Aleksi Pekkala

Sanaluokkien automaattisen tunnistamisen menetelmät

Tietotekniikan kandidaatintutkielma

18. 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 1

2 SANALUOKKIEN AUTOMAATTINEN TUNNISTAMINEN 3
2.1 Miksi sanaluokkien tunnistaminen on ongelmallista? 3
2.2 Automaattisten tunnistajien suorituskyky 5
2.3 Sanaluokkien tunnistajan vaatimukset 6
2.4 Harjoitusaineisto ja sanaluokkasetit 6

3 SÄÄNTÖPOHJAISET MENETELMÄT 8
3.1 Brillin sääntöpohjainen sanaluokkatunnistin 9
3.2 Arviota 12

4 TILASTOLLISET MENETELMÄT 14
4.1 Markovin malli 15
4.2 Lähtökohta 16
4.3 Tunnistusongelma Markovin piilomallina 17
4.4 Cyclic Dependency Network (TODO mahd. suomennos) 19
4.5 Arviota 20

5 YHTEENVETO 21

KIRJALLISUUTTA 22

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 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, esimerkiksi kielten sekä tekstilajien osalta. Lisäksi havaitaan, että nykyisten tunnistusmenetelmien saavuttamat tunnistustarkkuudet liikkuvat kaikki suunnilleen samoissa lukemissa. Kun ilmiselvin valintakriteeri on näin poissuljettu, on tehokkaimman menetel-

män valinta vaikeampaa. Tunnistusmenetelmien toimintaperiaatteiden vaihdellessa merkittävästi on kuitenkin väistämätöntä, että jotkin menetelmät soveltuvat toisia paremmin tiettyihin tunnistustehtäviin. Tässä tutkielmassa pyritäänkin selventämään sitä, millaisia eri ratkaisuja sanaluokkien tunnistusongelmaan on olemassa ja mitkä ovat niiden tärkeimmät 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 3-4 tarkastellaan kahta erilaista lähtökohtaa tunnistusongelman ratkaisemiseksi: sääntöpohjaista Brillin tunnistinta sekä tilastollista Markovin piilomalleihin perustuvaa tunnistinta. Luvuissa esitellään menetelmien keskeiset ominaisuudet, sekä pyritään hahmottamaan niiden suhteelliset vahvuudet ja heikkoudet. Lopuksi vielä kootaan yhteen menetelmistä kerätyt huomiot ja esitellään johtopäätökset.

TODO maininta menetelmien valintaperusteista ja/tai esitysjärjestyksestä

2 Sanaluokkien automaattinen tunnistaminen

Sanaluokkien tunnistaminen on tärkeä ja käytännöllinen ongelma, joka kohdataan lähes kaikissa luonnollisten kielten käsittelyyn liittyvissä tehtävissä. Tällaisia tehtäviä ovat muun muassa 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 Miksi sanaluokkien tunnistaminen on ongelmallista?

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

Time flies like an arrow.

Fruit flies like a banana.

sana *flies* esiintyy ensin verbinä ja sitten substantiivina. Sanan *time* ilmeisin sanaluokka on substantiivi, mutta se voidaan mieltää myös imperatiiviverbinä, jolloin lauseen merkitys muuttuu täysin. Itse asiassa kummatkin esimerkkilauseet voidaan tulkita kymmenin eri tavoin, joista ilmeisimmän tulkinnan valitseminen automaattisesti on haastavaa. Lisäksi, vaikka tosielämässä tulkittavat lauseet ovat harvoin yhtä ongelmallisia kuin edellämainitut lingvistiset esimerkkilauseet, on monitulkintaisuus hyvin yleistä: arviolta 40% englanninkielisen proosan sanastosta voidaan luokitella useampaan kuin yhteen sanaluokkaan (DeRose, 1988).

Jatkokäsittelyn kannalta automaattisen tunnistajan oleellisin tehtävä onkin valita kaikista mahdollisista sanaluokista se, joka tuottaa luontevimman tulkinnan. Tällaisen yksikäsitteistämisen mahdollistavat luonnollisten kielten sisäänrakennetut rajoitteet, jotka voidaan jakaa lokaaleihin sekä kontekstuaalisiin vihjeisiin: lokaaleista vihjeistä ilmeisin on itse sana (”sana *can* on on todennäköisemmin modaaliverbi kuin substantiivi”), mutta päätelmiä voidaan tehdä myös muun muassa 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.

On tärkeää huomata, ettei sanaluokkien tunnistaminen itsessään ole ratkaisu kieliopilliseen monitulkintaisuuteen: monitulkintaisuudella on useita tasoja, joista osaa käsitellään vielä prosessointiketjun myöhemmissä vaiheissa. Esimerkiksi syntaktinen eli 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 ylimääräiseltä työltä. (Manning & Schütze, 1999, s. 341)

Monitulkintaisuuden lisäksi toinen merkittävä ongelma on tuntemattomien eli harjoitusaineistosta puuttuvien sanojen käsitteleminen. 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, johon tuntemattomien sanojen on havaittu todennäköisimmin kuuluvan, eli yleensä substantiivi. Parempia tuloksia on saavutettu määrittämällä tuntemattoman sanan sanaluokka sen päätteen perusteella; esimerkiksi englannin kielen *able-*

päätteiset sanat ovat hyvin todennäköisesti adjektiiveja (Samuelsson, 1993). Menetelmä ei kuitenkaan sovellu kaikille kielille: esimerkiksi 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.

2.2 Automaattisten tunnistajien suorituskyky

Kuten mainittua, nykyisten automaattisten sanaluokkien tunnistajien tunnistustarkkuus — englanninkielistä kirjakieltä analysoitaessa — on hieman yli 97% (mm. Toutanova ym., 2003, Shen ym., 2007, Spoustova ym., 2009). Manning (2011) kuitenkin osoittaa, että kyseistä tulosta ei ole syytä tulkita liian optimistisesti: esimerkiksi lukuisat välimerkit 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 aineistot eivät saavuta täydellistä tunnistamistarkkuutta: yksittäisen ihmisen tunnistustarkkuuden 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. Myös tunnistajan mahdollisen opettamisen tulisi onnistua kohtuullisen nopeasti.

Tarkkuus Tunnistajan tulee kyetä ehdottamaan sanaluokka jokaista annettua sanaa kohden.

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

Sanaluokkien tunnistaminen on luonteeltaan sarjanluokitteluongelma, joka taas on yksi koneoppimisen (tarkemmin hahmontunnistuksen) aliongelmatyypeistä. Tämän seurauksena nykyiset sanaluokkien tunnistusmenetelmät perustuvat usein ohjattuun oppimiseen, eli tunnistimet tulee ”opettaa” jonkinlaisella harjoitusaineistolla ennen käyttöä. Sanaluokkien tunnistimille syötettävä harjoitusdata koostuu suurista tekstiaineistoista (engl. *corpus*), joissa jokaisen sanan yhteyteen on merkattu oikea sanaluokka. Nykyisin ehkä yleisimmin käytetty englanninkielinen harjoitusaineisto on *Penn Treebank*-aineisto (Marcus ym., 1993), joka sisältää noin viiden miljoonan sanan edestä uutisartikkeleita, kaunokirjallisuutta, tieteellisiä julkaisuja ynnä muita tekstilajeja. Kaikki tässä tutkielmassa raportoidut tunnistustarkkuudet ovat mitattu

Penn Treebank-aineistolla.

Tunnistusongelmaan on olemassa myös 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 määräytyvät yleensä harjoitusaineiston mukaan: esimerkiksi Penn Treebank-aineiston käyttämä sanaluokkasetti koostuu 48 sanaluokasta, joista 12 ovat välimerkkejä. On tärkeää huomata, että käytettävän sanaluokkasetin laajuus vaikuttaa suoraan tunnistustarkkuuteen: mitä enemmän sanaluokkia, sitä suurempi mahdollisuus monitulkintaisuuteen. Toisaalta sanaluokkien tunnistamisen tuottama lingvistinen informaatio on sitä arvokkaampaa, mitä tarkempi jaottelu eri sanaluokkien välillä on. (Marcus ym., 1993)

TODO esimerkkipätkä harjoitusaineistosta, jotain siitä miksi suomen kieli ym. vahvasti synteettiset/agglutinatiiviset kielet ovat hankalia sanaluokkien tunnistamisen kannalta.

3 Sääntöpohjaiset menetelmät

Ensimmäiset automaattiset sanaluokkatunnistimet (mm. Greene & Rubin, 1971) olivat lähtöisin kielitieteen piiristä, ja ne perustuivat pitkälti käsin laadittuihin kieliopillisiin sääntöihin. Tällainen sääntöpohjainen menetelmä toimii seuraavasti: ensin kunkin sanan mahdolliset sanaluokat haetaan sanakirjasta tai vastaavasta tietolähteestä. Seuraavaksi monitulkintaisten sanojen kohdalla sovelletaan valmiita kielioppisääntöjä sanaluokkien poissulkemiseen, kunnes jäljelle on enään yksi sanaluokka. Kielioppisäännöt voivat hyödyntää sekä lokaaleja että kontekstuaalisia vihjeitä; tyypillisiä sääntöjä ovat esimerkiksi

1. *hylkää sanaluokka x jos sanalla on iso alkukirjain* (lokaali vihje), sekä
2. *hylkää sanaluokka x, jos edeltävä sanaluokka on y* (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 erikoispiirteistä, 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 muutostöitä.

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. 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 kieliop-

pisääntöjä käytettäessä. (Brill, 1992)

Brill (1992) huomauttaakin, että tilastolliset tunnistajat saavuttavat korkean tunnistamistarkkuuden 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) esittää artikkelissaan vaihtoehtoisen lähestymistavan, joka pohjautuu varhaisimpien tunnistusmenetelmien tavoin kieliopillisiin sääntöihin. Aikaisemmista sääntöpohjaisista menetelmistä poiketen sääntöjä ei kuitenkaan syötetä manuaalisesti, vaan tunnistin oppii ne automaattisesti oikeilla sanaluokilla merkitystä harjoitusaineistosta. Menetelmän kantava idea on (1) aloittaa jostakin yksinkertaisesta ratkaisusta, (2) tunnistaa tehdyt virheet ja (3) inkrementaalisesti soveltaa virheitä korjaavia transformaatiosääntöjä, kunnes ne eivät enään paranna kokonaistarkkuutta.

Brillin tunnistinta alustettaessa harjoitusaineisto jaetaan kahteen osaan, joista pienempää käytetään niin sanottuna sääntöaineistona (engl. *patch corpus*) ja suurempaa varsinaisena harjoitusaineistona. Tunnistimen alustus alkaa siten, että sääntöaineiston kukin sana luokitellaan ensin sillä sanaluokalla, joka useimmiten esiintyy sanan yhteydessä harjoitusaineistossa. Tuntemattomien sanojen kohdalla sanaluokka määräytyy sanan kolmen viimeisen kirjaimen mukaan: esimerkiksi kaikki *ous*-päätteiset tuntemattomat sanat luokitellaan adjektiiveiksi, koska harjoitusaineiston *ous*-päätteiset sanat ovat useimmiten adjektiiveja. Lisäksi kaikki isolla alkukirjaimella alkavat tuntemattomat sanat luokitellaan erisnimiksi. Tämän yksinkertaisen menetelmän tarkkuus on noin 92%. (Brill, 1992)

Seuraavaksi verrataan edellämainitulla menetelmällä tunnistettuja sanaluokkia sääntöaineiston oikeisiin sanaluokkiin. Vertailun tuloksena saadaan lista virheistä muodossa $< tag_a, tag_b, number >$, josta ilmenee montako kertaa jokin sana tunnistettiin

luokkaan tag_a, kun oikea sanaluokka olisi ollut tag_b. Nyt käyttämällä valmiita sääntörunkoja voidaan laskennallisesti selvittää se transformaatiosääntö, joka laskee virheprosenttia eniten. Kunkin virhe-sääntörunko-parin muodostamaa sääntöä siis sovelletaan vuorostaan sääntöaineistoon, ja säännön arvo lasketaan vähentämällä korjattujen virheiden määrästä mahdollisesti aiheutettujen uusien virheiden lukumäärä. Brill (1992) käyttää artikkelissaan seuraavia sääntörunkoja:

Vaihda sanaluokka *a* sanaluokkaan *b*, kun:

1. Edellinen (seuraava) sanaluokka on *z*.
2. Edellistä edeltävä (seuraavaa seuraava) sanaluokka on *z*.
3. Jokin kahdesta edellisestä (seuraavasta) sanaluokasta on *z*.
4. Jokin kolmesta edellisestä (seuraavasta) sanaluokasta *z*.
5. Edellinen sanaluokka on *z* ja seuraava sanaluokka on *w*.
6. Edellinen (seuraava) sanaluokka on *z* ja seuraavaa seuraava (edellistä edeltävä) sanaluokka on *w*.
7. Nykyinen sana alkaa isolla (pienellä) alkukirjaimella.
8. Edellinen sana alkaa isolla (pienellä) alkukirjaimella.

Kun arvokkain transformaatiosääntö on löydetty, sääntö laitetaan muistiin ja sääntöaineistoon tehdään löydetyn säännön mukaiset muutokset. Prosessia jatketaan keräämällä taas lista sääntöaineiston tunnistusvirheistä, ja etsimällä uusi paras transformaatiosääntö. Tätä toistetaan, kunnes ei enään löydetä sellaista sääntöä, joka tuottaisi enemmän korjauksia kuin aiheuttaisi uusia virheitä. Tällöin alustamisprosessi on valmis, ja tuloksena saatuja transformaatiosääntöjä voidaan soveltaa uuden aineston tunnistamiseen seuraavasti: ensin aineisto luokitellaan edellämainitulla yksinkertaisella menetelmällä. Tämän jälkeen aineistoon sovelletaan kutakin alustusvaiheessa löydettyä transformaatiosääntöä, jolloin virheprosentti laskee. (Brill, 1992)

3.1.1 Tunnistimen laajentaminen

Yksi Brillin alkuperäisen tunnistimen puutteista on se, että käytetyt sääntörungot huomioivat pelkästään sanaluokkien väliset suhteet: itse sanoihin liittyvää kontekstuaalista informaatiota ei käsitellä lainkaan. Tällöin merkittävä osa kielen kontekstuaalisista vihjeistä jää hyödyntämättä. Tätä puutetta voidaan paikata laajentamalla tunnistinta seuraavilla sääntörungoilla:

Vaihda sanaluokka a sanaluokkaan b, kun:

1. Edellinen (seuraava) sana on w.
2. Edellistä edeltävä (seuraavaa seuraava) sana on w.
3. Jokin kahdesta edellisestä (seuraavasta) sanasta on w.
4. Nykyinen sana on w ja edellinen (seuraava) sana on x.
5. Nykyinen sana on w ja edellinen (seuraava) sanaluokka on z.

Tunnistimen leksikalisointi edellä mainituilla säännöillä vähentää tunnistusvirheiden määrää noin 10 prosentilla. (Brill, 1994)

Tunnistimen toinen merkittävä puute on tuntemattomien sanojen heikko tunnistustarkkuus. Tämä ei ole varsinaisesti yllättävää, sillä edellä kuvailtu sanojen pääteliitteisiin perustuva tuntemattomien sanojen erikoiskäsittely jää melko pinnalliseksi: lokaaleista vihjeistä huomioidaan vain pääte sekä alkukirjaimen koko, ja kontekstuaaliset vihjeet sivuutetaan täysin. Tätäkin ongelmaa voidaan lieventää laajentamalla tunnistinta uusilla sääntörungoilla:

Vaihda tuntemattoman sanan sanaluokka a sanaluokkaan b, kun:

1. Etuliitteen x, $|x| \leq 4$, poistamisesta seuraa uusi sana.
2. Sanan ensimmäiset (1,2,3,4) kirjainta ovat x.
3. Pääteliitteen x, $|x| \leq 4$, poistamisesta seuraa uusi sana.
4. Sanan viimeiset (1,2,3,4) kirjainta ovat x.
5. Merkkijonon x lisäämisestä sanan pääteliitteeksi seuraa uusi sana ($|x| \leq 4$).
6. Merkkijonon x lisäämisestä sanan etuliitteeksi seuraa uusi sana ($|x| \leq 4$).
7. Edellinen (seuraava) sana on w.

8. Sana sisältää kirjaimen *z*.

Huomataan, ettei yksikään sääntörunko ota kantaa sanaluokkiin, koska tuntemattomat sanat käsitellään ennen varsinaisten transformaatiosääntöjen soveltamista. (Brill, 1994)

Näillä kahdella lisäyksellä Brillin tunnistin saavuttaa 96,6% tunnistustarkkuuden, missä tuntemattomien sanojen tunnistustarkkuus on 85%. (Brill, 1995)

3.2 Arviota

TODO keskeneräinen luku, asiasisältö on jokseenkin selvillä.

1. Sääntörunkojen kanssa ei tarvitse olla tarkkana - huonot rungot eivät kerää sääntöjä. Sääntöjä ei tarvita paljon. Säännöt ovat helpompia tulkita kuin tilastodata. Sääntörunkojen laatiminen vaatii silti jonkinlaista ymmärrystä kielestä. 'Can capture more context than Markov models".

2. Koska säännöt ovat niin yksinkertaisia, on mahdollista parantaa tarkkuustuloksia tekemällä käsin uusia sääntöjä (Volk & Schneider, 1998).

3. Ei voi toimia osana probabilistista järjestelmää, koska ei käsitellä todennäköisyyksiä. Ainakaan ilman suuria muutoksia?

4. Opettaminen on hidasta (tunneista kymmeniin tunteihin) — apuun FastTBL-algoritmi (Ngai & Florian, 2001).

5. Soveltaminen on hidasta - "show a method for converting a list of tagging transformations into a deterministic finite state transducer with one state transition taken per word of input; the result is a transformation-based tagger whose tagging speed is about ten times that of the fastest Markov-model tagger." (Roche & Schabes, 1995)

6. Tarkkuus on noin prosenttiyksikön jäljessä nykyisistä huipputuloksista. Poikkeuksellisen matala tuntemattomien sanojen tunnistustarkkuus — voisiko käsittelyä parantaa? Brill (1995) myös väittää, että jos aineistopohjaisessa luonnollisten

kielten käsittelyssä halutaan saavuttaa edistysaskeleita, on pyrittävä ymmärtämään itse kieltä tilastollisten rinnakkaisilmiöiden sijaan. Tilastollisia menetelmiä on kehitetty (Brillin menetelmän jälkeen) jo 20 vuotta, mutta tarkkuus on noussut vain prosenttiyksikön, oliko Brill oikeassa? (Manning, 2011)

4 Tilastolliset menetelmät

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

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

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

Tilastolliset menetelmät voidaan vuorostaan jakaa diskriminatiivisiin ja generatiivisiin malleihin. Diskriminatiiviset mallit käsittelevät suoraan todennäköisyyttä $P(y|x)$, eli ne eivät ota kantaa syötteeseen x. Generatiiviset mallit 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 ja Jordan (2002) kuitenkin osoittavat, ettei tämä välttämättä aina pidä paikkaansa: sanaluokkien tunnistamiseen onkin olemassa kumpaankin malliin perustuvia tehokkaita ratkaisuja. Seuraavassa luvussa esitellään generatiivinen Markovin piilomalli; diskriminatiiviseen menetelmään perustuvan ratkaisun on esittänyt mm. Ratnaparkhi (1996).

4.1 Markovin malli

Markovin malli (mm. Rabiner, 1989) kuvaa sellaista stokastista prosessia, joka toteuttaa ns. Markov-ominaisuuden: seuraava tila riippuu aina vain N:stä edeltävästä tilasta. Markov-ominaisuus on siis eräänlainen riippumattomuusoletus, joka yksinkertaistaa stokastisen prosessin tilan estimointia rajoittamalla tilasiirtymien historian määrää. N:nen asteen Markov-ominaisuuden toimiessa esimerkiksi todennäköisyys

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

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-ominaisuudesta, jolloin $N = 1$. Käytännössä tämä tarkoittaa sitä, että tarkastellaan jotakin kahta peräkkäistä tilaa — nykyistä sekä tulevaa. Markov-ominaisuutta voidaan kuitenkin laajentaa myös korkeampiin asteisiin, jolloin myös tarkasteltavien tilasarjojen pituudet kasvavat. Sanaluokkia tunnistaessa näitä 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 todennäköisyysfunktio.

TODO: tähän kuva Markovin piilomallista

4.2 Lähtökohta

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

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

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

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

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 piilomallit hyödyntävät sitä yhteisjakauman laskusääntöä $P(x,y) = P(y)P(x|y)$. Tällöin edellämainittu funktio p (4.4) voidaan kuvata muodossa (TODO: parempi selitys siitä miten parametrit vastaavat yhteisjakauman osia, e = P(x | y) ja q = P(y)

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

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

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

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

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

laskee todennäköisyyden sanaluokalle t_i, kun kaksi edeltävää sanaluokkaa ovat t_{i-1} ja t_{i-2}. Parametri voidaan myös mieltää todennäköisyytenä trigrammille t_{i-2}, t_{i-1}, t_i. Tästä huomataan, että kyseessä on toisen asteen Markovin piilomalli. Mallin toinen parametri

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

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

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

missä funktio f merkitsee annetun n-grammin lukumäärää harjoitusaineistossa. Brants (2000) kuitenkin osoittaa, että datan harvuuden vuoksi tällainen estimaatti ei ole käyttökelpoinen: laajassakaan harjoitusaineistossa ei ole tarpeeksi montaa kappaletta kutakin eri trigrammia. Lisäksi osa trigrammeista t_{i-2}, t_{i-1}, t_i ovat väistämättä sellaisia, että $f(t_{i-2}, t_{i-1}, t_i) = 0$, 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 bigrammien suhteellisia frekvenssejä:

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

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

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

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

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

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:

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

TODO: Maininta Viterbi-algoritmista (Viterbi, 1967).

4.3.2 Tuntemattomien sanojen käsittely

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

Yksinkertaisin ratkaisu on määrätä tuntemattoman sanalle aina harjoitusaineiston yleisin sanaluokka, yleensä 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 diskriminatiivisia 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.

4.4 Cyclic Dependency Network (TODO mahd. suomennos)

TODO Tähän kuvaus Cyclic Dependency Network-menetelmästä (Heckerman ym., 2000; Toutanova ym., 2003), jolla HMM-tunnistin voi oppia laajemmin kontekstuaalisista vihjeistä.

4.5 Arviota

TODO

1. Trigrammi-oletus on melko vahva ja lingvistisesti naiivi — johtaa kuitenkin käytännössä hyödyllisiin malleihin.

2. Ei poimi sanojen välisiä suhteita, pelkästään sanaluokkien.

3. ???

5 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”. Vastataan tutkimuskysymykseen.

Kirjallisuutta

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. 1994. *Some advances in transformation-based part of speech tagging*. AAAI 1994, s. 722–727.

Brill, E. 1995. *Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging*. Computational linguistics, 21(4), s. 543–565.

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

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.

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.

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. & 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.

Ngai, G. & Florian, R. 2001. *Transformation-based learning in the fast lane*. Proceedings of the 2nd meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, s. 1–8.

Ng, A. Y. & Jordan, M. 2002. *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes*. 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.

Roche, E., & Schabes, Y. 1995. *Deterministic part-of-speech tagging with finite-state transducers*. Computational linguistics, 21(2), s. 227–253.

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

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.

Viterbi, A. 1967. *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*. Information Theory, IEEE Transactions on, 13(2), s. 260–269.

Volk, M. & Schneider, G. 1998. *Comparing a statistical and a rule-based tagger for German*. Proceedings of KONVENS-98.