Aleksi Pekkala

Sanaluokkien automaattisen tunnistamisen menetelmät

Tietotekniikan kandidaatintutkielma

15. tammikuuta 2014

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: 27+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: tarkoituksena ei ole jäsentää kokonaisia lauserakenteita tai tulkita lauseiden merkitystä — tarkastelun alla ovat vain yksittäisten sanojen sanastolliset kategoriat. Sanaluokkien tunnistaminen on kuitenkin välttämätön ensimmäinen askel monissa luonnollisten kielten käsittelyprosesseissa, ja siten yksi aihealueen keskeisistä osaongelmista. Lisäksi tunnistusongelman eri ratkaisut ovat usein suoraan siirrettävissä muiden yksikäsitteistämisongelmien, kuten sanan merkitysanalyysin piiriin (Brill, 1995).

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. (Manning & Schütze, 1999, s. 342)

Sanaluokkien tunnistimia käytetään monissa erilaisissa luonnollisiin kieliin liittyvissä sovelluksissa, ja tunnistimille 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 sopivimman menetelmän valinta vaikeampaa. Tunnistusmenetelmien toimintaperiaatteiden vaihdellessa mer-

kittävästi 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 muunmuassa puheentunnistus, konekääntäminen sekä semanttinen haku ja analyysi. Kyseisissä tehtävissä sanaluokkien tunnistin 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 edellä mainitut lingvistiset esimerkkilauseet, on monitulkintaisuus hyvin yleistä: arviolta 40% englanninkielisen proosan sanastosta voidaan luokitella useampaan kuin yhteen sanaluokkaan (DeRose, 1988).

Jatkokäsittelyn kannalta automaattisen tunnistimen 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 (esimerkiksi 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. Tuntemattomien sanojen käsittely on suuri tekijä siinä, kuinka hyvin tunnistin on siirrettävissä erilaisille aineistoille: käytännössä tunnistimien eriävät tarkkuustulokset erilaisilla aineistoilla selittyvät enimmäkseen tuntemattomien sanojen määrällä sekä tunnistimen poikkeuskäsittelyn onnistumisella. (Manning & Schütze, 1999, s. 351)

2.2 Automaattisten tunnistimien suorituskyky

Kuten mainittua, nykyisten tunnistimien saavuttama tunnistustarkkuus on hieman yli 97% (mm. Spoustova ym., 2009; Søgaard, 2011). 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 edellä mainitullakin tunnistustarkkuudella jokaisessa lauseessa on keskimäärin yksi virhe (Manning & Schütze, 1999). Artikkelissa huomautetaankin, että realistisempaa olisi tarkastella automaattisten tunnistimien kykyä tunnistaa kokonaiset lauseet oikein, sillä pienikin virhe lauseessa voi vahingoittaa tunnistimen hyödyllisyyttä myöhempien prosessointivaiheiden kannalta; tällä saralla tunnistimet saavuttavat noin 55-57% tarkkuuden, mikä on huomattavasti vaatimattomampi tulos. Lisäksi Giesbrecht ja Evert (2009) ovat osoittaneet, että kyseiset huipputulokset ovat saavutettavissa vain hyvin keinotekoisissa koetilanteissa: todellisissa käyttötapauksissa tunnistimien tarkkuus laskee alle 93 prosentin, tarkkuuden vaihdellessa merkittävästi tekstilajeittain.

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 noin 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 edellä mainittua automaattisten tunnistimien huipputulosta.

2.3 Sanaluokkien tunnistimien vaatimukset

Jotta tunnistinta voidaan käyttää osana laajempaa kielenprosessointijärjestelmää, tulee sen toteuttaa seuraavat ominaisuudet (Cutting ym., 1992):

Kestävyys Tunnistimen tulee kyetä selviytymään kaikista tekstisyötteen mah-

dollisista poikkeamista, kuten otsikoista, taulukoista sekä tuntemattomista sanoista.

Tehokkuus Voidakseen käsitellä laajoja tekstiaineistoja tunnistimen tulee toimia lineaarisessa ajassa. Myös tunnistimen mahdollisen harjoittamisen tulisi onnistua kohtuullisen nopeasti.

Tarkkuus Tunnistimen tulee kyetä ehdottamaan sanaluokka jokaista annettua sanaa kohden.

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

Uudelleenkäytettävyys Tunnistimen 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, missä tunnistimet harjoitetaan jonkinlaisella harjoitusaineistolla ennen käyttöä. Sanaluokkien tunnistimille syötettävä harjoitusdata koostuu suurista tekstiaineistoista (engl. *corpus*), joissa jokaisen sanan yhteyteen on merkattu oikea sanaluokka. Tekstiaineistoista yleisin on englanninkielinen *Penn Treebank*-aineisto (Marcus ym., 1993), joka sisältää noin viiden miljoonan sanan edestä uutisartikkeleita, kaunokirjallisuutta, tieteellisiä julkaisuja ynnä muita tekstilajeja. Penn Treebank-aineistoa käytetään usein myös tunnistimien evaluointiin: tällöin on tärkeää käyttää sellaista harjoitusaineiston osaa, jota ei ole käytetty varsinaiseen harjoittamiseen. Myös kaikki tarkkuustulokset, joihin tässä tutkielmassa viitataan, on mitattu Penn Treebank-aineistolla.

Tunnistusongelmaan on olemassa myös ohjaamattomaan oppimiseen perustuvia ratkaisuja, jotka eivät vaadi ennalta luokiteltua harjoitusaineistoa. Toisaalta tällai-

set 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)

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 vain 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).

Kyseisen 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 tunnistimet 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)

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 enää 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 edellä mainitulla 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 lukumää-

rä. 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 enää 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 mainitulla yksinkertaisella menetelmällä. Tämän jälkeen aineistoon sovelletaan kutakin alustusvaiheessa löydettyä transformaatiosääntöä, jolloin virheprosentti laskee. (Brill, 1992)

3.2 Brillin 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. Kyseistä 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)

Brillin 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. Näiden kahden lisäyksen myötä Brillin tunnistin saavuttaa 96,6% tunnistustarkkuuden, missä tuntemattomien sanojen tunnistustarkkuus on 85%. (Brill, 1994, 1995)

Edellisten puutteiden lisäksi Brillin tunnistin on itsessään suhteellisen epätehokas: Ngai ja Florian (2001) mainitsevat, että tunnistimen harjoittaminen miljoonan sanan

aineistolla vie tyypillisesti yli 38 tuntia. Samoin Volk ja Schneider (1998) raportoivat Brillin tunnistimen harjoittamisen vievän päiviä siinä missä vastaava tilastollinen tunnistin harjoitetaan minuuteissa. Harjoittamisen nopeuttamiseksi on esitetty inkrementaalinen menetelmä, jonka avulla tunnistin skaalautuu paremmin laajoille harjoitusaineistoille (Ramshaw & Marcus, 1994). Menetelmässä jokainen transformaatiosääntö sisältää listan niistä sääntöaineiston kohdista, joissa sääntöä voidaan soveltaa. Samoin jokainen sääntöaineiston indeksi sisältää listan kyseisessä kohdassa sovellettaviin sääntöihin. Näiden indeksien avulla vältytään koko sääntöaineiston läpikäymiseltä transformaatiosäännön arvoa laskiessa. Inkrementaaliseen menetelmään perustuva FastTBL-algoritmi suoriutuu mainitusta 38 tunnin harjoittamisesta noin 17 minuutissa (Ngai & Florian, 2001). Harjoittamisen nopeuttamisen lisäksi Roche ja Schabes (1995) ovat osoittaneet kuinka tunnistimen käyttämistä voidaan nopeuttaa muokkaamalla löydetyt transformaatiosäännöt eräänlaiseksi äärelliseksi automaatiksi: artikkelissa raportoidaan noin 20-kertainen tunnistusnopeus Brillin alkuperäiseen tunnistimeen verrattuna.

3.3 Brillin tunnistimen arviointi

Brillin tunnistin osoittaa, että sääntöpohjaisilla menetelmillä voidaan saavuttaa sekä kilpailukykyisiä tarkkuustuloksia että tilastollisia menetelmiä vastaava helppokäyttöisyys. Tunnistin myös säilyttää kaikki aikaisemmin mainitut sääntöpohjaisten menetelmien edut: transformaatiosäännöt peittoavat vastaavan tilastodatan havainnollisuudessa, kompaktiudessa sekä muokattavuudessa. Yltääkseen esimerkiksi siihen tarkkuuteen, mihin Markovin malleihin perustuva tilastollinen tunnistin vaatii 10 000 todennäköisyyslukemaa, Brillin tunnistin tarvitsee vain 217 transformaatiosääntöä (Brill, 1994). Tunnistimen kompaktiuden ansiosta sen tuloksia on helpompi analysoida ja tarvittaessa muokata manuaalisesti: esimerkiksi Schneider ja Volk (1998) ovat esittäneet, kuinka tunnistimen tarkkuutta voi parantaa huomattavasti hyödyntämällä lisäämällä sääntöjä käsin. Edellä esitettyjen laajennusten myötä Brillin tunnistin on myös hyvin tehokas.

Koska Brillin tunnistin sisältää itsessään hyvin vähän lingvististä tietoa, se on hel-

posti kohdennettavissa erilaisille aineistoille. Esimerkiksi saksankielisten tekstien analysointiin Brillin tunnistin soveltuu lähes sellaisenaan (Volk & Schneider, 1998). Megyesi (1999) puolestaan on osoittanut, että tunnistin on siirrettävissä myös agglutinatiivisille kielille käytettyjä sääntörunkoja muokkaamalla. Huipputulosten saavuttamiseksi Brillin tunnistin voi siis joissakin tapauksissa vaatia ainestokohtaista hienosäätöä. Toisaalta Brill (1992) huomauttaa, ettei sääntörunkojen laatimisessa tarvitse olla erityisen varovainen, sillä huonot sääntörungot eivät heikennä tunnistustarkkuutta: jos sääntörunko ei päde aineistoon, sen pohjalta ei luoda yhtään sääntöä. Täten erilaisten sääntölaajennusten kokeileminen on helppoa.

Brillin tunnistimen on myös havaittu olevan kohtalaisen immuuni ylisovittamiselle (Ramshaw & Marcus, 1994). Ylisovittaminen (engl. *overfitting*) tarkoittaa tilannetta, missä luotu malli kuvastaa hyvin harjoitusaineistoa, muttei enää päde laajemmalle aineistolle. Brillin tunnistin minimoi ylisovittamisen riskiä hylkäämällä ne säännöt, jotka esiintyvät tiettyä raja-arvoa harvemmin. Tämän ominaisuuden perusteella voidaan spekuloida, että tunnistin suoriutuu hyvin myös todellisissa käyttötapauksissa.

Yksi Brillin tunnistimen puutteista on se, että sen tulosten varmuus ei ole suoraan määritettävissä. Tilastollisten tunnistimien tuloksiin liittyy aina todennäköisyysarvo, jonka perusteella tunnistamisen onnistumista voidaan arvioida; Brillin tunnistin taas asettaa jokaiselle sanalle aina yhden sanaluokan ottamatta kantaa todennäköisyyksiin. Todennäköisyysarvot ovat hyödyllisiä esimerkiksi silloin, jos tunnistin ei pysty käsittelemään jotakin aineiston osaa tarpeeksi suurella varmuudella: tällöin voidaan yrittää samaa jollakin toisella tunnistimella ja käyttää sitä tulosta, jolla on suurempi todennäköisyys. Brillin tunnistin ei ilman laajoja muokkauksia pysty toimimaan osana tällaista järjestelmää. Lisäksi Brillin tunnistimen tarkkuus — vaikkakin aikanaan ennätyksellinen — on noin prosenttiyksikön jäljessä nykyisistä tilastollisista tunnistimista (Spoustova ym., 2009; Søgaard, 2011). Erityisesti tuntemattomien sanojen käsittely on Brillin tunnistimelle ongelmallista (Schneider & Volk, 1998).

4 Tilastolliset menetelmät

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

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

Tilastollisissa sanaluokkien tunnistusmenetelmissä syötteitä ja luokkia vastaavat lauseet sekä sanaluokkasarjat. Edelleen tavoitteena on löytää kullekin lauseelle sitä todennäköisimmin vastaava sanaluokkien sarja:

$$f(w_1, w_2, \ldots, w_n) = \arg\max_{t_1 \ldots t_n} p(w_1, w_2, \ldots, w_n, t_1, t_2, \ldots, t_n) \tag{4.2}$$

missä p ilmaisee todennäköisyyden sille, että jokin lause $w_1 \ldots w_n$ esiintyy jonkin sanaluokkasarjan $t_1 \ldots t_n$ yhteydessä. Mahdollisten sanaluokkasarjojen määrä kuitenkin kasvaa eksponentiaalisesti sanojen ja sanaluokkien määrän mukaan, jolloin maksimiarvon ratkaiseminen suoraan on käytännössä mahdotonta.

Tilastolliset menetelmät voidaan 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. Intuition mukaan diskriminatiivinen malli on generatiivista mallia tehokkaampi, sillä yhteisjakauman mallintaminen on laskennallisesti vaativampaa kuin ehdollisen jakauman mallintaminen. Ng ja Jordan (2002) kuitenkin osoittavat, ettei tämä välttämättä aina pidä paikkaansa: sanaluokkien tunnistamiseen onkin olemassa kumpaankin malliin perustuvia tehokkaita ratkaisuja. Tässä tutkielmassa käsitellään generatiiviseen Markovin piilomalliin perustuvaa tunnistinta (Brants, 2000);

diskriminatiivisen tunnistimen on esittänyt muun muassa Ratnaparkhi (1996).

Brill (1992) kritisoi tilastollisia tunnistusmenetelmiä siitä, että ne saavuttavat korkean tunnistustarkkuuden kiinnittämättä varsinaisesti huomiota aineiston taustalla olevaan kieliopilliseen rakenteeseen. Brill (1995) myös väittää, että jos aineistopohjaisessa luonnollisten kielten käsittelyssä halutaan saavuttaa edistysaskeleita, on pyrittävä ymmärtämään itse kieltä tilastollisten rinnakkaisilmiöiden sijaan. Väitettä tukee havainto siitä, ettei tilastollisilla menetelmillä ole virheanalyysin perusteella odotettavissa suuria korotuksia nykyiseen sanaluokkien tunnistustarkkuuteen (Manning, 2011).

4.1 Markovin malli

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

$$P(x_k|x_1,\ldots,x_{k-1}) \tag{4.3}$$

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

$$P(x_k|x_{k-N},\ldots,x_{k-1}) \tag{4.4}$$

Markovin ominaisuudesta puhuttaessa on yleensä kyse juuri ensimmäisen asteen ominaisuudesta, jolloin $N = 1$. Käytännössä tämä tarkoittaa sitä, että tarkastellaan jotakin kahta peräkkäistä tilaa — nykyistä sekä tulevaa. Markovin ominaisuutta voidaan kuitenkin laajentaa myös korkeampiin asteisiin, jolloin myös tarkasteltavien tilasarjojen pituudet kasvavat. Sanaluokkia tunnistaessa näitä tilasarjoja vastaavat n:n peräkkäisen sanaluokan sarjat, eli n-grammit. (Rabiner, 1989)

Markovin mallin hyödyllisyyttä rajoittaa se, että malli ei pysty itsessään mallinta-

maan 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. (Rabiner, 1989)

4.2 Tunnistusongelma Markovin piilomallina

Hyödyntämällä yhteisjakauman laskusääntöä $P(x,y) = P(y)P(x|y)$, kaavan 4.2 funktio p voidaan kuvata muodossa

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

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

$$= \prod_{i=1}^{n+1} q(t_i | t_{i-2}, t_{i-1}) \prod_{i=1}^{n} e(w_i | t_i) \tag{4.7}$$

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

$$q(t_i | t_{i-2}, t_{i-1}) \tag{4.8}$$

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

$$e(w_i | t_i) \tag{4.9}$$

kuvaa todennäköisyyttä sille, että sana w_i esiintyy sanaluokan t_i yhteydessä. Parametreja q ja e voidaan kutsua myös kontekstuaaliseksi sekä leksikaaliseksi, eli sanastolliseksi parametriksi. (Brants, 2000)

4.2.1 Parametrien estimointi

Yksinkertaisimmillaan parametria q voidaan estimoida laskemalla

$$q(t_i|t_{i-2},t_{i-1}) = \frac{f(t_{i-2},t_{i-1},t_i)}{f(t_{i-2},t_{i-1})} \tag{4.10}$$

missä funktio f merkitsee annetun n-grammin lukumäärää harjoitusaineistossa. Datan harvuuden vuoksi tällainen estimaatti ei ole käyttökelpoinen: laajassakaan harjoitusaineistossa ei ole tarpeeksi montaa esiintymää jokaisesta eri trigrammista. 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 sarjan $t_1 \ldots t_n$ todennäköisyys on 0. Luotettavampi tapa estimoida parametria q on hyödyntää trigrammien lisäksi myös harjoitusaineistosta johdettujen uni- ja bigrammien suhteellisia frekvenssejä:

$$Unigrammi : P(t_i) = \frac{f(t_i)}{M} \tag{4.11}$$

$$Bigrammi : P(t_i|t_{i-1}) = \frac{f(t_{i-1},t_i)}{f(t_{i-1})} \tag{4.12}$$

$$Trigrammi : P(t_i|t_{i-2},t_{i-1}) = \frac{f(t_{i-2},t_{i-1},t_i)}{f(t_{i-2},t_{i-1})} \tag{4.13}$$

missä M merkitsee harjoitusaineiston sanojen kokonaislukumäärää. Nyt parametrin q arvoa voidaan estimoida interpoloimalla edellä mainittuja n-grammeja:

$$q(t_i|t_{i-2},t_{i-1}) = \lambda_1 P(t_i) + \lambda_2 P(t_i|t_{i-1}) + \lambda_3 P(t_i|t_{i-2},t_{i-1}) \tag{4.14}$$

missä λ-arvot määräytyvät harjoitusaineiston mukaan siten, että $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Vastaavasti parametrin e arvoa estimoidaan vertaamalla sana-sanaluokka-yhdistelmän frekvenssiä pelkän sanaluokan frekvenssiin:

$$e(w_i|t_i) = \frac{f(w_i,t_i)}{f(t_i)} \tag{4.15}$$

Estimaattien myötä alkuperäinen tunnistusongelma (4.2) on valmis ratkaistavaksi:

$$f(w_1,w_2,\ldots,w_n) = \arg\max_{t_1\ldots t_n} p(w_1,w_2,\ldots,w_n,t_1,t_2,\ldots,t_n) \tag{4.16}$$

$$= \arg\max_{t_1\ldots t_n} \left(\prod_{i=1}^{n+1} q(t_i|t_{i-2},t_{i-1}) \prod_{i=1}^{n} e(w_i|t_i) \right) \tag{4.17}$$

Funktion f arvo voidaan nyt ratkaista tehokkaasti dynaamisella Viterbi-algoritmilla (Viterbi, 1967). (Brants, 2000)

4.2.2 Tuntemattomien sanojen käsittely

Todennäköisyyden e estimaatti (4.15) ei 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 tuntemattomalle 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: esimerkiksi englannin kielen *able*-päätteiset sanat ovat hyvin todennäköisesti adjektiiveja. 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.

Bikel ym. (1999) esittävät 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 Tilastollisen tunnistimen arviointi

Luvussa esitelty tilastollinen tunnistin perustuu vahvasti toisen asteen Markovin ominaisuuteen, missä tilan todennäköisyys riippuu siis vain kahdesta edeltävästä tilasta. Tällainen oletus on kielitieteellisesti melko naiivi: tunnistin tarkastelee kus-

sakin käsiteltävän aineiston kohdassa vain kahta edellistä sanaluokkaa sekä käsiteltävää sanaa. Esimerkiksi seuraavia sanaluokkia tai ympäröiviä sanoja ei huomioida ollenkaan. On selvää, ettei kyseinen rajallinen malli pysty kaappaamaan kaikkia luonnollisen kielen ominaisuuksia, jotka usein pohjautuvat monimutkaisiin ja etäisiin riippuvuussuhteisiin.

Manning ja Schütze (1999, s. 361–362) väittävät, ettei Markovin malliin perustuvaa tunnistinta voida laajentaa huomioimaan mainittuja kontekstuaalisia vihjeitä, koska vaadittavien parametrien määrä tekisi tunnistimesta käyttökelvottoman. Jo trigrammeja käyttäessä joudutaan turvautumaan interpolaatioon, jotta laskettu estimaatti olisi tilastollisesti uskottava: laajemman kontekstin tarkastelu vaikeuttaisi ongelmaa entisestään. Erityisesti ympäröivien sanojen lisääminen tarkasteltavaan kontekstiin on Markovin malliin perustuvalle tunnistimelle ongelmallista. Tunnistimen toteuttama toisen asteen Markovin ominaisuus on siis eräänlainen kompromissi ennustavuuden sekä luotettavuuden välillä.

Väitteestä huolimatta Markovin mallin kontekstualisoinnin suhteen on saavutettu pieniä edistysaskeleita: Banko ja Moore (2004) esittävät kuinka tunnistin voi käyttää hyväkseen ympäröivää kontekstia tarkasteltavan sanan molemmin puolin, mikä johtaa noin 17% vähennykseen tunnistusvirheiden määrässä. Menetelmän toiminta perustuu siihen, että tunnistimen leksikaalinen parametri $e(w_i|t_i)$ (4.9) laajennetaan muotoon $e(w_i|t_{i-1}, t_i, t_{i+1})$. Laajennuksenkin myötä Markovin malliin perustuva tunnistin kuitenkin häviää kontekstin hyödyntämisessä Brillin tunnistimelle sekä hienostuneemmille tilastollisille koneoppimismenetelmille (mm. Toutanova ym., 2003).

Lisäksi esitelty tunnistin on — kaikkine rajoitteineen — kohtuullisen tarkka: Brants (2000) raportoi artikkelissaan 96,7% tunnistustarkkuuden, jossa tuntemattomien sanojen tunnistustarkkuus on 85,5%. Koeolosuhteissa tunnistin ei siis yllä nykyisten huipputunnistimien tarkkuuslukemiin, mutta toisaalta se suoriutuu hyvin todellisissa käyttötapauksissa: Giesbrecht ja Evert (2009) osoittavat, että Markovin malliin perustuva tunnistin on heikosti jäsenneltyä web-tekstiä käsitellessä sekä tarkempi että tehokkaampi kuin kehittyneemmät tilastolliset tunnistimet. Mainittuihin huip-

putunnistimiin verrattuna esitelty tunnistin on lisäksi suhteellisen yksinkertainen, ja edellisen tuloksen perusteella vastustuskykyisempi ylisovittamista vastaan. Tunnistimen yllättävänkin korkean tarkkuuden myötä voidaan spekuloida, ettei harjoitusaineiston kontekstuaalisten vihjeiden merkitys ole leksikaalisten vihjeiden tasolla.

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

Banko, M., & Moore, R. C. 2004. *Part of speech tagging in context*. Proceedings of the 20th international conference on Computational Linguistics, s. 556–561. Association for Computational Linguistics.

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.

Giesbrecht, E., & Evert, S. 2009. *Is Part-of-Speech tagging a solved task? An evaluation of POS taggers for the German Web as Corpus*. Proceedings of the 5th Web as Corpus Workshop (WAC5), s. 27–35.

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

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 proces-*

sing. 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.

Megyesi, B. 1999. *Improving Brill's POS tagger for an agglutinative language*. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, s. 275–284.

Ng, A. Y. & Jordan, M. 2002. *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes*. In NIPS 14.

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.

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.

Ramshaw, L., & Marcus, M. 1994. *Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging*. Proceedings of the ACL Workshop on Combining Symbolic and Statistical Approaches to Language, s. 128–135.

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.

Schneider, G., & Volk, M. 1998. *Adding manual constraints and lexical look-up to a Brill-tagger for German*. Proceedings of the ESSLLI-98 Workshop on Recent Advances in Corpus Annotation, Saarbrücken.

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. 2011. *Semi-supervised condensed nearest neighbor for part-of-speech tagging*.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2, s. 48–52. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. 2003. *Feature-rich part-of-speech tagging with a cyclic dependency network*. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 1, s. 173–180. Association for Computational Linguistics.

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*. IEEE Transactions on Information Theory, 13(2), s. 260–269.

Volk, M. & Schneider, G. 1998. *Comparing a statistical and a rule-based tagger for German*. Computers, Linguistics, and Phonetics between Language and Speech. Proceedings of the 4th Conference on Natural Language Processing - KONVENS-98, s. 125–137.