Aleksi Pekkala

Sanaluokkien automaattisen tunnistamisen menetelmät

Tietotekniikan kandidaatintutkielma

3. joulukuuta 2013

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Aleksi Pekkala

Yhteystiedot: aleksi.v.a.pekkala@student.jyu.fi

Työn nimi: Sanaluokkien automaattisen tunnistamisen menetelmät

Title in English: Methods for automated part-of-speech tagging

Työ: Kandidaatintutkielma

Sivumäärä: 26+0

Tiivistelmä: Tiivistelmä on tyypillisesti 5-10 riviä pitkä esitys työn pääkohdista (tausta, tavoite, tulokset, johtopäätökset).

Avainsanat: kieliteknologia, luonnollisten kielten käsittely, sanaluokkien tunnistaminen, koneoppiminen

Abstract: Englanninkielinen versio tiivistelmästä.

Keywords: computational linguistics, natural language processing, part-of-speech tagging, machine learning

Sisältö

1 Johdanto

Sanaluokkien automaattinen tunnistaminen (engl. *part-of-speech tagging*) tarkoittaa sanojen sanaluokkien tunnistamista tekstiyhteyden perusteella. Tunnistamisprosessissa tarkastellaan tekstiaineistoa, kuten

a black cat jumped on the table

jonka perusteella pyritään päättelemään se sanaluokkien sarja, joka todennäköisimmin vastaa kyseistä aineistoa; tässä tapauksessa tuloksena voisi olla esimerkiksi

Det Adj Noun Verb Prep Det Noun

Sanaluokkien tunnistaminen on laajuudeltaan rajallinen ongelma: sen tarkoituksena ei ole jäsentää kokonaisia lauserakenteita tai tulkita lauseiden merkitystä — tarkastelun alla ovat vain yksittäisten sanojen ~~syntaktiset~~ leksikaaliset kategoriat. Sanaluokkien tunnistaminen on kuitenkin välttämätön ensimmäinen askel useimmissa luonnollisten kielten käsittelyprosesseissa, ja siten yksi aihealueen keskeisimmistä osaongelmista.

Rajallisen laajuutensa myötä sanaluokkien tunnistaminen on paljon helpommin lähestyttävä ongelma kuin kielen täydellinen ymmärtäminen, ja sen ratkaisemiseksi onkin kehitetty useita kohtuullisen luotettavia menetelmiä. Täysin ratkaistusta ongelmasta ei kuitenkaan voida puhua, sillä yksikään tunnettu menetelmä ei vielä saavuta täydellistä tunnistustarkkuutta.

Sanaluokkien tunnistajia käytetään monissa erilaisissa luonnollisiin kieliin liittyvissä sovelluksissa, ja tunnistajalle asetetut vaatimukset vaihtelevat sovelluksittain. Myös tunnistettavien aineistojen välillä on valtavasti poikkeamia, esim. kielten sekä tekstilajien osalta. Lisäksi havaitaan, että nykyisten tunnistusmenetelmien saavuttamat tunnistustarkkuudet liikkuvat kaikki suunnilleen samoissa lukemissa. Kun il-

miselvin valintakriteeri on näin poissuljettu, on tehokkaimman menetelmän valinta vaikeampaa. Tunnistusmenetelmien toimintaperiaatteiden vaihdellessa merkittävästi on kuitenkin väistämätöntä, että jotkin menetelmät soveltuvat toisia paremmin tiettyihin tunnistustehtäviin. Tässä tutkielmassa pyritäänkin selventämään sitä, ~~kuinka eri tunnistusmenetelmät käyttäytyvät suhteessa toisiinsa erilaisissa toimintaympäristöis~~ millaisia eri ratkaisuja sanaluokkien tunnistusongelmaan on olemassa ja mitkä ovat niiden ~~oleellisimmat vahvuudet sekä~~tä~~ä~~~~heikkoudet~~rkeimmät erot. Menetelmien suhteelliset ominaisuudet johdetaan tarkastelemalla lähemmin kunkin menetelmän toimintaa ~~,~~ sekä menetelmään liittyvää tutkimuskirjallisuutta.

Tutkielma rakentuu seuraavasti: toisessa luvussa annetaan lyhyt johdanto sanaluokkien tunnistamiseen ja sen haasteisiin. Luvuissa 3-5 tarkastellaan kolmea erilaista tunnistusmenetelmää~~,~~: transformaatiosääntöjä, Markovin piilomalleja sekä log-lineaarisia malleja. Luvuissa esitellään ~~niiden~~ menetelmien keskeiset ominaisuudet, ~~ja~~ sekä pyritään hahmottamaan ~~kunkin menetelmä~~niiden suhteelliset vahvuudet ja heikkoudet. Lopuksi vielää ~~n suhteelliset vahvuudet . TODO mainitaan~~ kootaan yhteen menetelmistä kerätyt huomiot ja esitellään johtopäätökset.

TODO olisiko syytä esitellä lyhyesti valitut menetelmät ~~ja valintaperusteet (~~tässä, ja/tai mainita jotain valintaperusteista tai esitysjärjestyksestä ~~:~~ (kaksi ensimmäistä ovat tavallaan toistensa vastakohtia, ja kolmas menetelmä yhdistää piirteitä kummastakin aikaisemmasta menetelmästä. Samalla menetelmät ovat järjestetty yksinkertaisimmasta monimutkaisimpaan.).

2 Sanaluokkien automaattinen tunnistaminen

2.1 ~~Mihin sanaluokkien tunnistamista käytetään?~~

Sanaluokkien tunnistaminen on tärkeä ja käytännöllinen ongelma, joka kohdataan ~~useimmissa~~ lähes kaikissa luonnollisten kielten käsittelyyn liittyvissä tehtävissä. Tällaisia tehtäviä ovat mm. puheentunnistus, konekääntäminen sekä semanttinen haku ja analyysi. Kyseisissä tehtävissä sanaluokkien tunnistaja toimii jonkin laajemman prosessointiketjun alkupäässä, koko prosessille välttämättömänä esikäsittelijänä, joka mahdollistaa syötteen jatkokäsittelyn korkeammalla tasolla. Jatkokäsittelyn tyypillisin seuraava vaihe on tekstin jäsentäminen eli lauserakenteiden tunnistaminen. Oikeiden lauserakenteiden tunnistamisen kannalta on oleellista, että lauseiden sanaluokat on tunnistettu mahdollisimman virheettömästi: yksikin virheellinen sanaluokka voi tehdä oikean lauserakenteen tunnistamisesta mahdotonta, ja siten vääristää lauseen tulkittua merkitystä.

2.1 ~~Sanaluokkien tunnistamisen lyhyt historia~~

~~TODO~~

2.1 Miksi sanaluokkien tunnistaminen on ongelmallista?

Sanaluokkien tunnistaminen voi intuitiivisesti tuntua helpolta, mutta tehtävän automaatiota hankaloittavat kaksi oleellista ongelmaa: sanojen monitulkintaisuus sekä tuntemattomat sanat. Esimerkiksi ~~lauseet~~ lauseissa

Time flies like an arrow.

Fruit flies like a banana voidaan tulkita lukuisin eri tavoin, joista mik. sana flies esiintyy ensin verbin jasitten

voidaan luokitella useampaan kuin yhteen sanaluokkaan (DeRose, 1988) .

Jatkokäsittelyn kannalta automaattisen tunnistajan oleellisin tehtävä onkin ~~sen sopivimman~~

~~merkityksen valinta, eli ns. morfologinen~~ valita kaikista mahdollisista sanaluokista se, joka tuottaa luontevimman tulkinnan. Tällaisen yksikäsitteistä~~minen. Yksikäsitteistä~~misen mahdollistavat luonnollisten kielten sisäänrakennetut rajoitteet, ~~ja erityisesti kaksi oleellista vihjetyyppiä~~jotka voidaan jakaa lokaaleihin sekää ~~: lokaalit vihjeet~~ kontekstuaalisiin vihjeisiin: lokaaleista vihjeistä ilmeisin on itse sana (”sana *can* on on todennäköisemmin modaaliverbi kuin substantiivi”), ~~sekä~~mutta pää~~kontekstuaaliset vihjeet (”~~ätelmiä voidaan tehdä myös mm. sanan prefiksin, suffiksin tai kirjainten koon perusteella. Kontekstuaalisia vihjeitä ovat kaikki lauseen muut sanat sanaluokkineen: esimerkiksi sana *fly* on todennäköisimmin substantiivi, jos edeltävä sana on artikkeli~~”)~~.

~~Toinen~~ On tärkeää huomata, ettei sanaluokkien tunnistaminen itsessään ole ratkaisu kieliopilliseen monitulkintaisuuteen: monitulkintaisuudella on useita tasoja, joista osaa käsitellään vielä prosessointiketjun myöhemmissä vaiheissa. Esimerkiksi syntaktinen, tai rakenteellinen monitulkintaisuus on ongelma, joka on huomattavasti helpompi ratkaista lauseiden jäsennyksen yhteydessä. Sanaluokkien tunnistamista ei myöskään tule sekoittaa semanttiseen yksikäsitteistämiseen, eli sanan merkityksen selvittämiseen: esimerkiksi sana *mouse* on semanttisesti monitulkintainen, vaikka sen sanaluokka tunnettaisiinkin. Sanaluokkien tunnistamisen rooli on pikemminkin rajata mahdollisten tulkintojen määrää prosessointiketjun alkupäässä, jotta myöhemmissä vaiheissa vältytään turhalta työltä.

Monitulkintaisuuden lisäksi toinen merkittävä ongelma on tuntemattomien, eli harjoitusaineistosta puuttuvien sanojen käsitteleminen. ~~Tuntemattomia sanoja~~ Englannin kielessä yleisiä tuntemattomia sanoja ovat erisnimet sekä puhekieliset, vieraskieliset ja muut harvinaiset ilmaisut. Tällaisia sanoja kohdataan usein, ja koska niitä koskevaa tilastollista informaatiota tai sääntöjä ei tunneta, joudutaan turvautumaan jonkinlaiseen poikkeuskäsittelyyn. Useat tunnistajat hyödyntävät tuntemattomien sanojen kohdalla kieliopillisia ominaisuuksia: yksinkertainen menetelmä on määrätä sanalle se sanaluokka, joihin tuntemattomien sanojen on havaittu todennäköisimmin kuuluvan, eli yleensä substantiivi. Parempia tuloksia on saavutettu määrittämällä tuntemattoman sanan sanaluokka sen päätteen perusteella; esim. englannin kielen *able*-päätteiset sanat ovat hyvin todennäköisesti adjektiiveja (Samuels-

son, 1993). Menetelmä ei kuitenkaan sovellu kaikille kielille: esim. Tseng ym. (2005) osoittavat, että kiinan kielessä esiintyy huomattavan suuri määrä yleisiä affikseja, poiketen englannin ja saksan kielistä, joissa affiksit ovat vahva indikaattori sanan sanaluokasta.

~~Ongelmallista on myös tunnistamisessa käytettävien sanaluokkien (engl. *POS tagset*) valinta. Yleisiä englannin kielen sanaluokkasettejä ovat Brownin aineiston 87 sanaluokkaa (?) , tai uudemman Penn Treebank-aineiston 48 sanaluokkaa (Marcus ym., 1993) . Myös yleisiä, kielestä riippumattomia sanaluokkasettejä on kehitetty, joskin tällöin joudutaan väistämättä tinkimään tunnistamistarkkuudesta (?) .~~

2.2 Automaattisten tunnistajien suorituskyky

Kuten mainittua, nykyisten automaattisten sanaluokkien tunnistajien tunnistustarkkuus — englanninkielistä kirjakieltä analysoitaessa — on hieman yli 97% (Toutanova ym., 2003, Shen ym., 2007, Spoustova ym., 2009, Søgaard, 2010). Manning (2011) kuitenkin osoittaa, että kyseistä tulosta ei ole syytä tulkita liian optimistisesti: esimerkiksi lukuisat välikemerkit ja muut yksikäsitteiset elementit vääristävät evaluaatiotuloksia. Lisäksi useissa tekstilajeissa, kuten uutisiartikkeleissa, lauseiden keskipituus on yli 20 sanaa, jolloin edellämainitullakin tunnistustarkkuudella jokaisessa lauseessa on keskimäärin ainakin yksi virhe (Manning & Schütze, 1999). Artikkelissa huomautetaankin, että realistisempaa olisi tarkastella automaattisten tunnistajien kykyä tunnistaa kokonaiset lauseet oikein, sillä pienikin virhe lauseessa voi vahingoittaa tunnistajan hyödyllisyyttä myöhempien prosessointivaiheiden kannalta; tällä saralla tunnistajat saavuttavat noin 55-57% tarkkuuden, mikä on huomattavasti vaatimattomampi tulos.

Tarkkuustuloksia arvioidessa tulee myös ottaa huomioon varsin korkea lähtötaso: jo yksinkertaisimmalla metodilla, eli valitsemalla kullekin sanalle se sanaluokka, joka esiintyy harjoitusaineistossa useiten annetun sanan yhteydessä, saavutetaan 90% tunnistustarkkuus (Charniak ym., 1993).

On myös mielenkiintoista huomata, että edes ammattilaisten käsin luokittelemat ai-

neistot eivät saavuta täydellistä tunnistamistarkkuutta: ihmisten sanaluokkien tunnistamistarkkuuden on arvioitu olevan noin 97% (Manning, 2011), mikä vastaa edellämainittua automaattisten tunnistajien huipputulosta.

2.3 Sanaluokkien tunnistajan vaatimukset

Jotta tunnistajaa voidaan käyttää laajan kielenprosessointijärjestelmän komponenttina, tulee sen toteuttaa seuraavat ominaisuudet (Cutting ym., 1992):

Kestävyys Tunnistajan tulee kyetä selviytymään kaikista tekstisyötteen mahdollisista poikkeamista, kuten otsikoista, taulukoista sekä tuntemattomista sanoista.

Tehokkuus Voidakseen käsitellä laajoja tekstiaineistoja tunnistajan tulee toimia lineaarisessa ajassa.

Tarkkuus Tunnistajan tulee kyetä ehdottamaan sanaluokka jokaista annettua sanaa kohden. Myös tunnistajan mahdollisen opettamisen tulisi onnistua mahdollisimman nopeasti.

Viritettävyys Tunnistajan tulee osata hyödyntää erilaisia kielitieteellisiä huomioita siten, että tunnistajan tekemiä virheitä voidaan paikata määrittämällä sopivia vihjeitä.

Uudelleenkäytettävyys Tunnistajan tulee rakentua siten, että sen kohdistaminen uudelle kielelle, aineistolle tai sanaluokkasetille on mahdollisimman vaivatonta.

2.4 Harjoitusaineisto ja sanaluokkasetit

~~TODO~~ Sanaluokkien tunnistaminen on luonteeltaan sarjanluokitteluongelma, joka taas on yksi koneoppimisen (tarkemmin hahmontunnistuksen) aliongelmatyypeistä. Tässämääkappaleessa mainitaan yleisimmän seurauksena nykyiset sanaluokkien tunnistusmenetelmäät ~~harjoitusaineistot ja sanaluokkasetit (Brown, Penn Treebank). Mahd. syytä~~perustuvat läätähes poikkeuksetta ohjattuun oppimiseen, eli tunnistimet tulee ”opettaa” jonkinlaisella harjoitusaineistolla ennen kääsmentäyttöä. Sanaluokkien

tunnistimille syötettäävä harjoitusdata koostuu suurista tekstiaineistoista (engl. *corpus*), joissa jokaisen sanan yhteyteen on merkattu oikea sanaluokka. Nykyisin ehkä yleisimmin käytetty englanninkielinen harjoitusaineisto on ns. Penn Treebank-aineisto (Marcus ym., 1993), joka sisältää ~~, että~~noin viiden miljoonan sanan edestää ~~on kyse ohjatusta oppimisesta. Testiaineiston ja harjoitusaineiston erot; mitä~~uutisartikkeleita, kaunokirjallisuutta, tieteellisiää ~~seuraa jos liian erilaiset? Ongelmaan vaikuttaa~~ julkaisuja ym. tekstilajeja. (TODO harjoitusaineiston merkitys tunnistustuloksille).

Tunnistusongelmaan on olemassa myös ~~tunnistettavien sanaluokkien~~ ohjaamattomaan oppimiseen perustuvia ratkaisuja, jotka eivät vaadi valmiiksi luokiteltua harjoitusaineistoa. Toisaalta tällaiset menetelmät ovat väistämättä alttiimpia virheille kuin vastaavat ohjatun oppimisen menetelmät. Joissain tapauksissa — esimerkiksi harvinaisia kieliä analysoitaessa — luokiteltua harjoitusaineistoa ei kuitenkaan ole saatavilla, jolloin ohjaamaton oppiminen on ainoa vaihtoehto.

Tunnistamisessa käytetyt sanaluokat määrä~~:~~ ytyvät yleensä harjoitusaineiston mukaan: esimerkiksi Penn Treebank-aineiston käyttämä sanaluokkasetti koostuu 48 sanaluokasta, joista 12 ovat välikemerkkejä. On tärkeää huomata, että käytettävän sanaluokkasetin laajuus vaikuttaa suoraan tunnistustarkkuuteen: mitä enemmän sanaluokkia, sitä suurempi mahdollisuus monitulkintaisuuteen. ~~Harjoitusaineiston koko.~~ Toisaalta sanaluokkien tunnistamisen tuottama lingvistinen informaatio on sitä arvokkaampaa, mitä tarkempi jaottelu eri sanaluokkien välillä on. (Marcus ym., 1993)

(TODO: mahd. listaus Penn Treebank-sanaluokista liiteeksi, esimerkkipätkä harjoitusaineistosta

3 Transformaatiosäännöt

~~TODO~~

Ensimmäiset automaattiset sanaluokkatunnistimet (mm. Greene & Rubin, 1971) olivat peräisin kielitieteen piiristä, ja ne perustuivat pitkälti käsin laadittuihin kieliopillisiin transformaatiosääntöihin. Tällainen sääntöpohjainen menetelmä toimii seuraavasti: ensin kunkin sanan mahdolliset sanaluokat haetaan sanakirjasta tai vastaavasta tietolähteestä. Seuraavaksi niiden sanojen kohdalla, joiden sanaluokka ei ole yksikäsitteinen, sovelletaan valmiita transformaatiosääntöjä oikean sanaluokan tunnistamiseen. Transformaatiosäännöt voivat hyödyntää sekä lokaaleja että kontekstuaalisia kieliopillisia vihjeitä; tyypillisiä transformaatiosääntöjä ovat esimerkiksi *korvaa substantiivi erisnimellä, jos sanalla on iso alkukirjain* (lokaali vihje) tai *korvaa substantiivi verbillä, jos edeltävä sanaluokka on pronomini* (kontekstuaalinen vihje).

Tällaisen lähestymistavan ilmeisin heikkous on vaaditun manuaalisen työn määrä: erilaisille aineistoille tulee aina luoda uusi, aineiston kielelle ja tyylille spesifi sääntökokoelma. Huomattavan työpanoksen lisäksi sääntöpohjainen menetelmä vaatii myös ymmärryksen tulkittavan aineiston kieliopillisista ominaisuuksista, jotta luodut säännöt tuottavat toivotun tuloksen. Puhtaasti sääntöpohjaisilla menetelmillä voidaan saavuttaa — sääntöjen määrästä riippuen — korkeita tarkkuustuloksia, mutta kyseiset tulokset eivät ole siirrettävissä erilaisille aineistoille ilman mittavia muutoksia.

Sääntöpohjaisten menetelmien puutteiden vuoksi useimmat nykyiset sanaluokkien tunnistajat perustuvat tilastollisiin menetelmiin: sanaluokkia koskeva tilastollinen informaatio voidaan poimia harjoitusaineistosta automaattisesti, siinä missä sääntöjen laatiminen vaatii lingvististä asiantuntemusta ja manuaalista työtä. Seuraavissa kappaleissa esiteltävät Markovin piilomallit sekä log-lineaariset mallit ovat esimerkkejä tilastollisista tunnistusmenetelmistä. Sääntöpohjaisilla menetelmillä on kuitenkin joitakin etuja tilastollisiin menetelmiin verrattuna: ensinnäkin kielioppisääntöjen tallentaminen vaatii huomattavasti vähemmän tallennustilaa kuin vastaava tilastollinen informaatio. Tilastollista informaatiota on myös hankalampi tulkita ja käsitellä kuin yksinkertaisia

kielioppisääntöjä, jonka myötä tunnistusvirheiden tunnistaminen ja korjaaminen on helpompaa kielioppisääntöjä käytettäessä. (Brill, 1992)

Brill (1992) huomauttaakin, että tilastolliset tunnistajat saavuttavat korkean tunnistamistarkku kiinnittämättä varsinaisesti huomiota aineiston taustalla olevaan kieliopilliseen rakenteeseen. Hän laskee tilastollisten menetelmien puutteeksi sen, että ne poimivat aineistosta lingvistisen informaation sijaan vain suuren määrän vaikeselkoisia tilastoja.

3.1 Brillin sääntöpohjainen sanaluokkatunnistin

Brill (1992, 1994) esittää artikkeleissaan vaihtoehtoisen lähestymistavan, joka pohjautuu varhaisimpien tunnistusmenetelmien tavoin kieliopillisiin sääntöihin. Aikaisemmista sääntöpohjaisista tunnistamismenetelmistä poiketen sääntöjä ei kuitenkaan syötetä manuaalisesti, vaan tunnistin oppii ne automaattisesti oikeilla sanaluokilla merkitystä harjoitusaineistosta. Menetelmän kantava idea on tunnistaa tehdyt tunnistusvirheet, ja inkrementaalisesti soveltaa virheitä korjaavia transformaatiosääntöjä kunnes ne eivät enään paranna kokonaistarkkuutta.

3.1.1 Alustus

Brillin tunnistin alustetaan

3.1.2 Transformaatiosääntöjen soveltaminen

3.2 Arviota

TODO: yksinkertainen, tarkkuus?

4 Markovin piilomallit

~~TODO - esimerkkikuva Markovin ketjusta - markov-oletus - markovin ketjut - bigrammit/trigra - miksi piilotettu?~~

Edellisestä sääntöpohjaisesta menetelmästä poiketen useimmat nykyiset sanaluokkien tunnistajat perustuvat tilastollisiin menetelmiin. Tilastollisissa menetelmissä sanaluokkien tunnistaminen mielletään lingvistisen lähestymistavan sijaan koneoppimisongelmaksi, tai tarkemmin sarjanluokitteluongelmaksi. Sarjanluokitteluongelmassa tavoitteena on oppia funktio $f : \mathcal{X} \to \mathcal{Y}$, joka luokittelee kunkin syötteen x johonkin luokkaan y. Todennäköisyyslaskennan kautta funktio f voidaan määritellä muodossa

$$f(x) = \underset{y \in \mathcal{Y}}{\arg\max}\, P(y|x) \tag{4.1}$$

missä todennäköisyyttä $P(y|x)$ estimoidaan harjoitusaineiston perusteella.

Tilastolliset mallit voidaan vuorostaan jakaa diskriminatiivisiin ja generatiivisiin malleihin. Diskriminatiiviset mallit (esim. log-lineaariset mallit) käsittelevät suoraan todennäköisyyttä $P(y|x)$, eli ne eivät ota kantaa syötteeseen x. Generatiiviset mallit (esim. Markovin piilomallit) puolestaan käsittelevät koko yhteisjakaumaa $P(x,y)$, josta todennäköisyys $P(y|x)$ johdetaan Bayesin säännön avulla. Diskriminatiiviset mallit ovat siten generatiivisia malleja rajoittuneempia, mutta sarjanluokitteluongelman kannalta mallin rajoitteilla ei ole väliä. Intuition mukaan generatiivinen malli on diskriminatiivista mallia tehottomampi, sillä yhteisjakauman mallintaminen on laskennallisesti vaativampaa kuin ehdollisen jakauman. Ng & Jordan (2002) kuitenkin osoittavat, ettei tämä välttämättä aina pidä paikkaansa: sanaluokkien tunnistamiseen onkin olemassa kumpaankin malliin perustuvia tehokkaita ratkaisuja.

4.1 Markovin malli

Markovin malli (mm. Rabiner, 1989) kuvaa sellaista stokastista prosessia, ~~jonka vallitseva~~ joka toteuttaa ns. Markov-ominaisuuden: seuraava tila riippuu aina vain N:stä edel-tä~~vistä~~vää~~tiloista. Yksinkertaisimmillaan malli koostuu havaintoja kuvaavista tiloista, joille kullekin on~~ stä tilasta. Markov-ominaisuus on siis eräänlainen riippumattomuusoletus, joka yksinkertaistaa stokastisen prosessin tilan estimointia rajoittamalla tilasiirtymien historian määrä~~tty tilasiirtymien~~ ä. N:nen asteen Markov-ominaisuuden toimiessa esimerkiksi todennäkö~~isyydet~~isyys

$$P(x_k|x_1,\ldots,x_{k-1}) \tag{4.2}$$

voidaan laskea huomattavasti yksinkertaisemmin tarkastelemalla vain N:ää edellistä tilaa:

$$P(x_k|x_{k-N},\ldots,x_{k-1}) \tag{4.3}$$

Markov-ominaisuudesta puhuttaessa on yleensä kyse juuri ensimmäisen asteen Markov-omina jolloin $N = 1$. ~~Usein prosessin tila ei kuitenkaan ole suoraan havaittavissa, eli tila on piilotettu, joskin havainnosta riippuvainen ;~~ Käytännössä tä~~llä~~mä tarkoittaa sitä, että tarkastellaan jotakin kahta peräkkäistä tilaa — nykyistä sekä tulevaa. Markov-ominaisuutta voidaan kuitenkin laajentaa myää~~in on kyse Markovin piilomallista. Sanaluokkien tunnistamisongelma voidaan kuvata kyseisenä~~s korkeampiin asteisiin, jolloin myös tarkasteltavien tilasarjojen pituudet kasvavat. Sanaluokkia tunnistaessa nää~~piilomallina:~~ itä tilasarjoja vastaavat n:n peräkkäisen sanaluokan sarjat, eli ns. n-grammit.

TODO: Asteen valinnan merkitys.

Yksinkertaisimmillaan Markovin mallia voidaan kuvata tilakoneena, joka koostuu havaittavia tapahtumia kuvaavista tiloista, sekä tilasiirtymämatriisista, josta ilmenevät todennäköisyydet siirtyä kustakin tilasta mihin tahansa muuhun tilaan. Kukin tila on siis itsenäinen ja muistiton.

TODO Mahd. selitykset HMM:n viidestä elementistä (Rabiner, 1989) , esimerkkikuva Markovin ketjusta.

4.1.1 Markovin piilomalli

Markovin mallin hyödyllisyyttä rajoittaa se, että malli ei pysty itsessään mallintamaan prosesseja, joiden tilat eivät ole suoraan havaittavissa. Useimmissa mielenkiintoisissa tapauksissa prosessin tilat eivät kuitenkaan suoraan vastaa havaintoja, eli prosessin tila — vaikkakin havainnosta riippuvainen — on piilotettu: esimerkiksi sanaluokkien tunnistamisongelmassa havainto (sana) on tiedossa, tila (sanaluokka) on riippuvainen havainnosta, mutta tila itsessään ei ole tiedossa. Tällöin Markovin malli tulee laajentaa Markovin piilomalliksi, jossa havainto on aina sitä vastaavan tilan todennäköisyysfun

TODO: tähän kuva Markovin piilomallista

4.2 Lähtökohta

Tunnistamisongelmassa siis haetaan annetulla lauseelle sitä todennäköisimmin vastaavaa sanaluokkien sarjaa. Todennäköisyyttä voidaan mallintaa ~~funktiolla~~

yhteisjakaumalla

$$p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n)$$

$$p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) \tag{4.4}$$

mikä ilmaisee todennäköisyyden sille, että jokin lause $w_1 \ldots w_n$ esiintyy jonkin sanaluokkasarjan $t_1 \ldots t_n$ yhteydessä. Tällöin ratkaistavaksi jää

$$\arg\max_{t_1 \ldots t_n} p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n)$$

$$\arg\max_{t_1 \ldots t_n} p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) \tag{4.5}$$

eli sanaluokkasarja $t_1 \ldots t_n$, jolla saadaan maksimiarvo edeltävästä funktiosta. Mahdollisten sanaluokkasarjojen määrä kuitenkin kasvaa eksponentiaalisesti sanojen ja sanaluokkien määrän mukaan, jolloin maksimiarvon ratkaiseminen naiivisti on epätehokasta.

4.3 Tunnistusongelma Markovin piilomallina

Markovin ~~piilomallien avulla~~ piilomallit hyödyntävät sitä seikkaa, että yhteisjakauma $P(x,y)$ voidaan jakaa osiin $P(y)P(x|y)$. Tällöin edellämainittu funktio p (4.4) voidaan kuvata muodossa

$$p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) = \underbrace{\prod_{i=1}^{n+1} q(t_i|t_{i-2}, t_{i-1})}_{\text{Markovin ketju}} \prod_{i=1}^{n} e(w_i|t_i)$$

$$p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) \tag{4.6}$$

$$= p(t_1, t_2, \ldots, t_n)\, p(w_1, w_2, \ldots, w_n|t_1, t_2, \ldots, t_n) \tag{4.7}$$

$$= \prod_{i=1}^{n+1} q(t_i|t_{i-2}, t_{i-1}) \prod_{i=1}^{n} e(w_i|t_i) \tag{4.8}$$

missä t_0 ja t_{-1} ovat lauseen alkuun lisättyjä alkusanaluokkia, ja t_{n+1} on päätemerkkisanaluokka. Mallin ensimmäinen parametri

$$q(t_i|t_{i-2},t_{i-1})$$

$$q(t_i|t_{i-2},t_{i-1}) \tag{4.9}$$

laskee todennäköisyyden sanaluokalle t_i, kun kaksi edeltävää sanaluokkaa ovat t_{i-1} ja t_{i-2}. ~~Tä~~Parametri voidaan myös mieltää~~stää~~ä ~~trigrammista voidaan pä~~todennää~~ä~~köisyytenää ~~tellä~~trigrammille t_{i-2},t_{i-1},t_i. Täästä huomataan, että kyseessä on ~~ns.~~ toisen asteen Markovin piilomalli. Mallin toinen parametri

$$e(w_i|t_1)$$

$$e(w_i|t_i) \tag{4.10}$$

laskee todennäköisyyden sille, että sana w_i esiintyy sanaluokan t_i yhteydessä.

4.3.1 Parametrien estimointi

Yksinkertaisimmillaan parametri $q(t_i|t_{i-2},t_{i-1})$ voidaan estimoida laskemalla

$$q(t_3|t_1,t_2) = \frac{f(t_1,t_2,t_3)}{f(t_1,t_2)}$$

$$q(t_i|t_{i-2},t_{i-1}) = \frac{f(t_{i-2},t_{i-1},t_i)}{f(t_{i-1},t_{i-1})} \tag{4.11}$$

missä funktio f merkitsee annetun n-grammin lukumäärää harjoitusaineistossa. Brants (2000) kuitenkin osoittaa, ~~ettei~~ että datan harvuuden vuoksi tällainen estimaatti ei

ole käyttökelpoinen~~, sillä~~ : laajassakaan harjoitusaineistossa ei ole tarpeeksi montaa kappaletta kutakin eri trigrammia. Lisäksi osa trigrammeista ~~t_i, t_{i+1}, t_{i+2}~~ t_{i-2}, t_{i-1}, t_i ovat väistämättä sellaisia, että ~~$f(t_i, t_{i+1}, t_{i+2}) = 0$~~ $f(t_{i-2}, t_{i-1}, t_i) = 0$, jolloin koko sarja $t_1 \dots t_n$ saa todennäköisyyden 0. Luotettavampi tapa estimoida arvoa q on hyödyntää trigrammien lisäksi myös harjoitusaineistosta johdettujen uni- ja ~~digrammien~~ bigrammien suhteellisia frekvenssejä:

$$Unigrammi: P(t_3) = \frac{f(t_3)}{N}$$

$$Digrammi: P(t_3|t_2) = \frac{f(t_2, t_3)}{f(t_2)}$$

$$Trigrammi: P(t_3|t_1, t_2) = \frac{f(t_1, t_2, t_3)}{f(t_1, t_2)}$$

$$Unigrammi: P(t_i) = \frac{f(t_i)}{N} \tag{4.12}$$

$$Bigrammi: P(t_i|t_{i-1}) = \frac{f(t_{i-1}, t_i)}{f(t_{i-1})} \tag{4.13}$$

$$Trigrammi: P(t_i|t_{i-2}, t_{i-1}) = \frac{f(t_{i-2}, t_{i-1}, t_i)}{f(t_{i-2}, t_{i-1})} \tag{4.14}$$

missä N merkitsee harjoitusaineiston sanojen kokonaislukumäärää. Nyt funktion q arvoa voidaan silottaa interpoloimalla edellämainittuja n-grammeja:

$$q(t_3|t_1, t_2) = \lambda_1 P(t_3) + \lambda_2 P(t_3|t_2) + \lambda_3 P(t_3|t_1, t_2)$$

$$q(t_i|t_{i-2}, t_{i-1}) = \lambda_1 P(t_i) + \lambda_2 P(t_i|t_{i-1}) + \lambda_3 P(t_i|t_{i-2}, t_{i-1}) \tag{4.15}$$

missä $\lambda_1 + \lambda_2 + \lambda_3 = 1$ ja $\lambda_1, \lambda_2, \lambda_3 \geq 0$ (TODO muut silotusmenetelmät, perustelu interpolaatiolle ja lambda-arvoille). Vastaavasti todennäköisyys e voidaan estimoida

vertaamalla sana-sanaluokka-yhdistelmän frekvenssiä pelkän sanaluokan frekvenssiin:

$$e(w_3|t_3) = \frac{f(w_3,t_3)}{f(t_3)}$$

$$e(w_i|t_i) = \frac{f(w_i,t_i)}{f(t_i)} \tag{4.16}$$

4.3.2 Tuntemattomien sanojen käsittely

~~Edellinen todennä~~Todennääköisyyden e estimaatti (4.16) ei kuitenkaan ole luotettava, jos ~~sana w~~ jokin kohdattu sana ei esiinny harjoitusaineistossa kertaakaan. Tällöin~~$e(w|t) = 0$~~ , jos sana w_i on tuntematon, on $e(w_i|t_i) = 0$ millä tahansa sanaluokalla ~~t, ja samoin~~ t_i. Samoin $p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) = 0$, jos yksikään sanoista ~~$w_1 \ldots w_n$~~ $w_i \ldots w_n$ on tuntematon. Jotta vältytään mallin rikkovilta nollaestimaateilta, on tuntemattomien sanojen kohdalla sovellettava jonkinlaista poikkeuskäsittelyä.

Yksinkertaisin ratkaisu on määrätä tuntemattoman sanalle aina harjoitusaineiston yleisin sanaluokka, ~~kä~~yleensää ~~ytännössä~~ substantiivi. Joitain kieliä — kuten englantia — tulkittaessa voidaan saavuttaa parempia tuloksia suffiksianalyysin (Samuelsson, 1993) avulla. Tällöin tarkoituksena on hyödyntää sitä seikkaa, että sanan pääte on usein vahva indikaattori sen sanaluokasta. Lisäksi Toutanova ym. (2003) ovat esittäneet, kuinka seuraavassa kappaleessa esiteltäviä log-lineaarisia malleja voidaan hyödyntää tuntemattomien sanojen käsittelyssä.

Bikel ym. (1999) esittivät vaihtoehtoisen, pseudosanoihin perustuvan menetelmän tuntemattomien sanojen käsittelylle. Menetelmän perusajatuksena on korvata tunnistettavan aineiston kukin tuntematon sana jollakin pseudosanalla, joita on rajallinen määrä. Myös kaikki harvinaiset (vähemmän kuin 5 esiintymää) sanat voidaan korvata pseudosanoilla. Korvaava pseudosana määräytyy aina tuntemattoman sanan ominaisuuksien mukaan: esimerkiksi *isoAlkukirjain*, *lauseenEnsimmäinenSana* sekä *neljäNumeroa* ovat tyypillisiä pseudosanoja. Nyt kun harjoitusaineiston harvinaiset sanat korvataan vastaavilla

pseudosanoilla, voidaan tunnistettavan aineiston pseudosanoja käsitellä samoin kuin tavallisia sanoja.

4.3.3 Viterbin algoritmi

Tässä kappaleessa kuvaillaan lyhyesti Viterbin algoritmia, jolla ratkaistaan tehokkaasti em. arvo $\arg\max_{t_1 \ldots t_n} p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n)$.

4.4 Parannukset (tjsp)

Tähän kuvaukset Cyclic Dependency Network-menetelmästä (Toutanova ym., 2003), mahd. ohjaamattomasta oppimisesta (Banko & Moore, 2004).

~~The trigram assumption is arguably quite strong, and linguistically naive. However, it leads to models that are very useful in practice.~~

5 Log-lineaariset mallit

TODO

6 Menetelmien vertailua/analyysia

TODO Tässä kappaleessa esitellään tunnistimien eri toimintaympäristöt (esim. erilaiset kielet, aineistot, harjoitusaineiston saatavuus), ja pohditaan miten eri menetelmät toimivat eri olosuhteissa; vastataan siis tutkimuskysymykseen (miten eri menetelmät eroavat toisistaan, ja soveltuvatko jotkin menetelmät toisia paremmin tietyille toimintaympäris

Toisaalta voisi olla mielekkäämpää perustella menetelmien etuja jo aikaisemmissa kappaleissa, niiden esittelyiden yhteydessä. Menetelmien esitysjärjestys on sellainen, että myöhempi menetelmä tavallaan vastaa aikaisemman menetelmän puutteisiin. Lisäksi tässä kappaleessa esiteltäviä johtopäätöksiä voisi jättää yhteenveto-kappaleeseen; kokonaisen kappaleen verran johtopäätöksiä ja merkityksellistä analyysia vaikuttaa turhan kunnianhimoiselta tavoitteelta. Harkitsen siis tämän kappaleen poistamista.

6 Yhteenveto

TODO Yhteenvedossa kerrataan työn pääkohdat lyhyehkösti johtopäätöksiä tehden. Siinä voi myös esittää pohdintoja siitä, minkälaisia tutkimuksia aiheesta voisi jatkossa tehdä. viittaa kirjallisuuskatsauksen tarkoitukseen ja kertoo ”päätulokset”. TODO vastataan tutkimuskysymykseen

Kirjallisuutta

Banko, M. & Moore, R. C. ~~(2004).~~ 2004. *Part of speech tagging in context*. Proceedings of the 20th conference on Computational Linguistics, ACL, s. 556.

Bikel, D. M., Schwartz, R., & Weischedel, R. M. 1999. *An algorithm that learns what's in a name*. Machine learning, 34(1-3), s. 211–231.

Brants, T. 2000. *TnT - A statistical part-of-speech tagger*. Proceedings of the 6th Applied NLP Conference (ANLP).

Brill, E. 1992. *A simple rule-based part of speech tagger*. Proceedings of the 3rd conference on Applied Computational Linguistics, ACL, Trento, Italy.

Brill, E. ~~1995.~~ 1994. ~~*Transformation-based error-driven learning and natural language processing: a case study*~~ *Some advances in* ~~*part-of-speech*~~ *transformation-based part of speech tagging*. ~~Computational Linguistics, 21(4),~~ AAAI 1994, s. ~~543-565~~722-727.

Charniak, E., Hendrickson, C., Jacobson, N., & Perkowitz, M. 1993. *Equations for part-of-speech tagging*. Proceedings of AAAI-93, s. 784–789.

Church, K. 1988. *A stochastic parts program and noun phrase parser for unrestricted text*. Proceedings of the 2nd conference on Applied Natural Language Processing, s. 136–143.

Cutting, D., Kupiec, J., Pedersen, J. & Sibun, P. 1992. *A Practical Part-of-Speech Tagger*. Proceedings of the 3rd conference on Applied Natural Language Processing, s. 133–140.

DeRose, S. J. 1988. *Grammatical category disambiguation by statistical optimization*. Computational Linguistics, 14(1), s. 31–39.

~~Francis, W. N. 1964. *A standard sample of present-day English for the use with digital computers*. Report for the U.S Office of Education on Cooperative Research Project No. E-007. Brown University, Providence RI.~~

Garside, R. 1987. *The CLAWS word-tagging system*. Teoksessa R. Garside, G. Leech & G. Sampson (toim.) The Computational Analysis of English: A Corpus-based Approach. London: Logman, s. 30-41.

Greene, B. B., & Rubin, G. M. 1971. *Automatic grammatical tagging of English*. Department of Linguistics, Brown University, 1971.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R. & Kadie, C. M. 2000. *Dependency networks for inference, collaborative filtering and data visualization*. Journal of Machine Learning Research, 1(1), s. 49-75.

Jaynes, E. T. 1957. *Information Theory and Statistical Mechanics*. Physical Review, 106, s. 620-630.

Manning, C. D. 2011. *Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?* Proceedings of the 12th international conference on Computational linguistics and intelligent text processing, 1, s. 171–189.

Manning, C. D. & ~~Schä~~Schütze, H. 1999. *Foundations of statistical natural language processing*. Cambridge, MA~~:~~. MIT Press

Marcus, M. P., Marcinkiewicz, M. & Santorini, B. 1993. *Building a large annotated corpus of English: the Penn treebank*. Computational Linguistics, 19(2), s. 313-330.

Merialdo, B. 1994. *Tagging English text with a probabilistic model*. Computational Linguistics, 20(2), s. 155-171.

~~Petrov, S. , Das, D~~

Ng, A. Y. & ~~McDonald, R. 2011.~~ Jordan, M. 2002. *On Discriminative vs. Generative Classifiers: A ~~universal part-of-speech tagset~~comparison of logistic regression and Naive Bayes*. ~~ArXiv:1104.2086.~~

In NIPS 14.

POS Tagging State of the Art. 2013. The Wiki of the Association for Computational Linguistics. Haettu 28.10.2013, osoitteesta `aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)`

Rabiner, L. R. 1989. *A tutorial on Hidden Markov Models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2), s. 257-285.

Ratnaparkhi, A. 1996. *A maximum entropy model for part-of-speech tagging*. Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP), University of Pennsylvania.

Ratnaparkhi, A. 1997. *A simple introduction to maximum entropy models for natural language processing*. Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania, 1997.

Samuelsson, C. 1993. *Morphological tagging based entirely on Bayesian inference*. Procee-

dings of the 9th Nordic Conference on Computational Linguistics NODALIDA-93.

Shen, L., Satta, G. & Joshi, A. 2007. *Guided learning for bidirectional sequence classification*. In: ACL 2007. (2007)

Spoustova, D.j., Hajic, J., Raab, J. & Spousta, M. 2009. *Semi-supervised training for the averaged perceptron POS tagger*. Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), s. 763-711.

Søgaard, A. 2010. *Simple semi-supervised training of part-of-speech taggers*. Proceedings of the ACL 2010 Conference Short Papers, s. 205-208.

Toutanova, K., Klein, D., Manning, C.D. & Singer, Y. 2003. *Feature-rich part-of-speech tagging with a cyclic dependency network*. In: NAACL 3. (2003), s. 252-259

Tseng, H., Jurafsky, D. & Manning, C. 2005. *Morphological features help POS tagging of unknown words across language varities*. Proceedings of the 4th SIGHAN bakeoff.