Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2003

Luennot: **Timo Honkela**

Laskuharjoitukset: **Vesa Siivola**

Luentokalvot:

Krista Lagus ja **Timo Honkela**

Luentomateriaali, osa 1/2

(Tämä materiaali ei ole täysin identtinen painetun materiaalin kanssa: kolme kaaviokuvaa puuttuu teknisten syiden takia. Painettu materiaali on toimitettu saataville kurssimateriaalina.)

Contents

1 YLEISTÄ KURSSISTA

1.1 Kurssin suorittaminen

Kurssi suoritetaan tekemällä harjoitustyö ja läpäisemällä tentti.

1.2 Ilmoittautuminen

Ilmoittautukaa kurssille www-topin avulla. Kieliteknologian opetuksen verkoston (KIT) opiskelijat voivat ilmoittautua sähköpostitse kurssin luennoijalle, kunnes saavat opintokirjannumeron TKK:lle.

1.3 Luennot

Luennot pidetään keskiviikkoisin kello 10–12 salissa T2.

Luennoitsija opettava tutkija fil.tri Timo Honkela.

Kalvot perustuvat Krista Laguksen vuonna 2002 pitämään kurssiin.

1.4 Laskuharjoitukset

Laskuharjoitukset pidetään tiistaisin kello 16-18. Ensimmäiset laskuharjoitukset pidetään 28.1.

Harjoitukset pitää DI Vesa Siivola.

1.5 Kirja

Kurssi seuraa kirjaa:

Christopher D. Manning, Hinrich Schtze:
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.

1.6 Luentomonisteet

Laskuharjoitukset ratkaisuineen ja luentokalvot ilmestyvät opetusmonisteina kurssin lopuksi. Laskuharjoituksissa paikan päällä jaetaan mallivastaukset,

jotka ovat myös opetusmonisteessa.

1.7 Suhde muihin opintoihin

Kurssi soveltuu osaksi seuraavia opintoja

- Kieliteknologian pää- ja sivuaine TKK:lla (Tik, Sähkö)
- Informaatiotekniikan pää- ja sivuaineen valinnaiset opinnot
- KIT-verkoston opinnot (mm. Helsingin yliopistossa)
- Muut aiheeseen liittyvät jatko-opinnot TKK:lla ja muualla (hyväksytettävä erikseen)

1.8 Tentin järjestelyt

Tentti järjestetään 5. toukokuuta klo 9-12 sali T1. Lisäksi syksyn tenttikausilla järjestetään yksi tai kaksi tenttiä.

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 viimeistään 2 päivää etukäteen.

1.9 Tenttikysymyksistä

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

Tehtävät tulevat painottumaan luentomonisteiden ja laskarien 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.

1.10 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 sumeus
- Kielen muuttuminen
- Tulkinnassa tarvittavan tietämyksen määrä ja laatu
- Multimodaalinen kommunikaatio
- Tulkinnan subjektiivisuus ja intersubjektiivisuus

2.8 Perinteisen lähestymistavan ongelmia, 1

'Kaikki kieliopit vuotavat' (Edward Sapir, 1921)

Täydellisen kuvauksen saavuttamisen esteinä ainakin kielellinen variaatio (yksilöiden ja kieliyhteisöjen välillä), luovuus, kielen muuttuminen.

Kritiikki 1: Onko kovan kieliopillinen - ei-kieliopillinen -rajan etsiminen hyvin määritelty ongelma, ts., onko sellaista rajaa edes olemassa, vai onko kyse aidosti sumeasta ilmiöstä?

On paljon lauseita joiden kieliopillisuudesta voidaan olla *montaa* mieltä, ja ollaankin. Todellisuudessa kovaa rajaa ei ehkä ole.

2.9 Perinteisen lähestymistavan ongelmia, 2

Kritiikki 2: Onko kieliopillisuus relevantti ja riittävä kielen kuvauksen taso?

Esim. lause 'Colourless green ideas sleep furiously.' (Chomsky) on syntaktisesti ok, mutta semanttisesti ei kovin mielekäs tai ainakaan tavanomainen.

Ratkaisuyritys: määritellään myös semanttisia sääntöjä. Ongelmia kuitenkin tulee, mm. sanojen metaforisen käytön kanssa. Ehkä 'kovat' säännöt ylipäänsä eivät ole oikea malliperhe?

Esimerkki:
Sääntö: niellä-sanan subjektina täytyy olla elävä olento
Lause: Supernova nielaisi planeetan.

2.10 Kategoriset (diskreetit) vs. jatkuvat representaatiot

- a/ä p/b: äänisignaalissa jatkuva muutos, foneemitasolla havainto on kategorinen: havaitaan joko a tai ä, ja havainto muuttuu yhtäkkisesti jossain kohti signaalin muuttuessa vähitellen.

Havaittaessa puhetta muutos jatkuvalta representaation tasolta (äänisignaali) diskreetiksi tai kategoriseksi (foneemi). Puhetta tuotettaessa päinvastainen muutos.

Todellisissa systeemeissä eroa diskreetin ja jatkuvan välillä ei ole, koska:

- Kaikki todelliset systeemit ovat kohinaisia (fysiikan perusteet)
- kohinainen kommunikaatikanava aina diskretoi signaalin

Sen sijaan aidosti relevantti kysymys on, onko representaatioavaruuden pisteiden välille määritelty etäisyysrelaatio (metriikka) vai ei. Usein tarkoitetaankin tätä silloin, kun puhutaan jatkuvista representaatioista.

2.11 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.

2.12 Probabilistisen esitystavan ja sumean esitystavan suhde

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

2.13 Perusteluja datasta oppimiselle, 1

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'.

2.14 Perusteluja datasta oppimiselle, 2

- 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.

2.14.1 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).

2.14.2 Riskejä ja haasteita:

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

2.15 Ihmisen kielikyky ja kielen oppiminen

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

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

Keskeisiltä osin ihmismielen ja kielen rakenne on kiinnitetty (oletettavasti geneettisesti määrätty). Perustelu: 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.

2.15.2 Empiristinen näkemys: Kieli opitaan, kielikyky toteutuu osana yleistä kognitiivista laitteistoa

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 kognitivisten oppimisperiaatteiden ja 3. sopivanlaisen stimulin 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?

2.15.3 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 Todennäköisyyslaskenta

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 $\sum_A P(A) = 1$

3.1.2 Esimerkki 1

Jos tasapainoista kolikkoa heitetään 3 kertaa, mikä on todennäköisyys että saadaan 2 kruunaa?

Mahdolliset heittosarjat Ω:
{ HHH, HHT, HTH, HTT, THH, THT, TTH, TTT }

Heittosarjat joissa 2 kruunaa: A={ HHT, HTH, THH}

Oletetaan tasajakauma: jokainen heittosarja yhtä todennäköinen, $P = 1/8$

$P(A) = \frac{|A|}{|\Omega|} = \frac{3}{8}$

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

Palataan esimerkkiin 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

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

3.3 Riippumattomuus

Kaksi tapahtumaa on tilastollisesti riippumattomia, jos niiden yhteinen todennäköisyys on sama kuin niiden erikseen tarkasteltujen todennäköisyyksien tulo:

P(A,B) =P(A)P(B)

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

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

Tämä voidaan johtaa hyödyntäen em. ehdollisen todennäköisyyden kaavaa:

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

3.3.1 Kausaalisuudesta ja riippuvudesta

Huom: tilastollinen riippuvuus ≠ kausaalinen riippuvuus!

Esim. jäätelön syönnin ja hukkumiskuolemien välillä voisi olla havaittavissa tilastollinen riippuvuus:

P('henkilö X hukkuu tänään','henkilö X on syönyt tänään jäätelöä') >
P('henkilö X hukkuu tänään')P('henkilö X on syönyt tänään jäätelöä')

Todennäköisyydet voisivat olla:

P('hukkuu tänään') = 0.001
P('hukkuu tänään'|'syönyt tänään jäätelöä') = 0.001
P('syönyt tänään jäätelöä') = 0.1

3.3.2 Ehdollinen riippumattomuus

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

A ja B ovat riippumattomia ehdolla C mikäli on niin että jos jo tiedämme C:n, tieto A:sta ei anna mitään lisätietoa B:stä (ja päinvastoin).

Edellisessä esimerkissä yhteinen kausaalinen tekijä on ehkä lämmin kesäsää:

P('hukkuu tänään','syönyt tänään jäätelöä'|'tänään lämmin ilma') =
P('hukkuu tänään'|'tänään lämmin ilma') ×
P('syönyt tänään jäätelöä'|'tänään lämmin ilma')

Todennäköisyydet voisivat olla:

P('hukkuu tänään','syönyt tänään jäätelöä'|'tänään lämmin ilma') = 0.001
P('hukkuu tänään'|'tänään lämmin ilma') = 0.001
P('syönyt tänään jäätelöä'|'tänään lämmin ilma') = 1.0

3.4 Bayesin kaava

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

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

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

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

3.4.1 Thomas Bayes 1702-1761

T. Bayes. An Essay Towards Solving a Problem in the Doctrine of Chances. Philosophical Transactions of the Royal Society of London, 53, pp. 370-418, 1763.

"I now send you an essay which I have found among the papers of our deceased friend Mr Bayes, and which, in my opinion, has great merit..."

3.4.2 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: $arg\max_B P(B|A) = arg\max_B \frac{P(B)P(A|B)}{P(A)} = arg\max_B P(B)P(A|B)$

3.4.3 Useampia kuin yksi ehto Bayesin kaavassa

$P(A)$ voidaan myös laskea useamman ehdon yhdistelmänä:

$P(A) = \sum_i P(A|B_i)P(B_i)$

Kannattaa huomata, että kaikille i $\neq$ j: $B_i \cap B_j = \emptyset$

3.5 Satunnaismuuttuja

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: $X : \Omega => 0, 1(Bernoulli - jakautunut)$.

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).

3.6 Odotusarvo ja varianssi

Odotusarvolle $E(X) = \sum_x xp(x)$

(diskreetissä tapauksessa; jatkuvassa tapauksessa summan korvaa integraali)

Ts. odotusarvo on keskiarvo kussakin näytteessä (kokeessa) saadun satunnaismuuttujan arvon yli.

Varianssi kuvaa muuttujan arvon vaihtelua keskiarvon ympärillä:

$$\begin{aligned} Var(X) &= E((X - E(X))) \\ &= E(X) - E(X) \end{aligned}$$

3.7 Yhteisjakauma

3.7.1 Yhteistodennäköisyys

P(X,Y) = kahden tapahtuman tai väitteen yhteistodennäköisyys, ts. että molemmat toteutuvat (esim. X=jonain tiettynä ajanhetkenä kuultu sananmuoto on 'siitä' ja Y=samana ajanhetkenä kuullun sanan sanaluokka on 'NOM'.)

3.7.2 Yhteistodennäköisyysjakauma

p(x,y) = kahden satunnaismuuttujan yhteisjakauma. Kuvaa x:n ja y:n kunkin arvokombinaation todennäköisyydet.

Kaavoja kootusti:

$$p(x,y) = p(X=x, Y=y) \quad \text{Yhteisjakauma} \quad (1)$$

$$p_X(x) = \sum_y p(x,y) \quad \text{Reunajakauma} \quad (2)$$

$$p(x,y) = p_X(x)p_Y(y) \quad \text{Riippumattomuus} \quad (3)$$

$$p_{X|Y}(x|y) = \frac{p(x,y)}{p_Y(y)} \quad \text{Ehdollinen jakauma} \quad (4)$$

$$p(x,y,z,w) = p(x)p(y|x)p(z|x,y)p(w|x,y,z) \quad \text{Ketjusääntö} \quad (5)$$

3.8 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.

3.9 Esimerkki diskreetistä jakaumasta: Binomijakauma

Notaatio: Jakauma(satunnaismuuttuja; jakauman parametrit)

Satunnaismuuttujalla 2 mahdollista arvoa, onnistuu/ei, tai tarkasteltava ominaisuus (esim. tietty sana) joko on tai ei ole läsnä jossain tietyssä näytteessä (esim. lauseessa).

p = Onnistumistodennäköisyys yksittäisessä kokeessa
r = onnistumisten lukumäärä kun kokeita yhteensä n
$b(r;n,p) = \frac{n!}{(n-r)!r!}p^r(1-p)^{n-r}$, jossa $0 \leq r \leq n$

Odotusarvo: np, varianssi: $np(1-p)$

Oletus: riippumattomat kokeet. Kieleen sovellettaessa kuitenkin edellinen ja seuraava lause yleensä riippuvat toisistaan (samoin sanat), joten kokeet eivät ole todella riippumattomia.

3.9.1 Binomijakauman kuvaajaesimerkkejä

matlab-koodi:

```
> n = 10; p=0.7; for r = 0:n
> binomi(r+1) =
     factorial(n) / (factorial(n-r)*factorial(r))*
                    (p .^ r)*((1-p) .^ (n-r));
  end
> x = 1:n+1; plot(x-1,binomi(x))
```

(Kuvat on jätetty tästä versiosta pois niiden suuren koon takia. Vastaavat kuvat on helppo tuottaa oheisella matlab-koodilla.)

3.9.2 Muita diskreettejä jakaumia

Multinomijakauma: Binomijakauman yleistys kun lopputuloksia voi olla useampi kuin kaksi.

Poisson-jakauma: Kiinteän kokoinen tapahtumaikkuna, jossa jakauma uvaa tietyn, tutkittavan, asian tapahtumis- tai esiintymislukumäärän todennäköisyydet. Satunnaismuuttuja x on siis tapahtumien lukumäärä tietyssä aikaikkunassa (tai jollakin matkalla, jollakin pinta-alalla tms.)

$b(r; m) = \frac{m^r}{r!} e^{-m}$

Parametri on n:n ja p:n tulo ($n \times p$). Jakauman varianssi ja keskiarvo on m

3.9.3 Poisson-jakauman sovellus

Tietyssä suuressa populaatiossa tiedetään aiemmin olleen 4 prosenttia erään kielen taitajia. Nykyistä tilannetta selvitettäessä valittiin populaatiosta satunnaisesti 200 henkilöä. Millä todennäköisyydellä 200 valitun joukossa on korkeintaan viisi k.o. kieltä osaavaa, jos populaatiossa osaajia on edelleen tuo 4 prosenttia.

Poisson-jakaumalla päästään kohtuulliseen likiarvoon. Nyt $n \times p = 200 \times 0.04 = 8$.

$P(X \leq 5) = \sum_{k=1}^{5} \frac{8^k}{k!} e^{-8} = 0.191$

3.9.4 Normaalijakauma (gaussinen jakauma)

Määritelty, jos tunnetaan keskiarvo μ ja varianssi σ :

$N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(x-\mu)/2\sigma^2}$

Yleisesti: todennäköisyysjakauma voi olla mikä tahansa funktio jonka integraali = 1 välillä $[0, 1]$

3.9.5 Normaalijakauman kuvaaja

Vastaava matlab-koodi:

```
> x = -4:.1:4;
> y = 1/(sqrt(2*pi))*exp(-(x.^2)/2);
> plot(x, y);
```

3.10 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.

3.10.1 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.

3.10.2 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.

3.11 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*: $P(\text{Malli}|\text{data})$

3.11.1 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. $P(s|M_m) = m^i(1-m)^j$
Frekventistisestä näkökulmasta, 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.

3.11.2 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 kruunia, 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: $P(M1|data) = \frac{P(data|M1)P(M1)}{P(data)}$

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

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

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

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

(Vastaukset eri heittosarjoilla: taulukko 2.1 kirjan sivulla 58)

3.12 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.

3.12.1 Entropia

Olkoon $p(x)$ satunnaismuuttujan X jakauma diskreetin symbolijoukon (aakkoston) A yli: $p(x) = P(X = x), x \in A$

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

3.12.2 Yhteisentropia ja ehdollinen entropia

Kahden muuttujan X ja Y (aakkostot A ja B) yhteisentropia, eli paljonko informaatiota keskimäärin tarvitaan kummankin arvon kommunikointiin:

$H(X,Y) = -\sum_{x\in A}\sum_{y\in B} p(x,y)\log p(X,Y)$

Ehdollinen entropia: Jos X on jo kommunikoitu, paljonko lisäinformaatiota keskimäärin tarvitaan Y:n kommunikoimiseen:

$$H(Y|X) = \sum_{x\in A} p(x)H(Y|X=x) \quad (6)$$

$$= -\sum_{x\in A}\sum_{y\in B} p(x,y)\log p(y|x) \quad (7)$$

Entropian ketjusääntö: $H(X,Y) = H(X) + H(Y|X)$

Erikoistapaus: Jos muuttujat riippumattomia toisistaan, kumpikin voidaan kommunikoida erikseen ja laskea koodinpituudet yhteen:
$H(X,Y) = H(X) + H(Y)$

Vrt. todennäköisyyksien ketjusääntö $P(X,Y) = P(X)P(Y|X)$
ja riipumattomille muuttujille $P(X.Y) = P(X)P(Y)$.

3.12.3 Yhteisinformaatio (Mutual Information, MI)

Yhteisinformaatio I muuttujien X ja Y välillä on $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

3.12.4 Kohinainen kanava-malli

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

Kanavan kapasiteetti C on tällöin: $C = \max_{p(X)} I(X;Y) = 1 - H(p)$ (kaavan johto kirjassa)

3.12.5 Relevanssi kielen mallintamisessa

Dekoodausongelmina voidaan tarkastella esimerkiksi

- konekäännöstä
- merkkien tunnistusta (OCR)
- puheentunnistusta

3.13 Minimum Description Length (MDL) -periaate

- Lähestymistapa mallin valintaan
- Rissanen et. al.
- Tavoite: pyritään löytämään datalle sellainen koodi että koko datajoukon koodauspituus minimoituu
- Koodinpituus = mallin kuvauspituus + datan kuvauspituus mallin avulla koodattuna + virheiden koodauspituus
- Koodinpituutta (todellista tai laskennallista) käytetään kustannusfunktiona mallia optimoitaessa
- Teoreettinen alaraja koodinpituudelle: entropia
- Suora yhteys myös bayesläiseen mallinnukseen

4 Yleisen kielitieteen perustietoja

4.1 Kielellisen analyysin eri tasoista

Käsiteltäviä kielellisiä yksiköitä

foneemi, morfeemi, sananmuoto, lekseemi, käsite, lause, virke, kappale, dokumentti, korpus

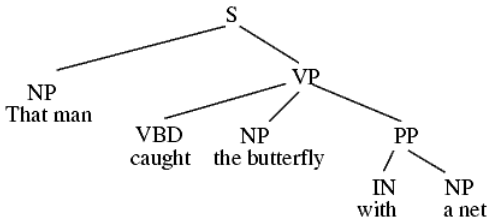
Tiedon lajeja, eri yksiköiden tasoilla

foneettinen ja fonologinen, morfologinen, syntaktinen, semanttinen, pragmaattinen, diskurssitieto, maailmantieto

4.1.1 Esimerkki syntaktisesta analyysista

Käsitteitä: Sanakategoriat, Lauserakennekielioppi, Dependenssikielioppi

Lauserakennekieliopin jäsennyspuu:



4.1.2 Esimerkki morfosyntaktisesta analyysistä

Tuotettu Connexorin FDG:llä
FDG=Functional Dependency Grammar
http://www.connexor.com

1	Haetaan	hakea	main: 0 &+MV V PASS IND PRES
2	rahtisatamasta	rahti#satama	sou: 1 &NH N SG ELA
3	lakkaa	lakka	obj: 1 &NH N SG PTV
4	,	,	
5	kun	kun	pm: 6 &CS CS
6	lakkaa	lakata	tmp: 1 &+MV V ACT IND PRES SG3
7	satamasta	sataa	obj: 6 &-MV V ACT INF3 SG ELA
8	.	.	

5 Korpustyöskentely

5.1 Välineitä ja tekniikoita

- Ohjelmointikieliä: Perl, Awk, Python, Prolog; isot korpukset käsitellään yleensä käyttäen ohjelmointikielenä C- tai C++ -kieltä tehokkuuden vuoksi
- Tekstihahmojen tunnistus: säännölliset lausekkeet (regexps)
- Sanojen koodaustapoja: sanojen korvaus numeroilla (taulukointi, hajautus (hash table))
- Frekvenssitietojen keräys

5.2 Kontekstivapaat kieliopit ja Prolog-kieli

- Prolog (PROgrammation LOGique) on nimensä mukaisesti logiikkaohjelmointikieli
- Prolog-kieltä hyödynnetään usein lauseenjäsennyksessä tai perinteisessä tietämyksen kuvaamisessa
- Prolog-kielen laajennuksena on monessa järjestelmässä mukana kontekstivapaiden ja niin sanottujen DFG (Definite Clause Grammar) -kielioppien kehittelyyn sopiva formalismi

5.3 Definite Clause Grammar -esimerkki

```
% DCG-kielioppi

sentence --> noun_phrase, verb_phrase.

noun_phrase --> determiner, noun.
noun_phrase --> noun.

verb_phrase --> verb.
verb_phrase --> verb, noun_phrase.


% Sanasto

determiner --> [the].
determiner --> [a].

noun --> [cat].
noun --> [cats].
```

```
noun --> [mouse].
noun --> [mice].

verb --> [scare].
verb --> [scares].
verb --> [hate].
verb --> [hates].
```

5.4 Tekstin esikäsittely

Tekstin esikäsittelyn tehtävänä on on mm.

- "roskan" poistaminen
- paloittelu käsiteltäviin yksiköihin (segmentointi tai tokenisointi)
- normalisointi: variaation eliminointi, monitulkintaisuuksien ratkominen

5.4.1 Roskan poistaminen

Kaikki mikä ei ole varsinaista tekstiä poistetaan, esim.

- Samoina toistuvat tiedot dokumenteissa (headerit, footerit)
- Kenttien nimet
- sähköpostiviestien headerit, signaturet

Perl-esimerkki (poista tekstit, jotka ovat sulkeiden sisällä):

```
#!/usr/bin/perl
while (<STDIN>) {
   s/\(.*?\)//g;
   print $_;
}
```

5.4.2 Paloittelu

Mikä on sana? Kahden tyhjän tilan (blanko, space) erottama alue? Puheessa kahden tauon erottama äänisignaali? Entä

- 'database' vs. 'data base'
- yhdyssanat: kesäilta, koti-ilta, venäläissotilaat, sivukonttori, elinkeinoelämä: elin keino elämä, elinkeino elämä, elin keinoelämä?

- rikas taivutus: huomaamattomuudellaansakaankohan
  huomaa mattomuu dellaansa kaanko han ?

- "John's": 1,2 vai 3 sanaa?
- puhelinnumerot, e-mail-osoitteet
- puheessa äänisignaalin tauot eivät osu sananrajoille vaan tietyille konsonanteille

Mikä on virke? (pisteen monitulkintaisuus, muut tavat lopettaa virke)

Mikä on dokumentti? Miten käsitellään pitkät dokumentit esim. indeksoitaessa tiedonhakua varten, paloina vai kokonaisina?

5.5 Variaatio informaation koodaustavoissa

Esim. käsite ITSEORGANISOIVA KARTTA; ilmauksia SOM, SOFM, self-organizing map, self-organising map, self organizing map, self-organizing feature map, Kohonen map, Kohonen SOM, Kohonen net.

Suomeksi: itseorganisoiva kartta, itseorganisoiTUva kartta, itsejärjestyvä kartta, Kohosen kartta, Kohosen verkko

Puhelinnumerojen koodaustapoja mm.:

040 123 4567	Suomi
040-123 4567	
040-1234567	
+358-40-1234567	
(040) 123 4567	
+411/284 3797	Sveitsi
(44.171) 830 1007	UK
+44 (0) 171 830 1007	
1-925-225-3000	USA
212.995.5402	

Information extraction, informaation ekstrahointi: yritetään oppia eri tavat ilmaista sama semanttinen informaatio. Tämä on *hahmontunnistusta*, pyrkimyksenä tunnistaa tietyn semanttisen informaation tyypit.

5.5.1 Morfologisen monimuotoisuuden käsittely

Sanan katkaisu 'juurimuotoon' (stemming)
'istuimme', 'istuttiin' .. → 'istu'

'yöt'→ 'yö'
'öisin'→ 'öi'

sanan perusmuotoistus
etsimme' → 'etsiä'
'lying' → 'lie'/'lay'
'istui', 'istuu', 'istunut', 'istumme' → 'istua'
'istuisitko', 'istuttaisiinko', 'ISTU!' → 'istua'

Kannattaako morfologinen variaation normalisointi tehdä, tai missä määrin?

Riippuu tavoitteesta, esim. halutaanko analysoida keskusteluja yksityiskohtaisesti (esim. interaktiivinen keskustelujärjestelmä) vai representoida pääasialliset puheenaiheet (esim. tiedonhaku).

5.6 Monitulkintaisuus (*ambiguity*)

Samalla sanalla tai ilmauksella voi olla monta erilaista tulkintaa:

- Englannin *title*: kirjan otsikko, elokuvan nimi, henkilön nimen etuliite tai arvonimi, omistusoikeus, jne
- *Kun lakkaa satamasta, hae lakkaa satamasta.*
- *'...pääsi perille...'* Montako mahdollista tulkintaa?

Edellisen kalvon monitulkintaisuuskysymykseen liittyen
(Lingsoftin TWOL)

<*pääsi*>	"pää" N NOM SG 2SG
	"pää" N GEN SG 2SG
	"pää" N NOM PL 2SG
	"päästä" V PAST ACT SG3
<*perille*>	"perä" N ALL PL
	"perille" ADV ALL
	"per" PROP N ALL SG

Mahdollisia tulkintoja ainakin:

- saapui sinne minne oli menossa
- saapui Per:in luo
- 'paina tämä asia pääsi perimmäiseen nurkkaan'
- 'näytitkö pääsi Perille?' (jos siinä oli vaikkapa haava ja Per on lääkäri)

5.6.1 Disambiguointi l. yksikäsitteistäminen

- Valitaan potentiaalisista tulkinnoista oikea tai todennäköisin
- Disambiguointi on tyypillinen kieliteknologinen tehtävä. Esim. morfologian yksikäsitteistäminen, lauserakenteen yksikäsitteistäminen, sananmerkitysten yksikäsitteistäminen
- Voidaan periaatteessa tehdä kontekstin perusteella; edellyttää *mallia* kontekstin ja vaihtoehtoisten tulkintojen välisistä riippuvuuksista.
- Tilastollinen malli: ehdollinen todennäköisyysjakauma $p(tulkinta|konteksti)$
- Ei-tilastollinen malli: kategoriset säännöt muotoa 'JOS konteksti NIIN tulkinta'.

5.7 Taggaus

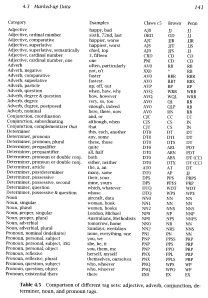
5.7.1 Syntaktisia tagijoukkoja Englannille

Brown, Penn, Claws 1-3

5.7.2 Taggauksen ääripäät eri tarkoituksiin

Puheentunnistus: Pelkästään puheessa esiintyvät sanat (ei välttämättä edes välimerkkejä)

Keskusteluanalyysi: vuoronvaihdot, vuorotyypit, epäröinnit, hiljaisuuden kesto jne.



4.3 Marked-up Data 141

Category	Examples	Claws C5	Brown	Penn
Adjective	happy, bad	AJ0	JJ	JJ
Adjective, ordinal number	sixth, 72nd, last	ORD	OD	JJ
Adjective, comparative	happier, worse	AJC	JJR	JJR
Adjective, superlative	happiest, worst	AJS	JJT	JJS
Adjective, superlative, semantically	chief, top	AJ0	JJS	JJ
Adjective, cardinal number	1, fifteen	CRD	CD	CD
Adjective, cardinal number, one	one	PNI	CD	CD
Adverb	often, particularly	AV0	RB	RB
Adverb, negative	not, n't	XX0	RB	RB
Adverb, comparative	faster	AV0	RBR	RBR
Adverb, superlative	fastest	AV0	RBT	RBS
Adverb, particle	up, off, out	AVP	RP	RP
Adverb, question	when, how, why	AVQ	WRB	WRB
Adverb, degree & question	how, however	AVQ	WQL	WRB
Adverb, degree	so, very, too	AV0	RG	RB
Adverb, degree, postposed	enough, indeed	AV0	QLP	RB
Adverb, context	here, there, now	AV0	RN	RB
Connective, text-initiator	then, thus	CJT	CS	RB
Conjunction, subordinating	although, when	CJS	CS	IN
Conjunction, complementizer that	that	CJT	CS	IN
Conjunction	and, or	CJC	CC	CC
Determiner, pronoun	no, some	DT0	DTI	DT
Determiner, pronoun, plural	these, those	DT0	DTS	DT
Determiner, prequalifier	quite	DT0	ABX	PDT
Determiner, prequantifier	all, half	DT0	ABN	PDT
Determiner, pronoun or double cons.	both	DT0	ABN	PDT CC
Determiner, pronoun or double cons.	either, neither	DT0	DTI	DT CC
Determiner, article	the, a, an	AT0	AT	DT
Determiner, possessive determiner	my, your, its	DPS	PP$	PRP$
Determiner, possessive	their, your	DPS	PP$	PRP$
Determiner, possessive, second	mine, yours	DPS	PP$	PRP
Determiner, genitive	Peter's, Smithers'	DPS	NP$	NNP
Determiner, possessive & question	whose	DTQ	WP$	WP$
Noun	aircraft, data	NN0	NN	NN
Noun, singular	woman, book	NN1	NN	NN
Noun, plural	women, books	NN2	NNS	NNS
Noun, proper, singular	London, Michael	NP0	NP	NNP
Noun, proper, plural	Australians, Methodists	NP0	NPS	NNPS
Noun, adverbial	tomorrow, home	NN0	NR	NN
Noun, adverbial, plural	Sundays, weekends	NN2	NRS	NNS
Pronoun, nominal (indefinite)	none, everything, one	PNI	PN	NN
Pronoun, personal, subject	you, we	PNP	PPSS	PRP
Pronoun, personal, subject, 3SG	he, she, it	PNP	PPS	PRP
Pronoun, personal, object	you, him, me	PNP	PPO	PRP
Pronoun, reflexive	herself, myself	PNX	PPL	PRP
Pronoun, reciprocal	themselves, ourselves	PNX	PPLS	PRP
Pronoun, question, subject	who, whoever	PNQ	WPS	WP
Pronoun, question, object	who, whoever	PNQ	WPO	WP
Pronoun, existential there	there	EX	EX	EX

Table 4.5 Comparison of different tag sets: adjective, adverb, conjunction, determiner, noun, and pronoun tags.

6 Kollokaatiot

6.1 Mikä on kollokaatio

- Kahdesta tai useammasta sanasta koostuva konventionaalistunut ilmaus
- Collocations of a given word are statements of the habitual or customary places of that word (Firth, 1957)
- Esimerkkejä:
  - 'weapons of mass destruction', 'disk drive', 'part of speech' (suomessa yhdyssanoina 'joukkotuhoaseet', 'levyasema', 'sanaluokkatieto')
  - 'bacon and eggs'
  - verbin valinta: 'prendre une dcision', mutta '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', 'Persianlahden sota' (viittaa tiettynä ajankohtana käytyyn sotaan)
- Kollokaation kanssa osittain päällekkäisiä käsitteitä: termi, tekninen termi, terminologinen fraasi. Huom: tiedonhaussa sanalla 'termi' laajempi merkitys: 'sana tai kollokaatio'.

6.2 Sanan frekvenssi ja sanaluokkasuodatus

6.2.1 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', ...

6.2.2 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.

6.3 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*'

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

Pohdittavaksi:

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

6.4 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:
$P(ww) = P(w)P(w)$

6.4.1 T-testi

Tilastollinen testi sille eroaako havaintojoukon odotusarvo oletetun, datan generoineen jakauman odotusarvosta. Olettaa, että todennäköisyydet ovat suunnilleen normaalijakautuneita. $t = \frac{\bar{x}-\mu}{\sqrt{\frac{s}{N}}}$, jossa $\bar{x}, s$: 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.

6.4.2 Soveltaminen kollokaatioihin:

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

$\mu = P(new)P(companies)$
$\bar{x} = \frac{c(new\ companies)}{c(\cdot,\cdot)} = \hat{p}$
$s = p(1-p) = \hat{p}(1-\hat{p}) \approx \hat{p}$ (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	43	20	20	Ayatollah	Ruhollah
4.4721	43	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
2.3724	14907	9017	20	first	made
2.2446	15684	10379	20	over	many
1.3685	14734	13478	20	into	them
1.2176	14093	14776	20	like	people
0.8038	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	$C(w)$	$C(strong\ w)$	$C(powerful\ w)$	Word
3.1622	933	0	10	computers
2.8284	1337	0	8	computer
2.4494	289	0	6	symbol
2.4494	588	0	6	machines
2.2360	2266	0	5	Germany
2.2360	1745	0	5	nation
2.2360	2845	0	5	chip
2.1828	1418	4	13	force
2.0000	5410	0	4	friends
2.0000	927	0	4	neighbor
7.0710	3085	10	0	support
6.2517	3010	18	7	enough
4.6904	986	12	0	safety
4.5825	3741	21	0	sales
4.0249	1091	19	1	opposition
3.9000	892	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).

6.5 Pearsonin khii-toiseen-testi χ

- χ-testi mittaa muuttujien välistä riippuvuutta perustuen riippumattomuuden määritelmään: jos muuttujat ovat riippumattomia, yhteisjakauman arvo tietyssä jakauman pisteessä on marginaalijakaumien tulo.
- Kahden muuttujan jakauma voidaan kuvata 2-ulotteisena kontingenssitaulukkona ($r \times c$).
- Lasketaan *kussakin taulukon pisteessä* (i, j) erotus havaitun jakauman O (tod. yhteisjakauma) ja odotetun jakauman E (marginaalijakaumien tulo) välillä, ja summataan skaalattuna jakauman odotusarvolla: $\chi = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})}{E_{i,j}}$ jossa siis $E_(i, j) = O(i, \cdot) * O(\cdot, j)$.
- χ on *asymptoottisesti* χ-jakautunut. Ongelma kuitenkin: herkkä harvalle datalle.
- Nyrkkisääntö: älä käytä testiä jos $N < 20$ tai jos $20 \leq N \leq 40$ ja jokin $E_{i,j} \leq 5$

6.5.1 Soveltaminen kollokaatioiden tunnistamiseen

Formuloidaan ongelma siten että kumpaakin sanaa vastaa yksi satunnaismuuttuja joka voi saada kaksi arvoa (sana joko esiintyy tai ei esiinny yksittäisessä sanaparissa).

Sanojen yhteistnjakauma voidaan tällöin esittää 2×2 taulukkoina. Esim.

	$w_1 = new$	$w_1 \neq new$
$w_2 = companies$	8	4667
$w_2 \neq companies$	15280	14287173

2×2-taulukon tapauksessa kaava voidaan johtaa muotoon:
$\chi = \frac{N(O_{11}O_{22} - O_{12}O_{21})}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}$

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

Ongelmallisuus kollokaatioiden tunnistamisen kannalta

Tässä soveltamistavassa ei erotella negatiivista ja positiivista riippuvuutta. Ts. jos sanat vierastavat toisiaan, testi antaa myös suuren arvon, koska tällöin sanojen esiintymisten välillä todellakin on riippuvuus. Kollokaatioita etsittäessä ollaan kuitenkin kiinnostuttu vain positiivisista riippuvuuksista.

Johtopäätös: Ainakaan näin soveltaminen ei välttämättä kannata.

Muita (parempia?) sovelluksia χ-testille:

- Konekäännös: Linjattujen korpusten sana-käännösparien tunnistaminen (cow, vache yhteensattumat johtuvat riippuvuudesta)
- Metriikka kahden korpuksen väliselle samankaltaisuudelle: $n \times 2$-taulukko jossa kullekin tutkittavalle sanalle $w_i, i \in (1 \dots n)$ kerrotaan ko. sanan lukumäärä korpuksessa j

6.6 Uskottavuuksien suhde

Kuinka paljon uskottavampi H_2 on kuin H_1? Lasketaan hypoteesien uskottavuuksien suhde λ: $\log \lambda = \log \frac{L(H_1)}{L(H_2)}$ Esimerkki:
H_1: w_1 ja w_2 riippumattomia: $P(w_2|w_1) = p = P(w_2|\ \not w_1)$
H_2: w_1 ja w_2 eivät riippumattomia: $P(w_2|w_1) = p1 \neq p2 = P(w_2|\ \not w_1)$

Oletetaan selvä positiivinen riippuvuus, eli $p_1 p_2$.

Käytetään ML-estimaatteja (keskiarvoja) laskettaessa p, p_1 ja p_2:
$p = \frac{c_2}{N}$, $p_1 = \frac{c_{12}}{c_1}$, $p_2 = \frac{c_2 - c_{12}}{N - c_1}$
Oletetaan binomijakaumat. Esim. $p(w_2|w_1) = b(c_{12}; c_1, p)$. Ilmaistaan kunkin mallin yhtäaikaa voimassa olevat rajoitteet tulona. Lopputulos: kirjan kaava 5.10.

$\log \lambda$ on asymptoottisesti χ-jakautunut. On lisäksi osoitettu että harvalla datalla uskottavuuksien suhteella saadaan parempi approksimaatio χ-jakaumalle kuin χ-testillä.

6.7 Suhteellisten frekvenssien suhde

Etsitään kollokaatioita jotka ovat *ominaisia tietylle keskustelunaiheelle* (subject). Verrataan frekvenssejä korpuksissa A ja B joista toinen on yleisaiheinen,

toinen erityisaiheinen: $r = \frac{c_1^A/N_A}{c_1^B/N_B}$ jossa c_1^A on sanan 1:n lukumäärä korpuksessa A jne.

$-2\log\lambda$	$C(w^1)$	$C(w^2)$	$C(w^1w^2)$	w^1	w^2
1291.42	12493	932	150	most	powerful
99.31	379	932	10	politically	powerful
82.98	932	934	10	powerful	computers
80.39	932	3624	13	powerful	force
57.27	932	291	6	powerful	symbol
51.66	932	40	4	powerful	lobbyist
51.52	171	932	5	economically	powerful
51.05	932	43	4	powerful	magnet
50.83	4458	932	10	less	powerful
50.75	6252	932	11	very	powerful
49.36	932	2064	8	powerful	position
48.78	932	591	6	powerful	machines
47.42	932	2319	8	powerful	computer
43.23	932	16	3	powerful	magnets
43.10	932	396	5	powerful	chip
40.45	932	3694	8	powerful	nerv
36.36	932	47	3	powerful	486
36.15	932	298	4	powerful	neighbour
35.24	932	5243	8	powerful	political
34.15	932	3	2	powerful	gadgets

Table 5.12 Bigrams of powerful with the highest scores according to Dunning's likelihood ratio test.

6.8 Pisteittäinen yhteisinformaatio

Muistellaan entropian $H(x)$ ja yhteisinformaation $I(x;y)$ kaavoja:

$$\begin{aligned} H(x) &= -E(\log p(x)) \\ I(X;Y) &= H(Y) - H(Y|X) = (H(X) + H(Y)) - H(X,Y) \\ &= E_{X,Y}(\log \tfrac{p(X,Y)}{p(X)p(Y)}) \end{aligned}$$

joka kuvastaa *keskimääräistä* informaatiota jonka sekä x että y sisältävät.

Määritellään *pisteittäinen yhteisinformaatio* joidenkin tiettyjen tapahtumien x ja y välillä (Fano, 1961):
$I(x;y) = \log \frac{p(x,y)}{p(x)p(y)}$

Voidaanko käyttää kollokaatioiden valintaan? Motivaationa intuitio: jos sanojen välillä on suuri yhteisinformaatio (ts. niiden kummankin kommunikoiman informaation yhteinen osuus on suuri), voisi olettaa että kyse on kollokaatiosta.

$I(w^1,w^2)$	$C(w^1)$	$C(w^2)$	$C(w^1w^2)$	w^1	w^2
18.38	42	20	20	Ayatollah	Ruhollah
17.98	41	27	20	Bette	Midler
16.31	30	117	20	Agatha	Christie
15.94	77	59	20	videocassette	recorder
15.19	24	320	20	unsalted	butter
1.09	14907	9017	20	first	made
1.01	13484	10570	20	over	many
0.53	14734	13478	20	into	them
0.46	14093	14776	20	like	people
0.29	15019	15629	20	time	last

Table 5.14 Finding collocations: Ten bigrams that occur with frequency 20, ranked according to mutual information.

Taulukosta 5.16 huomataan että jos jompikumpi sanoista on harvinainen, saadaan korkeita lukuja.

Täydellisen riippuville sanoille yhteisinformaatio: $I(x;y) = \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x)}{p(x)p(y)} = \log \frac{1}{P(y)}$

kasvaa kun sanat muuttuvat harvinaisemmiksi. Ääritilanne: kaksi sanaa esiintyy kumpikin vain kerran, ja tällöin yhdessä. Kuitenkin tällöin evidenssiä kollokaationa toimimisesta on vähän, mikä jää huomiotta.

I_{1000}	w^1	w^2	w^1w^2	Bigram	I_{10000}	w^1	w^2	w^1w^2	Bigram
16.95	5	1	1	Schwartz eschews	14.40	108	8	1	Schwartz eschews
15.82	1	19	1	fewest visits	13.00	78	22	1	FIND GARDEN
13.78	5	9	1	FIND GARDEN	11.25	22	267	1	fewest visits
12.00	5	31	1	Indonesian pieces	8.97	43	663	1	Indonesian pieces
9.82	20	27	1	flesh survived	8.04	170	1917	6	marijuana growing
8.21	13	82	1	marijuana growing	5.75	15828	51	5	now converts
7.37	24	159	1	doubt whether	5.20	680	3848	7	doubt whether
6.68	687	9	1	now converts	4.76	710	713	1	flesh survived
6.00	661	15	1	like offensive	1.95	3549	6276	9	must think
3.81	159	185	1	must think	0.44	14093	782	1	like offensive

Table 5.16 Problems for Mutual information from data sparseness. The table shows ten bigrams that occurred once in the first 1000 documents in the reference corpus ranked according to mutual information score in the first 1000 documents (left half of the table) and ranked according to mutual information score in the entire corpus (right half of the table). These examples illustrate that a large proportion of bigrams are not well characterized by corpus data (even for large corpora) and that mutual information is particularly sensitive to estimates that are inaccurate due to sparseness.

7 Tiedonhaku

Tiedonhaussa (information retrieval, text retrieval) tehtävänä on hakea käyttäjän tiedontarvetta vastaavaa tietoa suurista dokumenttikokoelmista.

7.1 Tiedonhaun perusteita

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.

Kaksi pääsuuntaa: *exact match* ja *ranking*.

7.1.1 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.

7.1.2 Ranking – Järjestetyt osumat

Täsmällisen osumajoukon palauttamisen sijaan järjestetään kaikki dokumentit paremmuusjärjestykseen sen mukaan, miten hyvin ne vastaavat hakulauseketta.

Lähestymistapoja esim. probabilistinen haku ja johonkin samankaltaisuusmittaan perustuva haku.

Nykyään täsmähakua yleisempi hakujärjestelmätyyppi.

7.1.3 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.

Lisätietoa tiedonhausta: Modern Information retrieval (Baeza-Yates & Ribeiro-Neto, 1999)

7.2 Tiedonhakujärjestelmien perusosia

7.2.1 Käänteisindeksi (inverted index)

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

7.2.2 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 sulkulista-sanoja. Vrt. myös 'it magazine'.

7.2.3 Stemming (juureksi palautus) tai perusmuotoistaminen

Stemming on approksimaatio morfologisele 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. suomelle.

Yhdellä perusmuodolla voi olla useita eri hakuvartaloita.

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

Tavallisia stemmereitä englannille ovat Porter ja Lovins. Suomen perusmuotoistus mm. TWOLilla (Koskenniemen 2-tasomalli morfologialle).

7.3 Hakumenetelmien evaluointimittoja

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

7.3.1 Saanti ja tarkkuus

Saanti ja tarkkuus ovat 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, yleensä tarkkuus 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 myös muita mittoja:

Un-interpolated average precision (interpoloimaton keskimääräinen tarkkuus)

Kerää useita tarkkuuslukuja yhteen mittaan. Tarkkuus mitataan *aina kun systeemi palauttaa relevantin dokumentin.* Näin saadut luvut keskiarvoistetaan. Relevantit dokumentit, joita ei palautettu, lasketaan mukaan tarkkuudella 0.

Esim. Menetelmälle 3: $1/2 + 2/3 + 3/6 + 4/7 + 5/8 = 0.5726$

(mikäli 10 ekan palautetun joukossa olivat kaikki relevantit dokumentit).

Interpolated average precision (interpoloitu keskimääräinen tarkkuus)

Eroina edelliseen:

1. Tarkkuudet lasketaan tietyillä saantitasoilla (tavallisesti 10% välein 0%:sta alkaen).

2. Mikäli tarkkuus jossain vaiheessa kohoaa, kaikkien aiempien lukujen tarkkuuksiksi otetaan tämä uusin, korkeampi luku.

7.3.2 F-mitta

Toinen tapa mitata tarkkuutta ja saantia yhtäaikaa, yhdellä mitalla:

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

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.

7.3.3 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: 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.

7.3.4 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.

7.4 Vektoriavaruusmalli

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

- Yleisesti käytetty, ad-hoc-retrievalin 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.

7.4.1 Termien painotusmenetelmä: tf.idf

tf: term frequency
idf: inverse document frequency

IDF yhdistää termin lokaalin merkittävyyden, eli esiintymistiheyden tässä 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:

$w(i,j) = (1 + \log(tf_{t,d})) \log \frac{N}{df_t}$ 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) $0.5 + \frac{0.5tft,d}{max_t(tf_{t,d})}$		

Muita harvinaisempia painotusmenetelmiä esitellään kirjan luvussa 15.3 (ei käsitellä tarkemmin kurssilla):

RIDF, K-mikstuurit, kahden Poisson-jakauman menetelmä.

7.5 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.

7.5.1 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.

Kritiikkiä: SVD optimoi representaation neliöllisen virheen mielessä (least-squares, L_2-normi). Tämä implikoi oletuksen että yhteisesiintymät ovat normaalijakautuneita latenteilla dimensioilla, mikä ei välttämättä pidä paikkaansa kielidatalla.

7.5.2 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:aa, mukaanlukien sen soveltaminen keskustelujen analyysiin, tutkitaan mm. TKK:lla Neuroverkkojen tutkimusyksikössä.

7.6 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:

- stoplistan 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, oten 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, early WEBSOM [Honkela, Kaski, Lagus, Kohonen, 1996])
- ICA (dimension pienennys on sivuvaikutus, mutta siis ainakin periaatteessa sovellettavissa)
- Satunnaisprojektio (on käytetty dokumenttien dimension pienennykseen)

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

7.6.1 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:

$$\mathbf{x}_i = \mathbf{R}\mathbf{n}_i \ . \tag{8}$$

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.

8 N-grammi-kielimallit

8.1 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.

8.1.1 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)

8.2 N-grammimallit

N-grammimalli: ennustetaan sanaa w_n edellisten $n-1$ sanan perusteella: $P(w_n|w_1 w_2 \cdots w_{n-1})$ 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}

8.3 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.

8.3.1 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.

8.3.2 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:
  $P(w_n|w_1, w_2, \ldots, w_{n-1})$

8.4 N-grammimallin tilastollinen estimointi

Annettuna: joukko näytteitä jotka osuvat kuhunkin ekvivalenssiluokkaan (biniin). Bayesin kaavoista: $P(w_n|w_1 \cdots w_{n-1}) = \frac{P(w_1 \cdots w_n)}{P(w_1 \cdots w_{n-1})}$ 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)

8.4.1 Maximum likelihood-estimaatti (MLE)

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

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

- 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*		*sisters*	
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$	
1	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034
2	to	0.032	to	0.032	to	0.032	**to**	**0.032**	to	0.032	to	0.032
3	and	0.030	and	0.030	and	0.030			and	0.030	and	0.030
4	of	0.029	of	0.029	of	0.029			of	0.029	of	0.029
...												
8	was	0.015	**was**	**0.015**	was	0.015			was	0.015	was	0.015
...												
13	**she**	**0.011**			she	0.011			she	0.011	she	0.011
...												
254					both	0.0005			**both**	**0.0005**	both	0.0005
...												
435					sisters	0.0003					**sisters**	**0.0003**
...												
1701					**inferior**	**0.00005**						
2-gram	$P(\cdot \mid person)$		$P(\cdot \mid she)$		$P(\cdot \mid was)$		$P(\cdot \mid inferior)$		$P(\cdot \mid to)$		$P(\cdot \mid both)$	
1	and	0.099	had	0.141	not	0.065	**to**	**0.212**	be	0.111	of	0.066
2	who	0.099	**was**	**0.122**	a	0.052			the	0.057	to	0.041
3	to	0.076			the	0.033			her	0.048	in	0.038
4	in	0.045			to	0.031			have	0.027	and	0.025
...												
23	**she**	**0.009**							Mrs	0.006	she	0.009
...												
41									what	0.004	**sisters**	**0.006**
...												
293									**both**	**0.0004**		
...												
∞					**inferior**	**0**						
3-gram	$P(\cdot \mid in,person)$		$P(\cdot \mid person,she)$		$P(\cdot \mid she,was)$		$P(\cdot \mid was,inf.)$		$P(\cdot \mid inferior,to)$		$P(\cdot \mid to,both)$	
1	UNSEEN		did	0.5	not	0.057	UNSEEN		the	0.286	to	0.222
2			**was**	**0.5**	very	0.038			Maria	0.143	Chapter	0.111
3					in	0.030			cherries	0.143	Hour	0.111
4					to	0.026			her	0.143	Twice	0.111
...												
∞					**inferior**	**0**			**both**	**0**	**sisters**	**0**
4-gram	$P(\cdot \mid u,l,p)$		$P(\cdot \mid l,p,s)$		$P(\cdot \mid p,s,w)$		$P(\cdot \mid s,w,i)$		$P(\cdot \mid w,i,t)$		$P(\cdot \mid i,t,b)$	
1	UNSEEN		UNSEEN		in	1.0	UNSEEN		UNSEEN		UNSEEN	
...												
∞					**inferior**	**0**						

8.4.2 Laplacen laki eli 'yhden lisäys'

Annetaan hiukan tn-massaa näkemättömille tapauksille lisäämällä jokaiseen lukuun 1: $P_{LAP}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n)+1}{N+B}$

- 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.

8.4.3 Lidstonen laki, Jeffreys-Perksin laki

$P_{Lid}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n)+\lambda}{N+B\lambda}$ Voidaan osoittaa että ylläoleva tarkoittaa lineaarista interpolointia tasajakauma-priorin ja MLE-estimaatin välillä. Asetetaan $\mu = N/(N + B\lambda)$: $P_{Lid}(w_1 \cdots w_n) = \mu \frac{C(w_1 \cdots w_n)}{N} + (1 - \mu)\frac{1}{B}$

- 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

8.4.4 Good-Turing -estimaattori

Ks. frekvenssien frekvenssi-histogrammeja taulukossa 6.7.

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

8.4.5 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.

8.5 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.

8.5.1 Lineaarinen interpolointi

(yleisemmin nimellä äärelliset mikstuurimallit tai sum of experts)

Lasketaan painotettu keskiarvo eri pituisten kontekstien antamista estimaateista: $P_{li}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n|w_{n-1}) + \lambda_3 P_3(w_n|w_{n-2}w_{n-1})$ ($0 \leq \lambda_1 \leq 1$ ja $\sum_i \lambda_1 = 1$)

Parametrit λ voidaan asettaa käsin tai optimoida datan avulla.

8.5.2 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: $P_{li}(w|h) = \sum_i \lambda_i(h) P_i(w|h)$ ($0 \leq \lambda_1 \leq 1$ 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.

8.5.3 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:

	$P(\text{she}\mid h)$	$P(\text{was}\mid h)$	$P(\text{inferior}\mid h)$	$P(\text{to}\mid h)$	$P(\text{both}\mid h)$	$P(\text{sisters}\mid h)$	Product
Unigram	0.011	0.015	0.00005	0.032	0.0005	0.0003	3.96×10^{-17}
Bigram	0.00529	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.

8.6 Mallien estimoinnista yleisesti

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

8.6.1 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.

8.6.2 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

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.

8.6.3 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.

8.7 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.

9 Sanaluokkien taggaus

Esimerkki:

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

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

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

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

9.1 Syntaktinen taggaus

Syntaktisen taggauksen sovelluskohteita:

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

9.2 Taggauksessa käytettävä informaatio

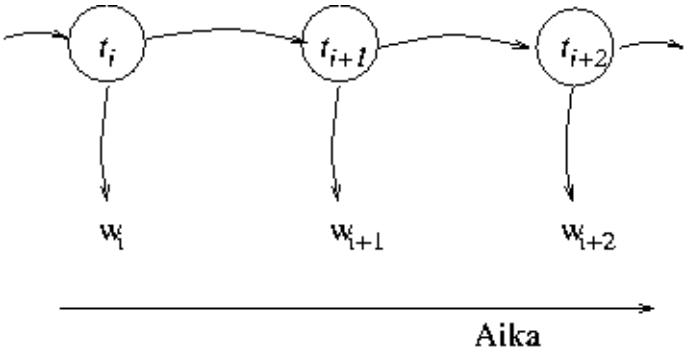
- Rakenneinformaatio, eli lähiympäristön syntaktiset tagit: tietyt tagisekvenssit tavallisempia kuin toiset
- Sanakohtainen informaatio: Sanan prioritn kuulua tiettyyn syntaktiseen luokkaan: eri luokkien tn-jakauma yleensä hyvin epätasainen kullekin yksittäiselle sanalle (vastaavasti kuin semanttisessa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

9.3 Huomioita koskien englannin kieltä

- Taggaus helpompi ongelma kuin jäsentäminen, tarkkuudet korkeita
- Parhaat taggerit luokkaa 96%-97% (oikein tagattujen sanojen osuus - tarkoittaa, että lauseissa keskimäärin 1-2 taggausvirhettä, jos lausepituus keskim. 20).

- Pelkkää rakenteista informaatiota käyttävälle sääntöpohjaiselle taggerille raportoitu (vain) 77% tarkkuus.
- Yksinkertainen menetelmä joka ei käytä tietoa rakenteista lainkaan: Luokitellaan sana aina yleisimpään POS (part of speech) -luokkaansa. Englannille raportoitu 90% tarkkuus → käytetään usein baseline-menetelmänä.

9.4 Markov-malli-taggerit



- Tagijono mallinnetaan Markov-ketjuna, eli
  $P(X_{i+1} = t^j | X_1, \ldots, X_i) = P(X_{i+1} = t^j | X_i)$ jossa i on ajanhetki ja t^j on tila jonka indeksi j. X:n arvojoukko on tilojen joukko, $\{t^1 \ldots t^n\}$.
- Sanat ovat havaintoja (observations), todennäköisyysmalli $P(w_i|X_i)$ eli sanan generointitn riippuu vain kulloisestakin tilasta.
- Bigrammitaggerissa jokainen tagi vastaa yhtä tilaa. Tällöin tämänhetkisen sanan tagi riippuu ainoastaan edeltävän sanan tagista.
- Malli opetetaan tagatulla datalla näkyvänä Markov-mallina eli tilamuuttujan arvo tunnetaan kullakin hetkellä (kullakin sanalla).
- Tagattaessa uutta dataa mallia käytetään HMM-mallina: tiloja ei tunneta vaan todennäköisin tilajono annetulla sanajonolla lasketaan mallista Viterbi-algoritmilla. Nyt Viterbin käyttö ei ongelmallista, koska ollaan kiinnostuttu nimenomaan todennäköisimmästä tilajonosta, ei sanajonosta.
- Markov-mallin rajallinen horisontti-riippumattomuusoletus ei aivan päde, edes englannille. Vielä vähemmän kielille, joissa sanajärjestyksen määräytymisessä syntaktiset tekijät eivät ole yhtä keskeisiä.
- Sanojen generointitodennäköisyydet tageista: $P(w_{1\ldots n}|t_{1\ldots n}) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$
  Estimoinnissa tehtäviä muita riippumattomuusoletuksia:
  - Sanat riippumattomia toisistaan.

– Sanan tn. riippuu ainoastaan sen tagista (eli tagi generoi sanan).

- Optimaalisen tagijonon estimointi lauseelle, sovelletaan Bayesin sääntöä:

$$\hat{t}_{1\ldots n} = arg\max_{t_{1\ldots n}} P(t_{1\ldots n}|w_{1\ldots n}) \quad (9)$$

$$= arg\max_{t_{1\ldots n}} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \quad (10)$$

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

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

- Todennäköisimmän tagijonon haku tehokkaasti Viterbi-algoritmilla.

9.5 Tuntemattomien sanojen käsittely

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

Strategioita

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

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

Lisää strategioita

- Eräs malli (Weichschedel jne): katsotaan todennäköisyyttä jolla tagi generoi tuntemattomia sanoja, sekä sanan eri piirteiden generointitn:iä tästä tagista: $P(w^l|t^j) = \frac{1}{Z}P(tuntematon|t^j)P(isoalkuk|t^j)P(lopuke_i|t^j)$ jossa Z on normalisointitekijä ja $lopuke_i$ sanan w^l lopuke (esim. '-ed'). Joissain kokeissa mallin todettiin pudottavan tuntemattomien sanojen virhetn:iä 40%:sta 20%:een (tosin ei kerrota mikä oli vertailumenetelmä, tai datan koko).

- Useimmat mallit olettavat piirteet riippumattomiksi (ns. Naive Bayes-malli, kuten yllämainittu), mikä yleensä ei pidä paikkaansa. Esim. yllä

isoalkukirjaimiset sanat ovat melko todennäköisesti myös tuntemattomia, koska ovat todennäköisesti erisnimiä, joten ko. piirteet eivät toisistaan riippumattomia.

9.5.1 Variantteja

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

9.6 HMM-taggerit

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

- Tavallisemmin käytettävissä on tunnettu tagijoukko, sekä sanakirja jossa kerrotaan mitkä tagit mahdollisia tai mahdottomia millekin sanalle (esim. JJ ei mahdollinen sanaluokka sanalle 'book'). Eri tyyppisen sanakirjatiedon vaihtoehdot on kuvattu taulukossa alempana.
- Vaihtoehtoisesti ryhmitellään sanat, joille sallittu samat tagit, ekvivalenssiluokiksi, joille yhteiset parametrit (ainakin mallin initialisointivaiheessa). Voidaan soveltaa myös pelkästään harvinaisille sanoille, koska yleisten osalta dataa on riittävästi.

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

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

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

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

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

9.7 Muunnoksiin perustuva taggaus

- Edellä kuvatut mallirakenteet, eli mallien tekemät riippumattomuusoletukset, eivät erityisen hyvin soveltuneet luonnollisen kielen kuvaamiseen. Tarvitaan siis parempia malleja.
- Kontekstia (n-grammin n) voitaisiin pidentää. Tai tagin tn voisi riippua myös edeltävistä sanoista (ei pelkästään tageista). Ongelma: parametrien määrä moninkertaistuu, estimointiongelmia.

Toisenlainen lähestymistapa on muunnoksiin perustuva taggaus (transformation-based tagging). Tähän tarvitaan:

- tagattua dataa,
- sanakirja, jossa kerrotaan sanalle sallitut tagit ja näiden tn:t.

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

Perusalgoritmi on seuraavanlainen:

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

9.7.1 Muunnokset

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

Esimerkkejä muunnossäännöistä ja herätteistä englannille:

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

9.7.2 Oppimisalgoritmi

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

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

Algoritmi:

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

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

Soveltaminen ohjaamattomassa oppimistilanteessa

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

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

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

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

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

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

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

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

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

9.8 Yhteys muihin menetelmiin

9.8.1 Päätöspuut (Decision Trees)

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

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

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

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

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

9.8.2 Eroja aitoon probabilistiseen mallinnukseen verrattuna

Ei käytettävissä prob. mallinnuksen kaikkea välineistöä.

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

Prioritiedon huomioiminen:

Muunnostaggaus pystyy helposti huomioimaan heräteympäristojen muodossa annetun prioritiedon.

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

Muita eroja:

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

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

9.8.3 Yhteys automaatteihin

Taggerin sääntöjen oppiminen tapahtuu kvantitatiivisesti.

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

9.9 Taggauksen evaluoinnista

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

Tulokseen vaikuttavat mm. seuraavat tekijät:

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

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

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

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

9.9.1 Taggausten sovelluksista

- Yllättävän vähän julkaisuja taggauksen sovelluksista.
- Useissa sovelluksissa tarvitaan lisäksi *osittaisjäsennys*
- Information extraction: taggausta jossa syntaktisten kategorioiden sijaan pyritään tunnistamaan (tiettyjä) semanttisia kategorioita (esim. erisnimet). Syntaktisesta kategoriasta voi olla apua.
- Tiedonhaun indeksointitermien valinta tai painotus (sekä taggaus että osittaisjäsennys)
- Kysymyksiin vastaus: 'Who killed president Kennedy' , vastaus 'Oswald' edellyttää että ymmärretään mikä asiaan liittyvä tieto (esim. aika, paikka, vai henkilö) kysyjälle pitäisi kertoa. Lisäksi on pystyttävä eristämään oikea tieto dokumenteista jotka aiheeseen liittyvät. Molemmissa voi olla hyötyä taggauksesta.
- Negatiivinen tulos taggauksen hyödyllisyyden kannalta: parhaat sanatietoa hyödyntävät prob.jäsentimet toimivat paremmin lähtemällä taggaamattomasta tekstistä ja taggaamalla sitä itse kuin hyödyntämällä valmista taggausta.