineisto, jossa paljon sanan esiintymiä kontekstissa

Korpukset voivat myös olla syntaktisesti tagattuja (sanaluokat jne.) tai sisältää pelkän tekstin.

9.2 Eri oppimisperiaatteista

Tunnistusmenetelmät voidaan jakaa ryhmiin mm. oppimisperiaatteen mukaan:

- Ohjaamaton (unsupervised) oppiminen
- Vahvistettu (reinforced) oppiminen
- Ohjattu (supervised) oppiminen

Seuraavaksi tarkastellaan tarkemmin ohjaamatonta ja ohjattua oppimista.

9.2.1 Ohjaamaton oppiminen

- Hahmojen *luokkia ei tiedetä* etukäteen
- Tavoitteena on muodostaa hahmoista ryhmiä, joiden sisällä hahmot ovat samankaltaisia ja joiden välillä on selkeitä eroja (klusterointi)
- Optimoitava funktio on klusteroinnin onnistumista kuvaava mitta
- Aina ei tiedetä edes ryhmien lukumäärää

9.2.2 Ohjattu oppiminen

- Hahmojen *luokat tunnetaan* etukäteen
- Tavoitteena on muodostaa kuvaus piirreavaruudesta luokka-avaruuteen
- Optimoitava funktio perustuu kuvauksessa tapahtuviin virheisiin, ts. pyritään minimoimaan tapahtuvien *luokitteluvirheiden todennäköisyys* tai, mikäli virheisiin liittyy toisistaan poikkeavia kustannuksia, virheiden kokonaiskustannuksen odotusarvo.

9.2.3 Bootstrapping

Luonnollisen kielen aineistoilla relevanttia on lisäksi ns. 'bootstrap' -oppiminen: Pieni osa aineistosta on luokiteltua, jonka avulla päästään alkuun. Tämän jälkeen oppiminen tapahtuu ohjaamattomasti.

9.3 Menetelmien onnistumisen mittaaminen

9.3.1 Keinotekoinen data: pseudosanat

- Menetelmiä voidaan kehittää ja testata keinotekoisella datalla, jonka ominaisuudet varmasti tunnetaan.
- Esim. korvataan kaikki sanojen 'banaani' ja 'ovi' esiintymät pseudosanalla 'banaaniovi'. Mitataan, kuinka hyvin onnistutaan tunnistamaan kutakin 'banaaniovea' vastaava oikea sana.
- Kyseessä on helppo, halpa ja nopea tuottaa laajoja testiaineistoja, joissa tunnetaan sekä disambiguoimaton data että alkuperäinen - oikea - data, joka menetelmän pitäisi löytää.

9.3.2 Onnistumisen laskennalliset ylä- ja alarajat

Mikäli yhteisiä testiaineistoja ei ole, pelkkä numeerinen tulos ei riitä menetelmien onnistumisen mittaamiseen: jotkut ongelmat ovat luonnostaan vaikeampia kuin toiset.

Pyritään siksi hahmottamaan ongelman vaikeus:

- Yläraja 'ground truth': paras mahdollinen tulos. Usein käytetään mittana ihmisen suoriutumista samasta tehtävästä (harvoin 100%).
- Ylärajan määritys tärkeää esim. jos verrataan menetelmien suoriutumista rajallisen mittaisella kontekstilla. Harvoin 100% esimerkiksi, jos ikkuna kovin kapea.
- Alaraja, 'baseline': yksinkertaisin mahdollinen perusmenetelmä. Esim. luokan valinta satunnaisesti tai luokan taustafrekvenssin perusteella.

9.4 Ohjattu disambiguointi

Käytetään notaatiota:

w	monimerkityksinen sana
$s_1 \dots s_K$	sanan eri merkitykset (senses)
$c_1 \dots c_I$	sanan w kontekstit korpuksessa
$v_1 \dots v_J$	piirrejoukko (esim. joukko sanoja), jota käytetään disambiguoinnissa

Seuraavaksi esitellään joitain ohjatun oppimisen lähestymistapoja, joita on sovellettu merkitysten disambiguointiongelmaan.

9.4.1 Piirteiden valinta

- Yleisesti piirrejoukko vaikuttaa suuresti luokittelun onnistumismahdollisuuksiin.
- Hyvä piirrejoukko on riippuvainen käytetyn luokittimen (tai mallin) ominaisuuksista, eli piirrejoukon valintaa ja mallinnusta ei voida täysin erottaa toisistaan.

9.4.2 Esimerkkejä mahdollisista piirteistä

- tietyn sanan esiintyminen jonkin etäisyyden päässä disambiguoitavasta sanasta, esim. etäisyys(w,'avasi') < 3
- tiettyjen kahden sanan esiintyminen kontekstissa yhdessä
- tietyn sanaluokan tai morfologisen luokan esiintymisfrekvenssi kontekstiikkunassa (jos data on POS- tai morfol. tagattua)
- tietyn sanan tai sanaluokan esiintyminen tietyssä täsmällisessä positiossa suhteessa disambiguoitavaan sanaan (esim. edeltävänä sanana)
- tieto jonkin semanttisen muuttujan, esim. keskustelunaihe, arvosta
- jokin ylläolevien funktio tai yhdistelmä

9.4.3 Bayesläinen luokitin

- Luokitin ei tee piirrevalintaa, vain yhdistää evidenssin eri piirteistä
- Valitaan piirteiksi joukko sanoja
- Luokitin soveltaa *Bayesin päätössääntöä* valitessaan luokan, ts. minimoi luokitteluvirheen todennäköisyyttä:

$$P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)} \tag{40}$$

  $P(s_k)$ on merkityksen s_k *prioritodennäköisyys*, eli tn jos emme tiedä kontekstista mitään.

9.4.4 Bayesläinen luokitin: todennäköisimmän luokan valinta

- Jos tehtävänä on vain valita todennäköisin luokka, voidaan jättää kontekstin c todennäköisyys $P(c)$ (joka ei riipu luokasta) laskuissa huomiotta: valitaan merkitys s' jos

$$
\begin{aligned}
s' &= arg\max_{s_k} P(s_k|c) && (41)\\
&= arg\max_{s_k} \frac{P(c|s_k)P(s_k)}{P(c)} && (42)\\
&= arg\max_{s_k} P(c|s_k)P(s_k) && (43)\\
&= arg\max_{s_k} [\log P(c|s_k) + \log P(s_k)] && (44)
\end{aligned}
$$

9.4.5 Estimointiongelma

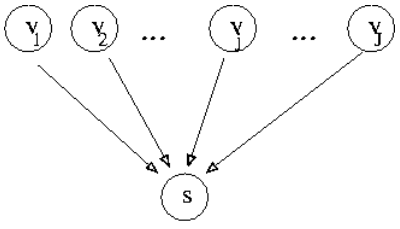
- Käytännön hankaluus: kontekstin piirteiden ehdollisen yhteistnjakauman $P(c|s_k)$ luotettava estimointi tietylle merkitykselle edellyttäisi, että meillä olisi datajoukko, jossa jokainen merkitys esiintyisi kaikissa periaatteessa mahdollisissa konteksteissaan, mieluiten useita kertoja.
- Ratkaisu: helpotetaan estimointia tekemällä sopivia yksinkertaistavia oletuksia (Naive Bayes, Naïve Bayes)

9.4.6 Naive Bayes -luokitin

- *Naive Bayes -oletuksessa* lähdetään siitä, että kukin piirre vaikuttaa luokitukseen toisista piirteistä riippumattomasti:

$$P(c|s_k) = P(v_1, \ldots, v_J|s_k) = \prod_{v_j inc} P(v_j|s_k) \quad (45)$$

Sama graafisesti:



- Tässä yksinkertaistetussa mallissa (jota kutsutaan myös 'bag of words'-malliksi) sanojen järjestyksellä kontekstissa ei ole merkitystä, ja sama sana voi esiintyä kontekstissa useita kertoja.

- Sovellettaessa edelliseen päätössääntöön (kaava 44) saadaan *Naive Bayes päätössääntö*:

$$Valitaans'joss' = arg\max_{s_k}[\log P(s_k) + \sum_{v_j inc} P(v_j|s_k)] \qquad (46)$$

- Näistä $P(v_j|s_k)$:lle ja $P(s_k)$:lle lasketaan ML-estimaatit luokitellusta opetusdatajoukosta:

$$\begin{aligned} P(v_j|s_k) &= \frac{C(v_j, s_k)}{C(s_k)} \\ P(s_k) &= \frac{C(s_k)}{C(w)} \end{aligned}$$

  jossa C(...) tarkoittaa lukumäärää opetusdatajoukossa

- Kuuden substantiivin *duty, drug, land, language, position, sentence* luokituksessa kun aineistona oli Hansard-korpus saatiin 90% tunnistustulos (Church, Gale & Yarowsky, 1992).

- Esimerkkejä 'drug'-sanan merkityksille sovelletuista piirteistä:

Merkitys	Piirteet ko. merkitykselle
medication	prices, prescription, patent, increase, consumer, pharmaceutical
illegal substance	abuse, paraphernalia, illicit, alcohol, cocaine, traffickers

9.4.7 Naive Bayes -luokitin, yhteenveto

- Naive Bayes-luokitin on yksinkertainen ja melko robusti; antaa kohtuullisia tuloksia monenlaisissa ongelmissa.

- Naive Bayes -ongelma: epärealistiset riippumattomuusoletukset

- ML-estimoinnin ongelma: käyttää kaikki piirteet, ts. ei kykene tekemään piirrevalintaa

9.4.8 Eräs informaatioteoreettinen lähestymistapa

- Edellisessä mallissa käytettiin kaikki kontekstin sanat estimoinnissa.

- Nyt päinvastainen lähestymistapa: valitaan yksittäinen mahdollisimman hyvä indikaattori, jonka arvo voidaan selvittää kustakin kontekstista kullekin merkitykselle.

- Esimerkki: Ranskan 'prendre'-sanan merkitykset 'tehdä päätös' (*prendre une decision* ja 'ottaa mitta' (*prendre une mesure*, luotettava indikaattori olisi verbin objektina oleva sana.
- Disambiguoitava sana: $w = prendre$
  Valitaan piirrejoukoksi (indikaattoriksi) esim. objektipositiossa olevat sanat
  $V = \{mesure,\ note,\ exemple,\ décision,\ parole\}$
  Käännösten joukko: $K = \{take, make, rise, speak\}$

9.4.9 Menetelmän kuvaus

Maksimoidaan yhteisinformaatio piirteen ja merkityksen välillä. Muistellaan yhteisinformaation kaavaa:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{47}$$

partitiointi = jonkin joukon jako osajoukkoihin.

9.4.10 Flip-Flop -algoritmi

- K= käännösten joukko, V= piirteiden joukko
- Valitse V:lle satunnaisesti jokin partitiointi Y
- Toista, kunnes parannus ei ole enää suuri:
  1. Etsi K:lle partitiointi X siten, että $I(X;Y)$ maksimoituu
  2. Etsi V:lle partitiointi Y siten, että $I(X;Y)$ maksimoituu

Esim. jos partitioidaan vain kahteen ryhmään:
Käännettävä sana: prendre
Mahdolliset käännökset $K = \{take, rise, make, speak\}$
Objektiposition piirteet $V = \{mesure,\ note,\ exemple,\ décision,\ parole\}$

9.4.11 Esimerkki

Data:

- 'prendre une mesure ': take measure,
- 'prendre notes': take notes,
- 'prendre exemple': take an example

- 'prendre une decision': 'make a decision'
- 'prendre la parole': 'rise to speak'

Paras partitiointi:

- $X_1 = \{take\}$, $X_2 = \{rise, make, speak\}$
- $Y_1 = \{mesure, note, exemple\}, Y_2 = \{decision, parole\}$

9.4.12 Kommentteja flip-flop -algoritmia koskien

- Jos tehdään täyshaku eli kokeillaan kaikki partitioinnit kummallekin joukolle, K ja V, menee exponentiaalinen aika
- Flip-Flop-algoritmi kuitenkin vie lineaarisen ajan
- Toistetaan Flip-Flop-algoritmi eri indikaattorelle (objekti, määre, edellinen sana, jne); valitaan indikaattori, joka maksimoi yhteisinformaation
- **Huom.** Käännöksistä saadut 'labelit' eivät välttämättä sanan eri merkityksiä vaan osaksi saman merkityksen eri ilmenemismuotoja (ks. sanan-muotojen variaatio, esim. 'cent' viitatessaan merkitykseen senttimetri voisi ilmetä suomeksi muodoissa ('cm','sentti'). Partitioidessaan myös eri labelien joukon S algoritmi itse asiassa ryhmittelee nämä eri muodot pienemmäksi joukoksi merkityksiä.
- Kääntämisen kannalta katsottuna tehtävä on siis ohjatun oppimisen tehtävä. Merkitysten etsimisen (esim. niiden oikea lukumäärä) ja disambiguoinnin kannalta taas ohjaamatonta oppimista.

9.5 Sanakirjapohjainen disambiguointi

9.5.1 Sanakirjamerkitysmäärittelyihin perustuva menetelmä

Notaatio ja piirteet:

sanan kuvaus	symboli	eri merkitykset	niiden määritelmät
tulkittava sana	w	$s_1 \dots s_K$	$D_1 \dots D_K$
w:n kontekstin sana j	v_j	$s_{j_1} \dots s_{j_L}$	$D_{j_1} \dots D_{j_L}$

Piirrejoukko: $E_{v_j} = \bigcup_{j_i} D_{j_i}$ eli ei välitetä kontekstin sanojen monimerkityksisyyksistä (yhdistetään määritelmät sanan w piirteitä laskettaessa).

Valintakriteeri: $s' = arg\max_{s_k} \bigcap(D_k, E_{v_j})$

Huom: Tämä on matemaattisesti sama kuin vektoriavaruusmenetelmä binääriarvoisilla, normalisoimattomilla vektoreilla. Myös paremmat samankaltaisuusmitat mahdollisia (piirteiden painotus ja vektorien pituuksien normalisointi).

9.5.2 Sanojen semanttisiin aiheluokkiin perustuva menetelmä

(kirjassa nimellä 'thesaurus-based disambiguation')

- Pohjana yleinen semanttinen luokitus (thesauruksessa, mm. Roget, tai muuten, mm. Longman)
- Luokat t aihealueita (topics),
  esim. {'urheilu', 'sota', 'musiikki', 'kalastaminen',... }
- score= monellako kontekstin sanalla löytyy yhteinen luokka sananmerkityksen s_k kanssa, ts. tässäkään ei käytetä normalisointeja tai painotuksia tai todennäköisyyksiä
- $$\text{score}(s_k) = \sum_{v_j \text{ in } c} \delta(t(s_k), v_j) \tag{48}$$
- $\delta(t(s_k), v_j) = 1 jos st(s_k)$ on jokin v_j:n luokista
- Oletetaan, että kukin merkitys s_k kuuluu täsmälleen yhteen luokista
- Sanan luokkien joukko on sen eri merkitysten luokkien unioni
  ts. ei yritä disambiguoida kontekstisanojen merkityksiä
- Jättää hyödyntämättä sanat, joita ei etukäteen sem. luokiteltu (esim. uudet sanat, jonain aikakautena kuuluisat henkilönnimet 'Navratilova', 'Jeltsin' jne., jotka voisivat olla oikein hyviäkin piirteitä)

9.5.3 2-kielisen aineiston käännöksiä hyödyntävä menetelmä

- tarvitaan: 2-kielinen sanakirja + toisen kielen korpus
- käännetään sana kaikilla eri tavoilla (kaikilla eri merkityksillä)
- käännetään sanan kontekstipiirre (yksinään, olet. vain yksi käännös)
- tarkastellaan käännösparien keskinäisiä frekvenssejä toisen kielen korpuksessa. Mikäli jokin vaihtoehto on riittävän todennäköinen, valitaan sen sisältämä tulkinta (testataan merkitsevyys, ja valitaan vain mikäli $p >$ esim. 90%).
- yleistyy suoraviivaisesti monen piirteen tutkimiselle

9.5.4 Yksi merkitys per aihe, yksi merkitys per kollokaatio

- Sanakirjapohjaiset menetelmät tarkastelivat jokaista sananesiintymää erillisenä
- Kuitenkin esiintymien välillä keskinäisiä riippuvuuksia
- Oletus 1: Yksi merkitys per aihe: sanalla yleensä yksi merkitys läpi koko dokumentin
- Oletus 2: Yksi merkitys per kollokaatio: sanan merkitys riippuu vahvasti aivan lähikontekstin sanoista, mukaanlukien sanojen järjestys ja sanaluokkatieto (ts. usein toisistaan täysin erilliset merkitykset ovat eri syntaktisessa ja/tai semanttisessa roolissa ympäristön suhteen)

9.6 Ohjaamaton merkitysten ryhmittely

- Edelläkuvatut menetelmät tarvitsevat leksikaalisia resursseja: sanakirjoja tai (pieniä) merkityksin tagattuja aineistoja jokaiselle monimerkityksisen sanan eri merkitykselle
- Aina sellaisia ei ole, esim. erikoistermien tai uusien merkitysten ilmaantuessa.
- Jos disambiguoinnilla tarkoitetaan merkitysten taggausta, täysin ohjaamattomasti ei voida disambiguoida.
- Voidaan kuitenkin klusteroida sanan esiintymät, ja toivoa/olettaa että kukin klusteri vastaa sanan yhtä merkitystä.
- Hyvä tai huono puoli: ryhmittely voi olla tarkempaa kuin esim. sanakirjoissa.
- Menetelmiä esim. EM-algoritmi, k-means, SOM, hierarkkiset klusterointimenetelmät, ...

9.6.1 EM-algoritmi disambiguoinnissa

Seuraavassa esitellään EM (Expectation Maximation) -algoritmin käyttö disambiguonnissa.

Esitetyissä kaavoissa K on eri merkitysten lukumäärä. $c_1, \ldots, c_i, \ldots, c_I$ ovat monitulkintaisen sanan kontekseja korpuksessa. $v_1, \ldots, v_j, \ldots, v_J$ ovat sanoja, joita käytetään piirteinä disambiguoinnissa.

9.6.2 EM-algoritmi disambiguoinnissa, alustusvaihe

- Alusta mallin μ parametrit satunnaisesti.
  Parametrit ovat
  - $P(v_j|s_k), 1{\leq}j{\leq}J, 1{\leq}k{\leq}K$ ja
  - $P(s_k), 1{\leq}k{\leq}K$.

  Laske log-likelihoodin arvo korpuksesta C annettuna malli μ. Arvo saadaan kertomalla keskenään yksittäisten kontekstien c_i todennäköisyydet $P(c_i)$, missä $P(c_i) = \sum_{k=1}^{K} P(c_i|s_k)P(s_k)$:

  $l(C|\mu) = log \prod_{i=1}^{I} \sum_{k=1}^{K} P(c_i|s_k)P(s_k) = \sum_{i=1}^{I} \sum_{k=1}^{K} P(c_i|s_k)P(s_k)$

9.6.3 EM-algoritmi disambiguoinnissa, EM-osuus

Niin kauan kuin $l(C|\mu)$:n arvo kasvaa, toistetaan E- ja M-askeleita.

- E-askel:

  Kun $1{\leq}j{\leq}J, 1{\leq}k{\leq}K$, estimoidaan h_{ik} eli posterioritodennäköisyys sille, että s_k generoi c_i:n seuraavasti:

  $h_{ik} = \frac{P(c_i|s_k)}{\sum_{k=1}^{K} P(c_i|s_k)}$

  Jotta $P(c_i|s_k)$ saadaan lasketuksi, tehdään Naive Bayes -oletus:

  $P(c_i|s_k) = \prod_{v_j \in c_j} P(v_j|s_k)$

9.6.4 EM-algoritmi disambiguoinnissa, M-askel

- M-askel:

  Estimoidaan uudelleen parametrit $P(v_j|s_k)$ ja $P(s_k)$ käyttämällä ML-estimointia:

  $P(v_j|s_k) = \frac{\sum_{i=1}^{I} \sum_{c_i:v_j \in c_i} h_{ik}}{Z_j}$

  Kaavassa $\sum_{c_i:v_j \in c_i}$ laskee summan kaikkien sellaisten kontekstien yli, joissa kontekstin sana v_j esiintyy. $Z_j = \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{c_i:v_j \in c_i} h_{ik}$ on normalisointivakio.

  Lasketaan uudelleen merkitysten todennäköisyydet seuraavasti:

  $P(s_k) = \frac{\sum_{i=1}^{I} h_{ik}}{\sum_{k=1}^{K} \sum_{i=1}^{I} h_{ik}} = \frac{\sum_{i=1}^{I} h_{ik}}{I}$

Huomioita EM-algoritmista

- Algoritmi on herkkä alustukselle.
- $l(C|\mu)$ paranee joka kierroksella
- Mitä suurempi määrä luokkia (merkityksiä), sen parempi $l(C|\mu)$. Ongelmana on siis ylioppiminen (jos soveltaa sellaisenaan eri luokkamäärien vertailuun).

9.6.5 Paluu lähtökuoppiin

Joissain tilanteissa on OK epäonnistua.

Esim. sanaleikit, 'In AI, much of the I is in the beholder'
Vrt. 'Beauty is in the eye of the beholder' (kauneus on katsojan silmässä)

Hypoteesi (Kilgariff): On tavallista että useat merkityksistä yhtäaikaa läsnä:

'For better or for worse, this would bring competition to the licenced trade'
competition - competitors vs. competition - the act of competing

Mahdollinen selitys: ihmiset eivät disambiguoi sanoja vaan tulkitsevat lauseita ja tekstejä. Jos kaksi sananmerkitystä johtavat samaan lauseen tulkintaan, sananmerkityksiä ei ole tarpeen eritellä.

Disambiguointiongelmia esim.

- systemaattinen polysemia (tekemisen akti vs. osallistujat/yhteisö)
- pisteen merkitys (lauseen loppu vs. muu)
- erisnimi vai yleisnimi 'Brown', 'Bush'
- etu- vai sukunimi 'Pentti Jaakko'

9.6.6 Muita menetelmiä

Muita ohjatun oppimisen menetelmiä: kNN

- valitaan k lähintä luokiteltua esimerkkiä, ja luokitellaan tämä esimerkki enemmistöäänestyksellä.
- Sopii hyvin harvalle datalle

- Edellyttää ainoastaan 'samankaltaisuuden' määrittelyn + mittauksen lähimpiin samankaltaisiin (kompleksisuus O(Nd) jossa N datan määrä ja d dimensio)

Muita ohjaamattoman oppimisen menetelmiä

Klusterointimenetelmiä:
K-means (Schütze soveltanut merkitysten ryhmittelyyn), SOM, hierarkkiset klusterointimenetelmät, erilaiset samankaltaisuusmitat

9.6.7 Menetelmien vertailua

- Senseval-projekti: laaja WSD-menetelmien vertailu yhteisellä datalla
- aineisto + evaluoinnit webissä

9.6.8 Kriittinen kommentti lähtökohtaoletuksesta

Alussa määriteltiin ongelma tähän tapaan: oletetaan sana w, jolle olemassa k erillistä merkitystä $s_1 \dots s_k$.

Kuitenkin on kyseenalaista, voidaanko sanojen eri merkityksiä jäännöksettömästi tarkastella "pistemäisinä", diskreetteinä oliona.