Kanditutkielma

Tietojenkäsittelytieteen kandiohjelma

Sanaluokkien merkitsemisen ja morfologisen analyysin työkalut eri kielissä

Tuomas Koukkari

16.5.2022

Matemaattis-luonnontieteellinen tiedekunta

Helsingin yliopisto

Yhteystiedot

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

Sähkopostiosoite: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Study programme
Matemaattis-luonnontieteellinen tiedekunta	Tietojenkäsittelytieteen kandiohjelma

Tekijä — Författare — Author

Tuomas Koukkari

Työn nimi — Arbetets titel — Title

Sanaluokkien merkitsemisen ja morfologisen analyysin työkalut eri kielissä

Ohjaajat — Handledare — Supervisors

Laura Ruotsalainen, Titti Malmivirta

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kanditutkielma	16.5.2022	31 sivua, 1 liitesivua

Tiivistelmä — Referat — Abstract

Automaattinen sanaluokkien merkitseminen ja morfologinen analyysi ovat keskeisessä asemassa luonnollisen kielen käsittelyn kentällä. Tässä tutkimuksessa on esitelty ja vertailtu näihin tehtäviin käytettyjä työkaluja ja teknologioita, joihin ne perustuvat; keskeinen tutkimuskysymys on, vaikuttaako käsiteltävän kielen taivutusopillinen monimutkaisuus siihen, millaiset työkalut sen analysointiin parhaiten soveltuvat. Teknologioista on käsitelty erityisesti sääntöpohjaisia malleja sekä neuroverkkoihin perustuvaa koneoppimista. Kielten osalta on vertailtu pääasiassa taivutusopiltaan monimutkaista suomea ja tässä suhteessa verrattaen yksinkertaista englantia, joskin muihinkin kieliin on tarpeen mukaan viitattu.

Tutkimuksen perusteella neuroverkkoihin perustuvat koneoppivat työkalut pääsevät nykyään sääntöpohjaisia malleja parempiin lopputuloksiin riippumatta käsiteltävän kielen taivutusopillisista ominaisuuksista. Tämä pätee kuitenkin vain siinä tapauksessa, että koneoppivan järjestelmän tarpeisiin on käytettävissä riittävästi opetusdataksi kelpaavaa aineistoa. Yleisesti ottaen suurilla ja vahvassa asemassa olevilla kielillä tällaista aineistoa on runsaasti, mutta suurin osa maailman kielistä on erilaisia vähemmistökieliä, joiden kohdalla näin useinkaan ei ole. Näiden kielten käsittelyyn tarvitaankin yhä sääntöpohjaisia malleja.

ACM Computing Classification System (CCS)
Applied computing → Document management and text processing → Document capture → Document analysis
Theory of computation → Formal languages and automata theory → Automata extensions → Transducers
Computing methodologies → Artificial intelligence → Natural language processing → Phonology / morphology

Avainsanat — Nyckelord — Keywords

kieliteknologia, morfologinen jäsentäminen, äärellistilaiset transduktorit, koneoppiminen, neuroverkot

Säilytyspaikka — Förvaringsställe — Where deposited

Helsingin yliopiston kirjasto

Muita tietoja — övriga uppgifter — Additional information

Sisällys

1 Johdanto

Kartoitan tutkielmassani sanaluokkien ja -muotojen automaattiseen merkitsemiseen käytettyjä menetelmiä ja niiden käyttöä erityyppisissä kielissä. Esimerkkinä runsaasti taivutusta sisältävästä kielestä on käytetty ensisijaisesti suomea, taivutusopiltaan yksinkertaisemmasta kielestä puolestaan englantia. Muihin kieliin viitataan tarpeen mukaan.

Aihe liittyy laajempaan luonnollisen kielen käsittelyn (*natural language processing*, lyh. NLP) alaan, missä *luonnollinen kieli* viittaa mihin tahansa ihmisten välisessä kommunikaatiossa käytettyyn kieleen (esim. suomi, englanti, latina, nicaragualainen viittomakieli) erotuksena *formaaleista kielistä* (esim. konekielet). Monet NLP:n käytännön sovelluksista (konekääntäminen, automaattinen oikeinkirjoituksen tarkastus ym.) vaativat pohjakseen sitä, että luonnollista kieltä pystytään analysoimaan koneellisesti, joten sanaluokkien ja taivutuksen analysoimista voidaan pitää eräänlaisena NLP:n "perustasona"(Kann ym., 2020). Lisäksi näiden automaattista merkitsemistä voidaan käyttää kielitieteellisen tutkimuksen apuna.

Menetelmiä on käytössä useita erilaisia. Karkealla tasolla ne voidaan jakaa *sääntöpohjaisiin* malleihin, erilaisiin tilastollisiin mallit sekä koneoppimismalleihin. Näistä tilastolliset mallit on pääosin rajattu tämän tutkimuksen ulkopuolelle yhtäältä siksi, ettei niiden käsittely mahtunut tämänlaajuisen tutkimuksen puitteisiin, ja toisaalta koska niiden kehitys on viime vuosina vähentynyt koneoppimismallien vallatessa alaa (Kłosowski, 2018).

Alustavana hypoteesinani oli, että kielen taivutusopillisilla ominaisuuksilla - eli käytännössä sillä, paljonko siinä esiintyy taivutusta - olisi jotain tekemistä sen kanssa, millaiset menetelmät sen käsittelyyn soveltuvat. Tekemäni kartoituksen perusteella vaikuttaisi kuitenkin siltä, ettei näin ole, vaan useimmiten tärkeimmäksi tekijäksi nousee se, paljonko kullekin kielelle on olemassa valmiiksi käsiteltyä dataa.

Työ jakautuu osiin siten, että luvussa 2 esittelen sanaluokkien ja taivutuksen kielitieteellistä taustaa sekä niiden merkitsemistä yleisellä tasolla, luvuissa 3 ja 4 taas automaattiseen merkitsemiseen käytettyjä teknologioita, eli vastaavasti sääntöpohjaisia malleja ja koneoppimista. Tekemäni kartoituksen tulokset on esitetty luvussa 5. Luku 6 sisältää yhteenvedon tuloksista sekä pohdintaa niiden merkityksestä mahdollisen tulevan tutkimuksen kannalta.

2 Tutkimuskysymys ja teoriatausta

Tässä luvussa käsittelen sanaluokkia ja taivutusta sekä näiden merkitsemistä teoreettisesta näkökulmasta. Koska keskeinen tutkimuskysymykseni koskee sitä, miten hyvin eri POST-työkalut toimivat eri kielissä, muodostuu olennaiseksi myös kysymys siitä, miten erilaisten työkalujen toimintaa voidaan arvioida ja vertailla.

2.1 Sanat ja niiden taivutus

Käsite *sana* on arkikielessä käytettynä jossain määrin moniselitteinen: tilanteesta riippuen sillä saatetaan viitata esimerkiksi kahden välilyönnin erottamaan merkkijonoon tekstissä, yksittäiseen taivutusmuotoon tai sitten ns. sanakirjamuotoon eli sanan *perusmuotoon*. Kielitieteen ja NLP:n näkökulmasta nämä ovat kuitenkin kaikki eri asioita, ja niinpä niihin on viitattava eri termein: täten yläkäsitteen *sana* alla voidaan erottaa käsitteet *sane*, jolla viitataan yksittäiseen sanaan tekstissä riippumatta sen muodosta, *lekseemi*, jolla tarkoitetaan tiettyä sanaa ja kaikkia sen taivutusmuotoja erotukseksi toisista (merkitykseltään eroavista) sanoista, sekä *lemma*, jota käytetään tiettyyn lekseemiin viittaavana nimenä (käytännössä tyypillisesti sanan perusmuoto). Niinpä esim. suomen kielessä muodot *talo*, *talon*, *talot*, *taloissa* jne. edustavat kaikki samaa lekseemiä *talo* (missä *talo* on lekseemin lemma), joka eroaa esimerkiksi lekseemeistä *kirja*, *viisi* tai *syödä* (joihin vastaavasti kuuluu useita muotoja, kuten *kirjoja* tai *söivät*).

Sanojen taivutusta tutkii kielitieteen osa-alueista *morfologia* eli *muoto-oppi*. Lisäksi morfologiaan lasketaan kuuluvaksi ns. *johto-oppi*, joka tutkii johdosten muodostamista kantasanoista (esim. suomen *talo* → *talous* tai *johtaa* → *johdos*); POST-työkalut kuitenkin yleensä käsittelevät johdokset itsenäisinä lekseemeinä pyrkimättä analysoimaan niiden johtosuhteita, joten johto-oppia ei ole tässä tutkimuksessa tarkemmin käsitelty.

Suomessa ja englannissa (sekä useimmissa muissa eurooppalaisissa kielissä) sanojen taivutus perustuu pääasiassa sanoihin lisättäviin *päätteisiin* eli *suffikseihin* (*suffix*). Esimerkiksi suomessa sanan *talo* monikko saadaan lisäämällä pääte *-t* (*talot*), genetiivi päätteellä *-n* (*talon*) jne.; vastaavasti englanniksi 'talo' on yksikössä *house* ja monikossa *houses* (pääte *-s*). Taivutus voi kuitenkin perustua myös esimerkiksi vartalon eteen liitettäviin *etuliitteisiin* eli *prefikseihin* (*prefix*) (esim. swahilin *jina* 'nimi' : *majina* 'nimet') tai vartalonsisäisiin

muutoksiin (esim. englannin *foot* 'jalka' : *feet* 'jalat').

2.2 Kielten morfologisista eroista

Kielet voidaan asettaa jatkumolle *synteettisistä analyyttisiin* sen mukaan, miten paljon taivutusta niissä esiintyy (Sapir, 1921): kieli on sitä synteettisempi, mitä runsaammin sanat siinä taipuvat, ja kääntäen sitä analyyttisempi, mitä vähemmän taivutusta esiintyy. Erittäin analyyttisiä kieliä, joissa taivutusta esiintyy vähän tai ei lainkaan (esim. mandariinikiina) voidaan kutsua myös *isoloiviksi* kieliksi, kun taas synteettiset kielet voidaan jakaa edelleen *agglutinoiviin* kieliin sekä *flekteeraaviin* eli *fuusioiviin* eli *fleksiokieliin* (Karlsson, 2008).

Agglutinoivissa kielissä yhdessä sanassa voi olla erotettavissa useampi erillistä merkitystä kantava yksikkö eli *morfeemi*: esim. suomen sanassa *taloissaniko*, jossa esiintyy sanan vartalo *talo*, monikon tunnus *-i-*, sijapääte *-ssa-*, omistusliite *-ni-* sekä kysymystä ilmaiseva *-ko*. Flekteeraavissa kielissä taas eri merkityksiä kantavia yksiköitä on vähän ja/tai ne eivät ole erotettavissa toisistaan: esim. latinan sanassa *noctium* 'öiden' pääte *-ium* kantaa sekä monikon että genetiivin merkitystä (vastaava. yksikön muoto on *noctis* 'yön', kun taas perusmuoto eli yksikön nominatiivi 'yö' on *nox* ja monikon nominatiivi 'yöt' on *noctēs*). Vastaavasti englannin sanassa *feet* ei ole erikseen erotettavaa monikon tunnusta, vaan sanan merkityksen ('jalka') ja monikollisuuden ilmaus ovat sulautuneet yhteen. Englannissa sanoilla myös on varsin vähän taivutusta; siinä missä esim. suomessa nominit taipuvat kahdessa luvussa ja (laskutavasta riippuen) n. 15 sijamuodossa, minkä lisäksi niihin voidaan liittää omistusliitteitä ja erinäisiä muita päätteitä, englannissa substantiiveilla on analyysista riippuen vain kaksi tai neljä muotoa (yksikkö ja monikko sekä joissain analyyseissa molemmista erikseen perusmuoto ja genetiivi*).

Jako isoloiviin, agglutinoiviin ja flekteeraaviin kieliin on siinä mielessä idealisoitu, että todelliset kielet harvoin jos koskaan edustavat puhtaasti mitään tiettyä morfologista tyyppiä, ts. (lähes) jokaisessa kielessä esiintyy useamman kuin yhden tyypin piirteitä (Karlsson, 2008). Voidaan kuitenkin sanoa, että suomi on varsin tyypillinen esimerkki melko synteettisestä ja pääosin agglutinoivasta kielestä, kun taas englanti on huomattavasti ana-

*Englannin ns. genetiivin tunnus *-'s* voidaan tulkita myös eräänlaiseksi adpositioksi, joka vain äännettäessä (ja kirjoitettaessa) liittyy edelliseen sanaan; tähän viittaavat esimerkiksi sellaiset ilmaukset kuin *the boy opposite me's sister* 'minua vastapäisen pojan sisko', jossa tunnus liittyy monisanaisen ilmauksen viimeiseen sanaan, vaikka merkitykseltään se viittaa ilmauksen pääsanaan (tässä *boy* 'poika') (Lowe, 2016).

lyyttisempi (mutta kuitenkin puolestaan synteettisempi kuin vaikkapa kiina) ja flekteeraavampi.

2.3 Sanaluokat

Iso suomen kielioppi (Hakulinen ym., 2004) määrittelee sanaluokat seuraavasti (§436):

> "SANALUOKALLA tarkoitetaan sanaston alaryhmää, jonka jäsenet käyttäytyvät olennaisilta osin keskenään samalla tavalla. Yhtäläisyydet ovat morfologisia ja syntaktisia, ja niiden pohjana on lisäksi ainakin jonkinasteinen semanttinen samankaltaisuus."

Suomen kielen sanat on tämän (mp.) mukaan jaettu kolmeen pääryhmään eli *nomineihin*, *verbeihin* sekä taipumattomien ja vajaasti taipuvien sanojen ryhmään, jonka muodostavat *adpositiot, adverbit* ja *partikkelit*. Lisäksi nominien ryhmä jakautuu usein erillisinä sanaluokkina pidettyihin alaryhmiin, eli *substantiiveihin, adjektiiveihin, numeraaleihin* ja *pronomineihin*.

Yllä esitetyn määritelmän *semanttinen samankaltaisuus* viittaa siihen, että usein saman sanaluokan muistuttavat toisiaan myös merkitykseltään: verbit kuvaavat karkeasti ottaen tekoja, toimintaa ja tapahtumista, substantiivit esineiden ja asioiden nimiä jne. Toisaalta esimerkiksi verbit *juosta, nähdä* ja *sijaita* ovat lopulta varsin erilaisia merkityksiltään: ensimmäinen kuvaa aktiivista toimintaa, toinen passiivista informaation vastaanottamista ja kolmas silkkaa olemista jossain (missä "toimijana"voi olla myös eloton esine tai vaikkapa kaupunki tai vuori).

Niinpä keskeisin sanaluokkajaon kriteeri onkin *taivutus* (Hakulinen ym., 2004; ks. §436). Suomen kielessä nominit taipuvat luvussa (yksikkö ja monikko: esim. *talo : talot*) sekä sijamuodoissa (*talo : talon : talossa* jne.); verbit taas mm. persoona- ja aikamuodoissa (esim. yksikön ensimmäisen persoonan preesens *juoksen, näen, sijaitsen* vs. monikon kolmannen persoonan pluskvamperfekti *olivat juosseet, olivat nähneet, olivat sijainneet*).

Sanaluokkajako on jossain määrin kielikohtainen: esimerkiksi englannin kielessä on tapana erottaa erillinen *determinatiivien* luokka (*determinatives* t. *determiners*), johon kuuluvat artikkelit (*a, the*), possessiivit (*my, your, their* ym.), demonstratiivit (*this, that, these, those*) sekä tietyt muut sanat, joille joko ei ole suoria suomenkielisiä vastineita tai joiden suomenkieliset vastineet on useimmiten tapana luokitella pronomineiksi. Englannin kielessä ei myöskään ole tapana luokitella sanaluokkia ylemmän tason pääryhmiin, eli nomineja

tai taipumattomien sanojen ryhmää ei sellaisinaan eroteta, vaan substantiivit, pronominit, adjektiivit, adverbit ym. katsotaan itsenäisiksi sanaluokiksi siinä missä verbitkin.

Suomen ja englannin sekä useimpien muiden eurooppalaiset kielten sanaluokkajaot sekä niiden perusteet kuitenkin muistuttavat toisiaan pääosin varsin paljon. Maailmassa on myös kieliä, joissa ei välttämättä voida erottaa muita sanaluokkia kuin nominit ja verbit - ja tämäkin on joissain tapauksissa kyseenalaistettu (Rijkhoff, 2001). Lisäksi esimerkiksi sellaisissa kielissä, joissa sanat eivät taivu (esim. mandariinikiina) sanaluokkajakoa ei luonnollisestikaan voida perustella taivutuksella, vaikka sellainen olisikin muilla perustein tehtävissä.

2.4 Sanaluokkien ja taivutusmuotojen merkitseminen

Sanaluokkien merkitseminen (engl. *part of speech tagging*, POST) tarkoittaa kirjaimellisesti ottaen sitä, että käsiteltävään tekstiin merkitään eli *annotoidaan* (*label* t. *tag*) kunkin sanan kohdalle sen sanaluokka (esim. verbi, substantiivi, adjektiivi tmv.). Käytännössä kuitenkin erityisesti englanninkielisessä kirjallisuudessa näyttäisi usein viitatun termillä POST myös sellaisiin työkaluihin, jotka merkisevät sanaluokan lisäksi myös muuta informaatiota kustakin sanasta, esim. taivutusmuodon. Englannin kielessä, jossa taivutusmuotoja on suhteellisen vähän, niiden merkitseminen ei ole merkittävästi monimutkaisempi tehtävä kuin sanaluokkien, mutta esim. suomen kielessä ero on varsin huomattava. Taivutusmuotojen merkitseminen (*morphological tagging*) eli *automaattinen morfologinen analyysi* (*automatic morphological analysis*) voidaankin tarvittaessa erottaa omaksi kategoriakseen; suomeksi tästä voidaan käyttää myös termiä *morfologinen jäsennys* (vrt. Pirinen, 2008). Käytännössä kuitenkin useimmat nykyiset työkalut pystyvät merkitsemään sekä sanaluokat että taivutusmuodot, joten tässä mielessä termien erottaminen on usein tarpeetonta. Tässä työssä onkin yleisesti käytetty lyhennettä POST-työkalut tarkoitettaessa myös taivutusmuotojen merkitsemiseen kykeneviä työkaluja.

Tyypillisesti nykyisten POST-työkalujen toiminta koostuu useista vaiheista (Voutilainen, 2003). Ensimmäiseksi tapahtuu *saneistus* (*tokenization*), jossa juoksevasta tekstistä erotetaan kukin sane omaksi yksikökseen. Useimmiten myös välimerkit tulkitaan eräänlaisiksi saneiksi ja merkitään erikseen. Seuraavaksi haetaan kullekin saneelle sitä vastaava analyysi tai analyysit; tähän vaiheeseen liittyy yleensä myös *lemmatisointi* (*lemmatization*), eli sanaluokan ja mahdollisen taivutusmuodon lisäksi sanan kohdalle merkitään sen lekseemin lemma, johon kyseinen sane kuuluu.

Useissa kielissä jotkin muodot voivat olla moniselitteisiä (*ambiguous*), ts. sanan oikea kieliopillinen analyysi ei käy ilmi pelkästään sanasta itsestään, vaan se on tulkittava asiayhteyden perusteella. Esimerkiksi englannissa on varsin tyypillistä, että sama muoto voidaan tulkita asiayhteydestä riippuen joko substantiiviksi tai verbiksi; täten esim. muoto *jumps* voi olla joko substantiivin *jump* 'hyppy' monikkomuoto tai verbin *to jump* 'hypätä' preesensin yksikön kolmas persoona. Tällaisissa tapauksissa voidaan joko listata kaikki mahdolliset tulkinnat tai sitten pyrkiä *disambiguoimaan* muoto, eli valitsemaan mahdollisista analyyseista asiayhteyden perusteella oikea. Automaattisissa POST-työkaluissa tämäkin tapahtuu tyypillisesti vaiheittain: työkalu hakee saneelle ensin kaikki sen mahdolliset analyysit, ja pyrkii tämän jälkeen karsimaan niistä pois sellaiset, jotka eivät kontekstin perusteella sovi (Voutilainen, 2003).

2.5 Annotaatioskeemat

Erilaiset POST-työkalut eroavat toisistaan paitsi käytetyn teknologian (ks. seuraava luku) suhteen, myös ns. *annotaatioskeeman* (*tagging scheme*, *tagset*, *annotation scheme*) osalta, eli sen, mitkä kaikki sanaluokat ja taivutusmuodot erotetaan ja miten (eli yleensä millä lyhenteellä) mikäkin niistä merkitään (Cloeren, 1999). Esimerkiksi englannin kielessä perinteisesti käytetyissä *Brownin korpuksen* annotaatioskeemassa ja siihen pohjautuvissa skeemoissa (esim. ns. *Pennin puupankin* skeema) yksikölliset substantiivit annotoidaan lyhenteellä NN, monikolliset substantiivit lyhenteellä NNS, verbien perusmuoto VB, mennyt aika VBD, jne. (mt.; Taylor ym., 2003).

Englantia varten kehitetyt annotaatioskeemat sellaisinaan sovi käytettäväksi muiden kielten annotointiin, mistä syystä viime vuosina on pyritty kehittämään yhtenäistettyjä kielestä riippumattomia annotaatioskeemoja. Jalansijaa on saanut erityisesti *Universal Dependencies*, (de Marneffe ym., 2015) (UD), johon sisältyy myös esim. suomen kielelle soveltuva annotaatioskeema (Pyysalo ym., 2015).

UD:n kanssa kilpaillut UniMorph-projekti (Kirov ym., 2018) ei ilmeisesti ole enää aktiivinen (uusia päivityksiä ei projektin verkkosivuilla* näyttäisi olevan enää parilta viime vuodelta), mutta joitakin sen periaatteita on sovellettu UD:n uudemman eli toisen version suunnittelussa†. Lisäksi UniMorph mahdollistaa kompaktimmat annotaatiot (ja on

*https://unimorph.github.io/

†https://universaldependencies.org/v2/features.html

suoraan yhteensopiva kielitieteessä yleisesti käytettyjen ns. *Leipzigin glossaussääntöjen*[*] kanssa), mistä syystä tässä tutkielmassa esimerkkiannotaatiot on tehty UniMorphin annotaatioskeemalla. Käytetyt annotaatiot on listattu liitteessä A.

2.6 Työkalujen suorituskyvyn arviointi

POST-työkalujen suorituskyvyn arvioimiseen on käytössä joukko mitattavia suureita, joista keskeisimmät ovat *ulkoinen tarkkuus* (*accuracy*), *sisäinen tarkkuus* (*precision*) sekä *herkkyys* (*recall*); näitä voidaan merkitä vastaavasti A, P ja R. Ulkoinen tarkkuus tarkoittaa yksinkertaisesti oikein annotoitujen sanojen tai muotojen määrää n_{OK} suhteessa kaikkien annotoitujen sanojen tai muotojen määrään n_w, eli $A = \frac{n_{OK}}{n_w}$ (Paroubek, 2007). Sisäisen tarkkuuden ja herkkyyden mittauksia tarvitaan tapauksissa, joissa työkalu voi antaa sanalle tai muodolle enemmän kuin yhden annotaation (mt.); mikäli kuitenkin vain yksi annotaatio on oikein (kuten yleensä on), herkkyys voidaan tällöin määritellä oikean annotaation sisältävien annotaatiojoukkojen määräksi m_{OK} suhteessa annotaatiojoukkojen kokonaismäärään m_{tot} ja sisäinen tarkkuus herkkyyden suhteeksi annotaatioiden määrän keskiarvoon per sana (mt.): $R = \frac{m_{OK}}{m_{tot}}$ ja $P = \frac{R}{n_{tot}/n_w}$.

Merkittävä tekijä menetelmän toimivuuden arvioimisessa on myös sen suoritusnopeus: täydellisenkin tuloksen tuottava algoritmi on jokseenkin hyödytön, jos sen suorittamiseen kuluu moninkertaisesti maailmankaikkeuden nykyistä ikää vastaava aika (vrt. Samuel, 1959). Lisäksi voidaan arvioida työkalun ominaisuuksien määrää ja soveltuvuutta erilaisiin tehtäviin; nykyiset työkalut saattavat sisältää (tai niihin voidaan yhdistää kolmansien osapuolien tarjoamia) monenlaisia eri tehtäviin erikoistuneita *moduuleja* oikeinkirjoituksen tarkistimesta esim. runousgeneraattoriin (ks. esim. Hämäläinen ja Alnajjar, 2019). Olen kuitenkin keskittynyt tässä tutkimuksessa ydinominaisuuksiin, eli sanaluokkien ja taivutusmuotojen merkitsemiseen.

[*] https://www.eva.mpg.de/lingua/resources/glossing-rules.php

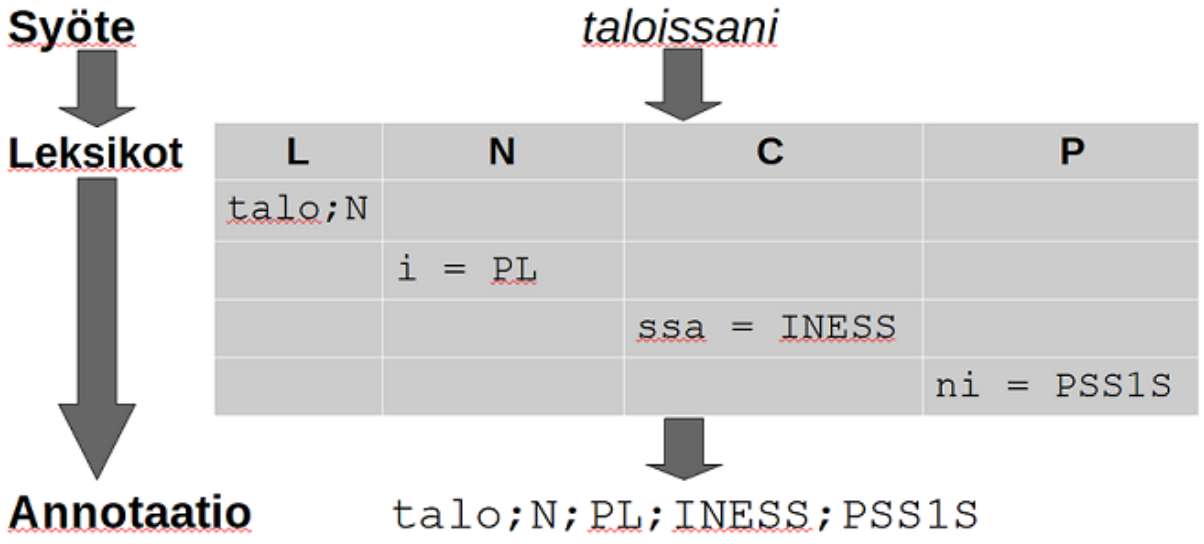
3 Sääntöpohjaiset mallit

Tässä luvussa esittelen sääntöpohjaisia POST-työkaluja ja niiden toimintaperiaatteita. Luku alkaa yleisen tason periaatteista ja etenee sitten tarkemmin annotoinnin äärellistilaisiin menetelmiin ja disambiguoinnin suorittaviin rajoitesääntöihin. Luvun loppupuolella esittelen lyhyesti yksittäisiä suomen ja englannin analysointiin käytettyjä työkaluja.

3.1 Yleistä

Sääntöpohjaiset mallit perustuvat nimensä mukaisesti malliin valmiiksi koodattuihin sääntöihin. Säännöt perustuvat käsiteltävän kielen kielioppiin, ja koska jokaisella kielellä on erilainen kielioppi, ovat sääntöpohjaiset mallit täten pakostakin kielikohtaisia.

Tyypillisesti sääntöpohjainen POST-työkalu hakee sanan lemman ja sanaluokan erillisessä tiedostossa tai tietokannassa sijaitsevasta *leksikosta* (*lexicon*) eli eräänlaisesta sanakirjasta, johon ne on merkitty (Karlsson, 1995). Lisäksi päätteille (tai etuliitteille tms.) on omat leksikkonsa, joita voi olla yksi tai useampi (mt.). Näin ollen työkalu pyrkii yhdistämään eli *konkatenoimaan* (*concatenate*) päätteitä koskevan informaation sanan lemmaa koskevaan informaatioon kunkin annotaation luomiseksi.



Kuva 3.1: Annotaatioiden haku leksikoista

Kuvassa 3.1 on kaavamaisesti esitetty yksi mahdollinen tapa, jolla sääntöpohjainen malli voisi luoda annotaation saneelle *taloissani*: ensimmäiseksi haetaan lemman ja sanaluokan merkintä leksikosta L ja tämän jälkeen päätteitä vastaavat merkinnät pääteleksikoista, joita tässä tapauksessa on kolme (N = luku, C = sija, P = omistusliitteet).

Disambiguointiin on omat sääntönsä, joissa hyödynnetään ympäröivien saneiden tarjoamaa informaatiota. Tietyissä malleissa (esim. rajoitekielioppi; ks. alla kohta 3.3) todennäköisyyslaskentaa voidaan kuitenkin hyödyntää tilanteissa, joissa säännöt osoittautuvat riittämättömiksi (Karlsson, 1995).

3.2 Äärellistilaiset morfologiset jäsentimet

Ensimmäiset sääntöpohjaiset mallit 1950-luvulla perustuivat puhtaasti äärellisiin automaatteihin (Voutilainen, 2003). Sittemmin erityisesti disambiguoinnin avuksi on otettu esim. em. rajoitesäännöt ja todennäköisyyslaskenta (ks. alla kohta 3.3) (mt.), mutta usein varsinaisesta annotaatioiden luomisesta vastaa edelleen *äärellistilainen transduktori* (*finite-state transducer*) eli äärellinen automaatti, joka ottaa syötteenä vastaan merkkijonon ja muuttaa tiettyjen sääntöjen mukaan sen toiseksi (Pirinen, 2008).

Äärellistilainen morfologinen jäsennin on äärellistilainen transduktori, joka pystyy ottamaan syötteenä vastaan taivutetun muodon sanasta ja tuottamaan näiden perusteella sanan lemman sekä tiedon siitä, mistä muodosta on kyse - tai päin vastoin, sillä transduktori toimii luonnostaan kahteen suuntaan (Pirinen, 2008). Täten esim. suomen kielen muoto *talon* voitaisiin muuntaa muotoon `talo;N;SG;GEN`, ja kääntäen sama transduktori pystyisi annetun tiedon `talo;N;SG;GEN` perusteella tuottamaan muodon *talon* (vrt. mt.); jälkimmäinen suunta tosin ei ole sanaluokkien ja -muotojen merkitsemisen kannalta tarpeen eikä näin ollen tämän tutkimuksen kannalta olennainen.

3.3 Rajoitekielioppi ja rajoitesäännöt

Rajoitekielioppi (*Constraint Grammar*, CG) on formalismi rajoitesääntöihin perustuvien työkalujen luomiseksi eri kielille (Karlsson, 1995). Rajoitekielioppiin ja sen toiseen versioon (CG2) perustuvia malleja kehitettiin erityisesti 1990-luvulla useille kielille, ml. suomelle ja englannille (mt.; Bick ja Didriksen, 2015). Uusin versio on 2010-luvulla kehitetty CG3 (Bick ja Didriksen, 2015).

Rajoitesäännöt (*constraint rules*, *pattern-action rules*) ovat sääntöjä, jotka pyrkivät karsimaan kontekstiin sopimattomat annotaatiot pois kultakin saneelta (Karlsson, 1995; Voutilainen, 2003). Käytännössä siis työkalun morfologinen jäsennin hakee ensin kullekin saneelle kaikki mahdolliset annotaatiot, minkä jälkeen rajoitesäännöillä karsitaan annotaatioita, kunnes jäljellä on vain yksi mahdollinen annotaatio tai kunnes kontekstin ehtoihin soveltuvia sääntöjä ei enää ole käytettävissä (Karlsson, 1995).

Seuraavassa on kuvattu sääntöpohjaisen työkalun toimintaa mukaillen vapaasti Voutilaisen (Voutilainen, 2003) käyttämää englanninkielistä esimerkkilausetta *"The table collapsed ('pöytä romahti')."* (suomennos minun). Sääntöpohjainen työkalu listaisi ensin kullekin sanalle kaikki mahdolliset analyysit:

- *the*:
  - `the;DET;DEF`
- *table*:
  - `table;N;SG`: substantiivi ('pöytä'; 'taulukko'), yksikkö; tai
  - `table;V;FIN`: verbi ('pöydätä'; 'taulukoida'), finiittinen, useita mahdollisia muotoja
- *collapsed*:
  - `table;V;FIN;PST`: verbi ('romahtaa'), finiittinen, mennyt aika; tai
  - `table;V.PTCP;NFIN;PRF`: verbi ('romahtaa'), infiniittinen, partisiipin perfekti
- *.*:
  - `.;PUNCT;EOS`: välimerkki, lause päättyy

Tässä *finiittinen* (*finite*, `FIN`) viittaa persoonan ja/tai aikamuodon ilmaisevaan verbimuotoon ja *infiniittinen* (*non-finite*, `NFIN`) muotoon, joka yksinään ei näitä ilmaise; nimestään huolimatta partisiipin perfekti ilmaisee englannissa perfektiä (tai pluskvamperfektiä) vain yhdessä apuverbin *have/has* (tai vastaavasti *had*) kanssa. Tämä on olennaista sikäli, että lähtökohtaisesti lauseessa on aina oltava yksi finiittimuotoinen verbi.

Kun mahdolliset annotaatiot on haettu, seuraa disambiguointi:

- *the*:
  - Vain yksi mahdollinen annotaatio, ei tarvitse disambiguoida.

- *table*:
  - Hyödynnetään tietoa siitä, että englannin kielessä finiittimuotoinen verbi ei lähtökohtaisesti koskaan välittömästi seuraa artikkelia; ts. sovelletaan sääntöä muotoa *jos edellinen sane on artikkeli, hylkää kaikki annotaatiot, joissa sanaluokkana on finiittimuotoinen verbi*. Pseudokoodilla esimerkiksi:

```
if Sane[i-1] == "the":
    for each Annotaatio in Annotaatiot:
        if Annotaatio.contains("FIN"):
            reject Annotaatio
```

  - Jäljelle jää ainoastaan yksi mahdollinen annotaatio: `table;N;SG`.

- *collapsed*:
  - Hyödynnetään tietoa siitä, että lauseessa on oltava aina yksi finiittimuotoinen verbi; ts. sovelletaan sääntöä muotoa *mikäli lauseessa ei ole muita saneita, joille analyysi finiittimuotoisena verbinä on mahdollinen, valitse analyysiksi finiittimuotoinen verbi.*

```
lauseessaFinV = False

for each Sane in Lause:
    for each Annotaatio in Sane.getAnnotaatiot:
        if Annotaatio.contains("FIN"):
            lauseessaFinV = True

if not lauseessaFinV:
    for each Annotaatio in Annotaatiot:
        if Annotaatio.contains("FIN"):
            select Annotaatio
```

- *.*:
  - Vain yksi mahdollinen annotaatio.

3.4 Sääntöpohjaiset mallit suomen kielessä

Käytetyin suomen kielen sääntöpohjainen työkalu on alun perin Tommi Pirisen pro gradu -tutkimustyönä (Pirinen, 2008) syntynyt Omorfi (Hämäläinen ja Alnajjar, 2015). Se perustuu avoimeen lähdekoodiin ja on sittemmin kerännyt aktiivisen kehittäjäyhteisön,

joka on laajentanut ja kehittänyt ohjelmaa huomattavasti (Pirinen, 2015). Omorfin ytimen muodostaa äärellistilainen morfologinen jäsennin (Pirinen, 2008); rajoitekielioppiin (CG3) perustuva disambiguointimoduuli* on lisätty vuonna 2015. Lisäksi Omorfi voidaan yhdistää esim. Tromssan yliopiston Giellatekno-keskuksen† tarjoamiin rajoitekielioppiin perustuviin työkaluihin (Hämäläinen ja Alnajjar, 2015).

3.5 Sääntöpohjaiset mallit englannin kielessä

Neuroverkkoihin perustuvia koneoppivia malleja on viime vuosina kehitetty erityisesti englannin kielelle, ja muuntyyppiset mallit vaikuttaisivat jääneen vähemmälle huomiolle (Pirinen, 2019). Yhä käytössä on silti ainakin rajoitekielioppiin perustuva EngGram (Bick, 2007; Bick, 2020). Varsinkin aiemmin on ollut laajalti käytössä myös Atro Voutilaisen kehittämä EngCG (Bick ja Didriksen, 2015). EngCG on niin ikään rajoitekielioppiin perustuva reduktionistinen työkalu, joka koostuu kolmesta moduulista: lemmatisoijasta, morfologisesta analysaattorista sekä disambiguoijasta (Voutilainen, 1997).

Lisäksi usein (ks. esim. Horsmann ym., 2015; Sadredini ym., 2018) esimerkkeinä sääntöpohjaisesta mallista mainitaan *Brillin malli* ja siihen perustuvat mallit. Brillin malli ei kuitenkaan ole siinä mielessä puhtaasti sääntöpohjainen malli, että se hyödyntäisi valmiiksi kovakoodattuja sääntöjä. Sen sijaan se luo itse omat annotointisääntönsä koneoppimista hyödyntäen (Hepple, 2000; Brill, 1992).

*http://frankier.github.io/omorfi/man/omorfi-disambiguate-text(1).html
†https://giellatekno.uit.no/

4 Koneoppiminen

Tässä luvussa käsittelen koneoppimista ja sen käyttöä POST-työkaluissa. Koneoppiminen on aihepiirinä sen verran laaja, ettei sen perusteellinen käsittely mahdu tämän tutkielman puitteisiin; luvussa käydään aluksi hyvin yleisellä tasolla läpi sen periaatteita ja tämän jälkeen hieman tarkemmin erilaisia neuroverkkoihin perustuvia syväoppimismenetelmiä. Luvun lopussa esittelen lyhyesti joitakin koneoppimiseen perustuvia POST-työkaluja.

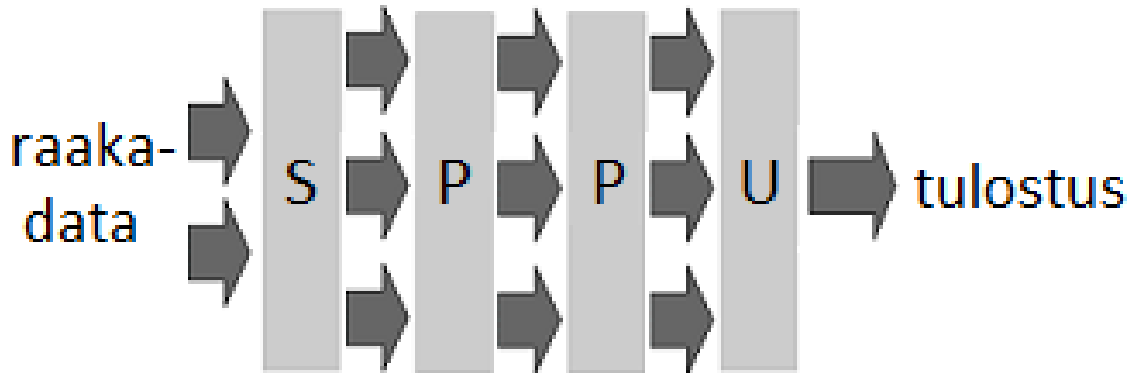
4.1 Yleistä

Koneoppimisessa on pohjimmiltaan kyse siitä, että kone pystyy itsenäisesti kehittämään jonkin tehtävän suorittamiseksi ratkaisuja, joita siihen ei ole alun perin valmiiksi ohjelmoitu, ja parantamaan niitä (Samuel, 1959). Jung (2022: 28) esittää karkeana määritelmänä, että koneoppimivat järjestelmät yleisesti yrittävät oppia ennustamaan, miten jokin datapiste tulisi annotoida (*label*), vain kyseisen datapisteen omien ominaisuuksien perusteella. Nähdäkseni tämä määritelmä pätee nimenomaan sanaluokkien merkitsemiseen erittäin hyvin (vrt. kohdat 2.4 ja 2.5 yllä); tässä tapauksessa siis datapisteiksi katsotaan yksittäiset sanamuodot ja annotaatioiksi niiden kieliopillinen analyysi ja sen kirjaaminen.

Koneoppimismenetelmät voidaan jakaa *ohjattuun*, *ohjaamattomaan* ja *vahvistusoppimiseen* (Jung, 2022). Lisäksi ns. *klassiset koneoppimismallit* voidaan erottaa *syväoppimisesta* (Chiche ja Yitagesu, 2022).

Ohjatussa oppimisessa koneelle annetaan ensin *opetusdataa* (esim. valmiiksi sanaluokkamerkittyä tekstiä), jonka perusteella se pyrkii päättelemään, millaisia sääntöjä sen tulee noudattaa (Jung, 2022). Tämän jälkeen sen annetaan itsenäisesti yrittää uuden datan parissa (esim. uusien tekstien merkitsemistä). Ohjaamattomassa oppimisessa koneella ei ole käsiteltävästä datasta valmiiksi mitään tietoa, vaan se pyrkii täysin itsenäisesti löytämään datasta piirteitä, joiden perusteella sitä voi luokitella (mt.). Vahvistusoppiminen muistuttaa ohjaamatonta oppimista sikäli, että koneen on opittava ennustamaan datasta piirteitä ilman valmiiksi annettuja ennakkotietoja, mutta lisäksi vahvistusoppimisessa sen on pystyttävä ennakoimaan tulevien datapisteiden ominaisuuksia edeltävien perusteella ja sopeutumaan muuttuviin tilanteisiin reaaliajassa (mt.); Jung (mt.) käyttää esimerkkinä vahvistusoppimiseen perustuvasta järjestelmästä itseohjautuvan auton tekoälyä.

Syväoppimisessa kone käyttää hyväkseen useita tasoja (*layers*) alkaen raakadatasta ja päätyen lopulta haluttuun tulkintaan siten, että kukin korkeamman tason representaatio on hieman edellistä abstraktimpi (LeCun ym., 2015). Yksinkertaisen syväoppivan mallin toiminta on esitetty kaavamaisesti kuvaajassa 4.1.: raakadata annetaan syötteenä *syötekerrokselle* (*input layer*) ja lopullisen tulkinnan tulostaa *ulostulokerros* (*output layer*); näiden välissä on tyypillisesti useita *piilokerroksia* (*hidden layers*), joissa datan käsittely pääosin tapahtuu (mt.; Bengio, 2009). Syväoppivat POST-menetelmät ovat vielä varsin uusi tutkimusala, mutta huomattava osa uusimmasta alan tutkimuksesta koskee nimenomaan syväoppivia menetelmiä (Chiche ja Yitagesu, 2022).



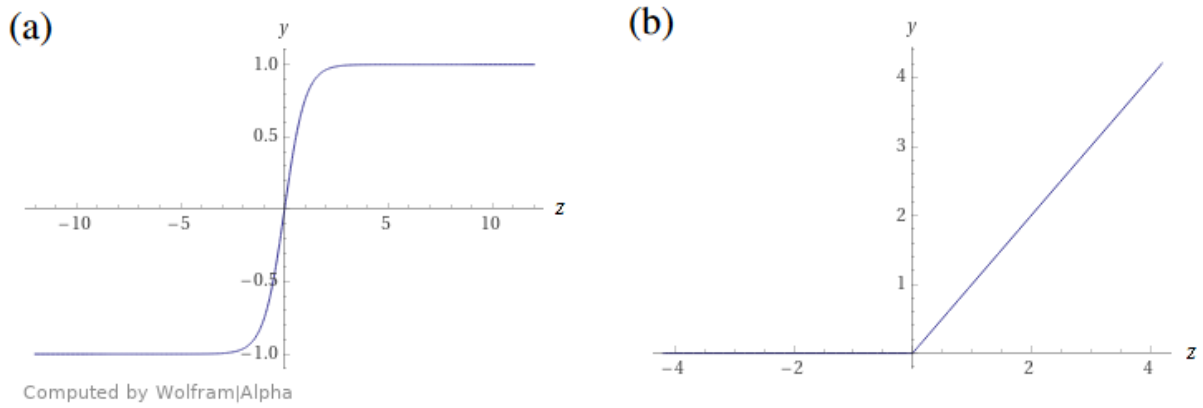
Kuva 4.1: Syväoppivan järjestelmän rakenne (S = sisääntulokerros, P = piilokerros, U = ulostulokerros)

Eräs koneoppimiseen ja erityisesti syväoppimiseen liittyvä ominaisuus on, ettei se, miten kone tarkkaan ottaen tuottaa annetusta datasta luomansa tuloksen tai tulkinnan, ole tyypillisesti edes järjestelmän kehittäjien tiedossa (Rudin ja Radin, 2019). Koneoppivan järjestelmän toiminnan perusperiaatteet ovat tiedossa, mutta annetun tehtävän ratkaisemiseen tarkoitetun mekanismin kone luo näiden periaatteiden pohjalta lopulta itse. Mekanismi voi useinkin olla siinä määrin monimutkainen, ettei sen tulkitseminen olisi ihmiselle edes teoriassa mahdollista; kone voi esimerkiksi ottaa huomioon datasta potentiaalisesti miljardeja erilaisia parametreja (Goodfellow ym., 2016).

4.2 Yksinkertaiset neuroverkot ja niiden toiminta

Syväoppivat järjestelmät perustuvat yleensä *neuroverkkoihin* (Bengio, 2009). Neuroverkko on karkealla tasolla biologista hermostoa (kuten ihmisaivoja) jäljittelevä järjestelmä (Goodfellow ym., 2016), joka koostuu useista toisiinsa kytkeytyvistä *neuroneista* (samaan

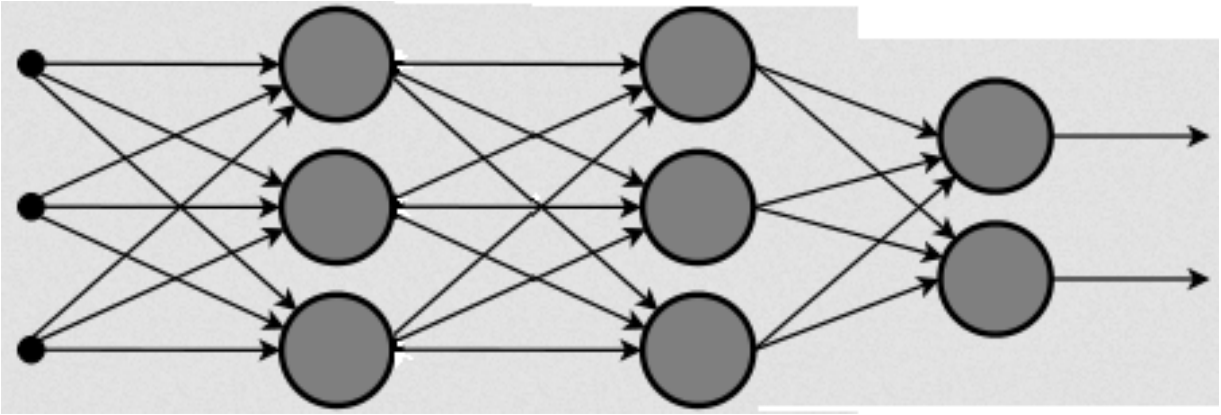
tapaan kuin biologiset hermostot koostuvat biologisista neuroneista eli hermosoluista) (Chiche ja Yitagesu, 2022). Neuronien väliset kytkökset välittävät neuroverkossa signaaleja neuronien välillä siten, että kullakin kytköksellä on tietty painoarvo (Jung, 2022). Neuroni vastaanottaa täten painottuneita signaaleja yhdeltä tai useammalta muulta neuronilta ja tuottaa niiden pohjalta jonkinlaisen *aktivaatiofunktion* avulla uuden signaalin, jonka se välittää edelleen seuraavalle neuronille (mt.). Yksinkertaisimmassa tapauksessa funktio on lineaarinen, jolloin kyseessä on ns. *perseptroni* (*perceptron*) (Jurafsky ja Martin, 2021); tällöin neuronin eteenpäin välittämän signaalin voimakkuus (tai sitä kuvaava muuttujan arvo) y on suoraan suhteessa sen syötteenä saaman signaalin voimakkuuteen (tai arvoon) z. Nykyään käytetään kuitenkin yleensä joko epälineaarisia funktioita (Goldberg, 2016) tai ns. *rektifioijafunktiota* (*rectifier*, *rectified linear unit*, lyh. ReLU), jonka arvo kasvaa lineaarisesti suhteessa syötteeseen tietystä syötteen arvosta alkaen mutta sitä pienemmillä syötteen arvoilla tuottaa arvon 0 (Jurafsky ja Martin, 2021). Epälineaarisista funktioista usein käytetty on *hyperbolinen tangenttifunktio* (*tanh*), jonka arvot kasvavat tietyllä varsin kapealla alueella jyrkästi suhteessa syötteeseen ja tämän alueen ulkopuolella kasautuvat joko lähelle -1:tä (pienillä syötteen arvoilla) tai 1:tä (suurilla) (mt.). ReLU- ja tanh-funktiot sekä niitä vastaavat käyrät on esitetty kuvassa 4.2*.



Kuva 4.2: Tyypillisiä aktivaatiofunktioita. a) $y = \tanh(z)$. b) $y = \mathrm{ReLU}(z) = \max(0, z)$.

Periaatteessa neuronien väliset kytkökset voivat olla mielivaltaisia, mutta käytännössä neuroverkoista muodostetaan yleensä tyypillinen syväoppiva järjestelmä siten, että neuronit on järjestetty tasoiksi, ja kukin neuroni vastaanottaa signaaleja vain edellisen tason neuroneilta ja välittää niitä seuraavalle tasolle (Jung, 2022). Kerrokset ja niiden väliset yhteydet on esitetty yksinkertaistetusti kuvassa 4.3.

*Kuvaajat luotu Wolfram|Alphan tarjoamalla työkalulla: https://www.wolframalpha.com/input?i



Kuva 4.3: Neuroverkon rakenne. Kuva: Offnpoft, Wikimedia Commons (muokattu).

Datan representaatiota neuroverkon kullakin tasolla voidaan tällöin ajatella n-ulotteisena vektorina, missä n on kyseisellä tasolla olevien neuronien määrä (Goldberg, 2016). Mikäli tietyn tason (jota merkittäköön i) jokainen neuroni on yhteydessä jokaiseen seuraavan tason ($i + 1$) neuroniin (ns. *täysin kytketty kerros*, engl. *fully connected layer*), voidaan datan siirtymistä näiden kerrosten välillä kuvata lineaarisena transformaationa d_i-ulotteisesta avaruudesta d_{i+1}-ulotteiseen avaruuteen (missä d_i on kerroksen i neuronien määrä ja d_{i+1} vastaavasti kerroksen $i + 1$) (mt.). Esimerkiksi kuvassa 4.3 kahden oikeanpuolimmaisen kerroksen välissä tapahtuisi transformaatio kolmesta ulottuvuudesta kahteen. Tärkeitä erikoistapauksia ovat syötekerroksen ulottuvuuksien määrä d_{in} sekä ulostulokerroksen ulottuvuuksien määrä d_{out} (mt.).

Yksinkertaisimmillaan neuroverkot ovat *eteenpäin suunnattuja neuroverkkoja* (*feedforward neural network*, FNN tai FFNN), joissa kukin neuroni saa dataa vain suoraan edellisen kerroksen neuroneilta (Otter ym., 2020). Eteenpäin suunnattu neuroverkko voidaan kuvata formaalisti funktiona $h(x) = u$, missä $x \in \mathbb{R}^{d_{in}}$ on syötevektori ja u on ulostulovektori (Goldberg, 2016). Lineaarisen perseptronimalliin perustuvan verkon tapauksessa tarkemmin:

$$h(x) = Wx + b$$

missä $W \in \mathbb{R}^{d_{in} \times d_{out}}$ on verkon painotusten muodostama matriisi (*weight matrix*) ja $b \in d_{out}$ on mahdollinen lopputulokseen lisättävä lisäpainotus (*bias*) (mt.).

Yhden epälineaarisen piilokerroksen sisältävä verkko voitaisiin mallintaa funktiona

$$h(x) = f(W^{(1)}x + b^{(1)})W^{(2)} + b^{(2)}$$

missä f on syötevektoriin alkioittain sovellettu aktivaatiofunktio, $W^{(1)} \in \mathbb{R}^{d_{in} \times d_1}$ on

syötekerroksen ja piilokerroksen välisten yhteyksien painotusten muodostama matriisi, $W^{(1)} \in \mathbb{R}^{d_1} \times d_{out}$ vastaavasti syötekerroksen ja ulostulokerroksen välinen painotusmatriisi, ja $b^{(1)}$ sekä $b^{(2)}$ vastaavasti piilokerroksen ja ulostulokerroksen mahdolliset lisäpainotukset (mt.; Jurafsky ja Martin, 2021).

Vastaavasti kaksi piilokerrosta sisältävälle verkolle (ja yleistäen samalla logiikalla useammallekin kerrokselle) pätee (Goldberg, 2016):

$$h(x) = (f^2(f^1(xW^1 + b^1)W^2 + b^2))W^3$$

Useampien kerrosten toiminta voidaan myös esittää silmukkana (Jurafsky ja Martin, 2021), eli jos kerrosten lukumäärä on n ja syötevektori on `x[0]`:

```
for i in 1 to n
    z[i] = W[i] * x[i-1] + b[i]
    x[i] = g[i](z[i])
u = x[n]
```

NLP:ssä (ja useissa muissa sovelluksissa) neuroverkon varsinaisena tehtävänä on usein arvioida mahdollisten eri tuloksen (esim. POST-työkalussa eri annotaatioiden) todennäköisyyksiä. Tätä varten ulostulovektori on muunnettava todennäköisyysjakaumaksi, mihin käytetään yleensä ns. *softmax*-funktiota σ (Jurafsky ja Martin, 2021). Mille tahansa vektorille z ja ulottuvuuksien määrälle d (mt.):

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{d} e^{z_j}} \quad 1 \leq i \leq d$$

4.3 Takaisinkytketyt neuroverkot ja transformerit

Takaisinkytketyt neuroverkot (*recurrent neural network*, RNN) eroavat FFN:istä siten, että neuronit eivät välitä vain juuri vastaanottamiaan signaaleja, vaan niillä on lisäksi *muisti* (*tilavektori*), jonka avulla ne voivat ottaa huomioon aiemmin käsittelemäänsä dataa (mt.; LeCun ym., 2015). Erityisesti NLP:n alalla hyödyllisiksi ovat osoittautuneet *kaksisuuntaiset takaisinkytketyt neuroverkot* (*bidirectional recurrent neural networks*); koska sanan oikea tulkinta lauseessa voi riippua sekä sitä edeltävistä että sitä seuraavista sanoista, on hyödyllistä käsitellä lause sekä alusta loppuun että lopusta alkuun (Otter ym., 2020).

Monet NLP:n alalla käytetyt neuroverkkomallit perustuvat *enkooderi-dekooderi*-malliin (Otter ym., 2020), joka muistuttaa sääntöpohjaisia transduktoreita siinä mielessä, että se

pyrkii muuntamaan syötteen tulosteeksi yksi elementti kerrallaan (Vaswani ym., 2017). Käytännössä enkooderi ja dekooderi ovat usein kaksi erillistä neuroverkkoa; enkooderi muuntaa ensin alkuperäisen syötteen tietyn pituiseksi vektoriksi, jonka puolestaan dekooderi ottaa vastaan syötteenä ja jonka perusteella se luo lopullisen tulostuksen (Otter ym., 2020).

Takaisinkytkeytyvien neuroverkkojen yhdistäminen enkooderi-dekooderi-malliin ei ole ollut täysin ongelmatonta. Yhtäältä niissä peräkkäisten operaatioiden määrä ja siten aika- ja muistivaatimukset kasvavat lineaarisesti suhteessa syötteen kokoon ($O(n)$) (Vaswani ym., 2017), ja toisaalta vaatimus välittää tietyn pituinen vektori enkooderilta dekooderille ei mahdollista sen arvioimista, ovatko jotkin osat syötteestä tärkeämpiä kuin toiset (Otter ym., 2020). Tämäntapaisten ongelmien ratkaisussa avuksi on aivan viime vuosina otettu *huomiomekanismi* (*attention mechanism*) (mt.) ja siihen pohjautuva *transformerimalli* (*transformer*) (Vaswani ym., 2017).

Huomiomekanismi toimii siten, että joko esim. enkooderi (Bahdanau ym., 2015) tai tietyt piilokerrokset (Vaswani ym., 2017) laskevat jonkinlaisen *huomiofunktion* avulla painotuksen käsittelemälleen datalle ennen sen välittämistä eteenpäin (mt.). Tämä mahdollistaa sen, että järjestelmä voi oppia esimerkiksi lausetta käsitellessään antamaan joillekin sanoille enemmän painoarvoa kuin toisille (Otter ym., 2020).

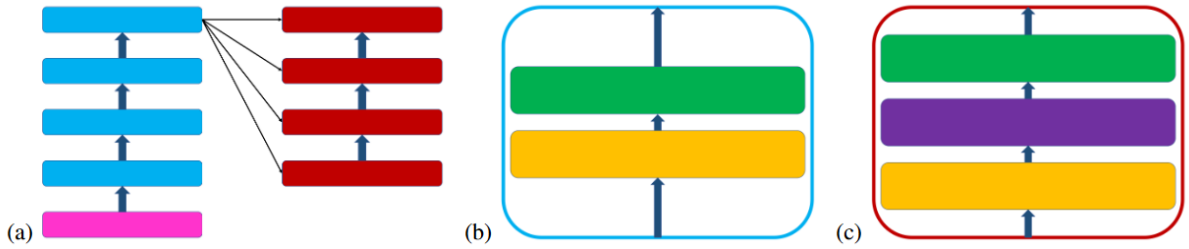
Vaswanin ym. (2017) mukaan huomiofunktio voidaan kuvata joukkona arvoja (*values*) ja niitä vastaavia avaimia (*keys*) sekä erityisiä *kyselyitä* (*queries*) siten, että ulostulovektori u lasketaan vektorina kuvatun arvojoukon V painotettuna keskiarvona ja arvojoukko V puolestaan erityisen *yhteensopivuusfunktion* (*compatibility function*; vrt. myös termi *alignment model*, jota käyttävät oman hieman erilaisen mallinsa kuvauksessa Bahdanau ym., 2015) avulla vastaavasti avainjoukkoa ja kyselyjoukkoa kuvaavista vektoreista K ja Q. Enkooderi-dekooderi-mallissa kyselyjoukko muodostuu dekooderin aiempien tuottamien ulostulojen perusteella, kun taas avain-arvo-parit muodostaa enkooderi (Vaswani ym., 2017; Bahdanau ym., 2015); transformerimallissa on erityisiä integroituja enkooderi-dekooderi-huomiokerroksia (Vaswani ym., 2017). Alkuperäisessä transformerimallissa (mt.) käytetty huomiofunktio $A(Q, K, V)$ on

$$A(Q, K, V) = \sigma(\frac{QK^T}{\sqrt{d_k}})V$$

missä σ on softmax-funktio ja d_k on K- ja Q-vektorien ulottuvuuksien määrä.

Transformerimallissa enkooderi koostuu vuorottelevista huomiokerroksista ja yksinkertai-

sista FNN-kerroksista, joiden välillä tapahtuu lisäksi syötteen normalisointi (Vaswani ym., 2017). Dekooderissa on näiden lisäksi kolmas kerros, joka sisältää enkooderin tuottamaan vektoriin kohdistuvan huomiomekanismin (mt.). Transformerimallin rakenne on esitetty kaavamaisesti kuvassa 4.4.



Kuva 4.4: Transformerimalli. a) Mallin toiminta kokonaisuudessaan: syöte (magenta), enkooderikerrokset (vaaleansininen) ja dekooderikerrokset (punainen). b) Enkooderikerroksen sisäinen rakenne: huomiokerros (keltainen) ja FNN (vihreä). c) Dekooderikerroksen sisäinen rakenne: huomiokerrokset (keltainen ja violetti) sekä FNN (vihreä). Kuva: Otter ym., 2020.

4.4 Koneoppimiseen perustuvat POST-työkalut

Klassiseen koneoppimiseen perustuvista malleista POST-työkaluissa ja muutoinkin NLP:n alalla käytetyimpiä ovat olleet erilaiset stokastista laskentaa koneoppimiseen yhdistävät mallit (Otter ym., 2020; Chiche ja Yitagesu, 2022). Viime vuosina kehitys on kuitenkin keskittynyt yhä voimakkaammin neuroverkkoihin perustuviin syväoppiviin malleihin (Otter ym., 2020).

Suomen kielellä toimivista koneoppimiseen perustuvista työkaluista merkittävin lienee TurkuNLP:n* Turku Neural Parser Pipeline (TNPP), joka suomen lisäksi pystyy käsittelemään yli 50:tä muuta kieltä. Kanervan ym. (2018) mukaan TNPP:n sanaluokkien- ja muotojen merkitsemiseen käyttämä moduuli perustui alun perin Stanfordin yliopiston ryhmän edellisenä vuonna kehittämään työkaluun (Dozat ym., 2017); TNPP:n dokumentaation† mukaan tilalle on kuitenkin kehitetty kokonaan uusi moduuli vuonna 2021, ja työkalun nykyinen implementaatio hyödyntää kauttaaltaan BERT-malliin perustuvaa neuroverkkoa. BERT puolestaan on alun perin Googlen kehittämä transformeripohjainen kaksisuuntainen neuroverkkomalli (Devlin ym., 2019).

Erittäin laaja-alainen koneoppimiseen perustuva työkalusarja on Apache OpenNLP, joka

*https://turkunlp.org/
†https://turkunlp.org/Turku-neural-parser-pipeline/

dokumentaationsa* mukaan pystyy nykyisellään käsittelemään yli sataa kieltä ja soveltuu useisiin erilaisiin luonnollisen kielen käsittelyyn liittyviin tehtäviin, mukaan lukien sanaluokkien ja -muotojen merkitsemisen. Se pystyy hyödyntämään sekä perinteisempiä koneoppimismalleja että perseptronipohjaista neuroverkkoa (Horsmann ym., 2015; ks. myös em. dokumentaatio).

*https://opennlp.apache.org/docs/1.9.4/manual/opennlp.html

5 Tulokset ja analyysi

Tämä luku sisältää tekemäni kartoituksen tulokset. Aloitan suomen ja englannin kielikohtaisista tuloksista ja käyn tämän jälkeen yleisemmällä tasolla läpi muita kieliä. Luvun loppupuolella on pohdintaosio, jossa analysoin saamiani tuloksia ja esitän tekemäni johtopäätökset.

5.1 Englanti

Kattavia vertailuja erilaisista englannin kielen työkaluista ei ilmeisesti aivan viime vuosina ole tehty. Toistaiseksi uusin vaikuttaisi olevan Horsmann ym., 2015, jossa on vertailtu erilaisten työkalujen toimivuutta englannin ja saksan kielissä. Esimerkkinä sääntöpohjaisesta työkalusta on Horsmannin et al. tutkimuksessa käytetty Brillin malliin pohjautuvaa *Hepplen mallia*. Tässä vertailussa OpenNLP:n eri versiot pääsevät joko samaan tai hieman parempaan tarkkuuteen kuin Hepplen malli (n. 87-88 %), mutta Hepple on kertaluokkaa nopeampi (mt.).

Rajoitekielioppiin pohjautuvalle EngCG:lle puolestaan on jo 1990-luvulla raportoitu joissain tutkimuksissa jopa yli 99 %:n tarkkuus (Samuelsson ja Voutilainen, 1997). On kuitenkin huomattava, että teknologian kehittyessä myös työkaluille asetetut vaatimukset ovat kehittyneet, ja nykyisissä tutkimuksissa ominaisuuksia testataan yleensä monipuolisemmin (vrt. esim. Pirinen, 2019).

5.2 Suomi

Suomen kielellä rajoitekielioppiin perustuva yhdistelmätyökalu (Omorfi-MarMot) pääsee Pyysalon ym. (2015) mukaan 94,3 %:n tarkkuuteen, kun taas Turku Neural Parser Pipeline ainakin tietyissä tapauksissa jopa 97,6 %:iin (Kanerva et al., 2018: 137). Myös Omorfin kehittäjä Tommi Pirinen on verrannut Omorfin ja TNPP:n ominaisuuksia ja todennut, että TNPP pääsee nykyään parempiin tuloksiin (Pirinen, 2019): TNPP:n tarkkuus vertailussa on tehtävästä riippumatta n. 97 %, kun taas Omorfin on tehtävästä riippuen 92-94 %. Omorfi tosin suoriutui tehtävistä puolet nopeammin (5 min vs. 10 min), mutta

TNPP:nkään nopeutta ei voitane pitää kohtuuttoman hitaana.

5.3 Muut kielet

Sääntöpohjaisia malleja kehitettiin useiden Euroopan maiden virallisille kielille 1990-luvun lopulla ja 2000-luvun alkupuolella (Bick ja Didriksen, 2015). Sittemmin suuntaus on siirtynyt kohti neuroverkkopohjaisia koneoppimismalleja (Pirinen, 2019), joista monet toimivat useilla eri kielillä. Karkeasti mallin toimivuus näyttäisi korreloivan sen kanssa, miten paljon kielellä on puhujia ja miten vahva sen asema on; erityisesti Euroopan maiden viralliset kielet (joista monet ovat myös suuria kansainvälisiä kieliä) näyttävät pärjäävän vertailussa hyvin (Soria ym., 2014; Heinzerling ja Strube, 2019).

Erityisesti Euroopassa työkaluja on viime aikoina on myös pienemmille vähemmistökielille, kuten saamelaiskielille (Moshagen ym., 2013), kymrille (Neale ym., 2018), bretonille (Kanerva ym., 2018) ja baskille (Otegi ym., 2016). Kartoitukseni perusteella varsin aktiivista kehitystyötä näkyy tehdyn myös Intiassa (esim. Reddy ja Sharoff, 2011; Rathod ja Govilkar, 2015; Kumawat ja Jain, 2015); sen korostuminen tuloksissa johtunee yhtäältä siitä, että kyseessä on poikkeuksellisen monikielinen maa, ja toisaalta siitä, että Intiassa tietojenkäsittelytieteen asema yliopistoissa on vahva (Loyalka ym., 2019) ja tutkimuksen kieli on englanti, jolla pääosin olen etsinyt tutkimusta (Loyalkan et al. mukaan muita merkittäviä maita alalla ovat mm. Kiina ja Venäjä, joissa näin ei välttämättä aina ole).

Keskeisimmäksi havainnoksi aineistosta nousee se, että suurille tai virallisessa asemassa oleville kielille kehitetyt mallit ovat viime vuosina enenevässä määrin olleet neuroverkkopohjaisia koneoppimismalleja, kun taas vähemmistökielille sääntöpohjaiset mallit ovat edelleen olleet suosittuja. Monissa lähteissä (esim. Neale ym., 2018; Pirinen, 2019; Hedderich ym., 2021) on mainittu tälle myös selkeä syy: koneoppimismallit tarvitsevat opetusdatakseen valmiiksi annotoitua tekstiä, jota ne voivat käyttää mallina. Tällaista dataa taas on suuremmille ja vahvemmassa asemassa oleville kielille taas yleensä olemassa enemmän.

5.4 Pohdintaa

Tulosten vertailukelpoisuutta rajoittaa se, että eri tutkimuksissa on tutkittu eri työkaluja eri kielillä ja erilaisin menetelmin. Keskeinen erottava tekijä on myös aika; teknologia on kehittynyt nopeasti, eikä esimerkiksi 1990-luvulla tehdyissä tutkimuksissa ole vielä juuri-

kaan viitattu neuroverkkoihin tai syväoppimiseen perustuviin työkaluihin siitä yksinkertaisesta syystä, ettei niitä silloin vielä ollut olemassa. Myös työkaluille asetetut vaatimukset ovat kasvaneet teknologian kehittyessä, ja niinpä niiden ominaisuuksia on uudemmissa tutkimuksissa testattu usein monipuolisemmin (vrt. esim. Pirinen, 2019).

Vaikuttaisi kuitenkin siltä, että kielen morfologisella kompleksisuudella tai muilla rakenteellisilla ominaisuuksilla ei ole juurikaan tekemistä sen kanssa, millaiset POST-työkalut sen käsittelyyn soveltuvat. Yleisesti ottaen neuroverkkoihin perustuvat koneoppivat järjestelmät ovat ilmeisesti viime vuosina kehittyneet sellaiselle tasolle, että ne tuottavat parhaan tuloksen niin suomen kuin englannin kuin useiden muidenkin kielten kohdalla kielen rakenteellisista ominaisuuksista riippumatta. Neuroverkoilla on tässä suhteessa myös se selkeä etu, että usein sama työkalu toimii useilla eri kielillä, kun taas sääntöpohjainen malli on luotava kullekin kielelle erikseen.

Aiemmin sääntöpohjaisten mallien etuna on ollut niiden *läpinäkyvyys*: koska kaikki säännöt on koodattu käsin, on virhelähteet verrattaen helppo löytää ja korjata muokkaamalla tai lisäämällä sääntöjä. Neuroverkkopohjaisen työkalun tekemien virheiden syy ei useinkaan ole samalla tavoin selvitettävissä. Nykyiset neuroverkot kuitenkin suoriutuvat yleensä riittävän hyvin, ettei tällä ole juurikaan käytännön merkitystä - ainakin silloin, kun opetusdataa on käytettävissä riittävästi.

Niinpä kielen rakenteellisten ominaisuuksien sijaan keskeiseksi tekijäksi erityyppisten mallien käyttökelpoisuuden arvioinnissa vaikuttaisikin nousevan koneoppivien mallien *opetusdataksi kelpaavan dokumentaation määrä*. Kielille, joille esimerkiksi valmiiksi annotoitua dataa on käytettävissä runsaasti (*resource-rich languages*), koneoppivat mallit soveltuvat hyvin. Tällaisia kieliä ovat esim. suomi ja englanti (Alnajjar, 2021). Suurin osa maailmassa puhutuista n. 7000 kielestä kuitenkin on erilaisia vähemmistökieliä, joille valmiiksi annotoitua dokumentaatiota on käytettävissä vähän tai ei lainkaan (*under-resourced* t. *low-resource languages* tms.) (Soria ym., 2014), ja näille kielille sääntöpohjaisia malleja saattaa olla helpompi luoda (Pirinen, 2019). Pirinen (mt.) on myös huomauttanut, että sääntöpohjaisia malleja voidaan hyödyntää valmiiksi annotoitujen dokumenttien *luomiseen* ja siten koneoppivien mallien apuna.

6 Yhteenveto

Tässä tutkielmassa on käyty läpi sanaluokkien merkitsemiseen ja morfologiseen analyysiin tarkoitettuja automaattisia työkaluja (POST-työkaluja) sekä niiden soveltuvuutta erilaisiin kieliin. Tällaiset työkalut muodostavat perustan laajemmalle luonnollisen kielen käsittelyn alalle (NLP).

POST-työkalun tehtävänä on käydä läpi sille syötettyä kieliaineistoa ja merkitä eli annotoida jokaisen sanan kohdalle sanan sanaluokka ja mahdollinen taivutusmuoto. Koska sama muoto voi usein olla tulkittavissa usealla eri tavalla riippuen sanan kontekstista, tulisi POST-työkalun myös kyetä ottamaan konteksti (esim. ympäröivä lause) huomioon. Tehtävän tarkempaan luonteeseen vaikuttaa käsiteltävä kieli, sillä eri kielet eroavat toisistaan huomattavasti siinä, kuinka paljon taivutusta niissä esiintyy (Voutilainen, 2003). Tässä tutkielmassa on ensisijaisesti käytetty esimerkkikielinä englantia, jossa esiintyy varsin vähän taivutusta, sekä suomea, jossa sitä esiintyy runsaasti.

POST-työkaluissa käytetyt teknologiat voidaan jakaa kolmeen päätyyppiin, eli sääntöpohjaisiin, tilastollisiin ja koneoppiviin malleihin, joista tässä tutkielmassa on keskitty sääntöpohjaisiin ja koneoppiviin. Sääntöpohjaiset mallit perustuvat nimensä mukaisesti malliin valmiiksi koodattuihin sääntöihin; tyypillisesti syöte muunnetaan mahdollisiksi annotaatioiksi äärellistilaisen transduktorin avulla, minkä jälkeen erityiset rajoitesäännöt pyrkivät kontekstin perusteella disambiguoimaan muodon (Karlsson, 1995). Koneoppimisessa taas kone pyrkii itsenäisesti kehittämään menetelmän, jolla se pystyy tuottamaan halutunlaisen tuloksen, tyypillisesti sille malliksi annetun *opetusdatan* (esimerkiksi valmiiksi annotoitua tekstiä) perusteella (Jung, 2022). Uusimmat koneoppivat työkalut perustuvat syväoppiviin neuroverkkomalleihin, eli ne koostuvat useista kerroksista, joista jokainen puolestaan koostuu biologisia hermosoluja karkealla tasolla muistuttavista yksiköistä eli neuroneista (Goodfellow ym., 2016; Chiche ja Yitagesu, 2022).

Keskeisin tulos tutkimuksessani on nähdäkseni se, että parhaisiin tuloksiin sanaluokkien merkitsemisessä ja morfologisessa analyysissa pääsevät nykyään neuroverkkoihin pohjautuvat koneoppimismallit riippumatta käsiteltävänä olevasta kielestä ja sen rakenteellisista ominaisuuksista. Poikkeuksen tähän muodostavat kuitenkin ne kielet, joille koneoppimismallien tarvitsemaa valmiiksi käsiteltyä dataa on saatavilla liian niukalti.

Koska kartoitukseni perustuu useisiin eri-ikäisiin ja eri tavoin tehtyihin tutkimuksiin, eri

kieliä koskevia tuloksia voi pitää vertailukelpoisina vain hyvin yleisellä tasolla. Tarpeen olisikin tehdä jatkossa tarkempi kartoitus tai mieluiten erillinen tutkimus, jossa eri kieliin olisi sovellettu mahdollisimman samanlaisia malleja mahdollisimman samanlaiseen aineistoon ja arvioitu mahdollisimman samanlaisin kriteerein.

Lähteet

Alnajjar, K. (maaliskuu 2021). ”When Word Embeddings Become Endangered”. *Multilingual Facilitation*, s. 275–288. DOI: 10.31885/9789515150257.24. URL: http://dx.doi.org/10.31885/9789515150257.24.

Bahdanau, D., Cho, K. ja Bengio, Y. (2015). ”Neural Machine Translation by Jointly Learning to Align and Translate”. *CoRR* abs/1409.0473.

Bengio, Y. (tammikuu 2009). ”Learning Deep Architectures for AI”. *Foundations* 2, s. 1–55. DOI: 10.1561/2200000006.

Bick, E. (kesäkuu 2007). ”Hybrid Ways to Improve Domain Independence in an ML Dependency Parser”. Teoksessa: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, s. 1119–1123. URL: https://aclanthology.org/D07-1120.

– (2020). ”Towards an Automatic Mark-up of Rhetorical Structure in Student Essays”. Teoksessa: *Proceedings of CICLing 2018 - 19th International Conference on Computational Linguistics*. null ; Conference date: 18-03-2018 Through 24-03-2018.

Bick, E. ja Didriksen, T. (toukokuu 2015). ”CG-3 - Beyond Classical Constraint Grammar”. Teoksessa: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, s. 31–39.

Brill, E. (1992). ”A simple rule-based part of speech tagger”. Teoksessa: *Proceedings of the third conference on Applied natural language processing (ANLC ’92)*. Stroudsburg, PA, USA: Association for Computational Linguistics, s. 152–155. DOI: 10.3115/974499.974526.

Chiche, A. ja Yitagesu, B. (2022). ”Part of speech tagging: a systematic review of deep learning and machine learning approaches”. *Journal of Big Data* 9.10, s. 891–921. DOI: https://doi.org/10.1186/s40537-022-00561-y.

Cloeren, J. (1999). ”Tagsets”. Teoksessa: *Syntactic Wordclass Tagging*. Toim. H. van Halteren. Dordrecht: Springer Science+Business Media, s. 37–54.

de Marneffe, M.-C., Manning, C., Nivre, J. ja Zeman, D. (2015). ”Universal Dependencies”. *Computational Linguistics* 47.2, s. 255–308. DOI: https://doi.org/10.1162/coli_a_00402.

Devlin, J., Chang, M.-W., Lee, K. ja Toutanova, K. (kesäkuu 2019). ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. Teoksessa: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, s. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Dozat, T., Qi, P. ja Manning, C. (2017). ”Stanford’s graph-based neural dependency parser at the conll 2017 shared task”. Teoksessa: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies pages*. Vancouver, Canada: Association for Computational Linguistics, s. 20–30.

Goldberg, Y. (syyskuu 2016). ”A Primer on Neural Network Models for Natural Language Processing”. *J. Artif. Int. Res.* 57.1, s. 345–420. ISSN: 1076-9757.

Goodfellow, I., Bengio, Y. ja Courville, A. (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Hakulinen, A., Vilkuna, M., Korhonen, R., Koivisto, V., Heinonen, T.-R. ja Alho, I. (2004). *Iso suomen kielioppi*. Helsinki: Suomalaisen Kirjallisuuden Seura.

Hedderich, M., Lange, L., Adel, H., Strötgen, J. ja Klakow, D. (tammikuu 2021). ”A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios”. Teoksessa: s. 2545–2568. DOI: 10.18653/v1/2021.naacl-main.201.

Heinzerling, B. ja Strube, M. (heinäkuu 2019). ”Sequence Tagging with Contextual and Non-Contextual Subword Representations: A Multilingual Evaluation”. Teoksessa: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, s. 273–291. DOI: 10.18653/v1/P19-1027. URL: https://aclanthology.org/P19-1027.

Hepple, M. (2000). ”Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models”. Teoksessa: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Hong Kong.

Horsmann, T., Erbs, N. ja Zesch, T. (2015). ”Fast or Accurate? – A Comparative Evaluation of PoS Tagging Models”. Teoksessa: *Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology*. University of Duisburg-Essen, Germany.

Hämäläinen, M. ja Alnajjar, K. (2015). ”The Current State of Finnish NLP”. Teoksessa: *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*. Syktyvkar, Komi Republic, s. 54–61.

Hämäläinen, M. ja Alnajjar, K. (marraskuu 2019). ”Generating Modern Poetry Automatically in Finnish”. Teoksessa: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, s. 5999–6004. DOI: 10.18653/v1/D19-1617. URL: https://aclanthology.org/D19-1617.

Jung, A. (2022). *Machine Learning: The Basics*. Singapore: Springer.

Jurafsky, D. ja Martin, J. (2021). *Speech and Language Processing*. Kolmannen painoksen julkaisematon käsikirjoitus. Prentice Hall.

Kanerva, J., Ginter, F., Miekka, N., Leino, A. ja Salakoski, T. (2018). ”Turku Neural Parser Pipeline: An End-to-End System for the CoNLL”. Teoksessa: *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels, Belgium: Association for Computational Linguistics, s. 133–142.

Kann, K., Lacroix, O. ja Søgaard, A. (2020). ”Weakly Supervised POS Taggers Perform Poorly on Truly Low-Resource Languages”. Teoksessa: *AAAI*.

Karlsson, F. (1995). ”Designing a parser for unrestricted text”. Teoksessa: *Constraint Grammar: A Language-Independent Framework for Parsing Unrestricted Text*. Toim. F. Karlsson, A. Voutilainen, J. Heikkilä ja A. Anttila. Berlin / New York: Mouton de Gruyter, s. 1–40.

– (2008). *Yleinen kielitiede*. Uudistetun laitoksen kolmas painos. Helsinki: Gaudeamus.

Kirov, C., Cotterell, R., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Mielke, S. J., McCarthy, A., Kübler, S., Yarowsky, D., Eisner, J. ja Hulden, M. (toukokuu 2018). ”UniMorph 2.0: Universal Morphology”. Teoksessa: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1293.

Kłosowski, P. (2018). ”Deep Learning for Natural Language Processing and Language Modelling”. Teoksessa: *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, s. 223–228. DOI: 10.23919/SPA.2018.8563389.

Kumawat, D. ja Jain, V. (2015). ”POS tagging approaches: A comparison”. *International Journal of Computer Applications* 118.6.

LeCun, Y., Bengio, Y. ja Hinton, G. (2015). ”Deep Learning”. *Nature* 521, s. 436–444. DOI: https://doi.org/10.1038/nature14539.

Lowe, J. J. (2016). ”English possessive ’s: clitic and affix”. *Natural Langugage and Linguistic Theory* 34, s. 157–195. DOI: https://doi.org/10.1007/s11049-015-9300-1.

Loyalka, P., Liu, O. L., Li, G., Chirikov, I., Kardanova, E., Gu, L., Ling, G., Yu, N., Guo, F., Ma, L., Hu, S., Johnson, A. S., Bhuradia, A., Khanna, S., Froumin, I., Shi, J., Choudhury, P. K., Beteille, T., Marmolejo, F. ja Tognatta, N. (2019). "Computer science skills across China, India, Russia, and the United States". *Proceedings of the National Academy of Sciences* 116.14, s. 6732–6736. DOI: 10.1073/pnas.1814646116. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1814646116. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1814646116.

Moshagen, S. N., Pirinen, F. A. ja Trosterud, T. (2013). "Building an open-source development infrastructure for language technology projects". Teoksessa: *NODALIDA*.

Neale, S., Donnelly, K., Watkins, G. ja Knight, D. (toukokuu 2018). "Leveraging Lexical Resources and Constraint Grammar for Rule-Based Part-of-Speech Tagging in Welsh". Teoksessa: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1623.

Otegi, A., Ezeiza, N., Goenaga, I. ja Labaka, G. (syyskuu 2016). "A modular chain of NLP tools for Basque". Teoksessa: vol. 9924, s. 93–100. ISBN: 978-3-319-45509-9. DOI: 10.1007/978-3-319-45510-5_11.

Otter, D., Medina, J. ja Kalita, J. (huhtikuu 2020). "A Survey of the Usages of Deep Learning for Natural Language Processing". *IEEE Transactions on Neural Networks and Learning Systems* PP, s. 1–21. DOI: 10.1109/TNNLS.2020.2979670.

Paroubek, P. (2007). "Evaluating Part-of-Speech Tagging and Parsing". Teoksessa: *Evaluation of Text and Speech Systems*. Toim. L. Dybkjær, H. Hemsen ja W. Minker. Dordrecht: Springer, s. 99–124.

Pirinen, T. (2008). "Suomen kielen äärellistilainen morfologinen jäsennin avoimen lähdekoodin resurssein". Tutkielma. Helsingin yliopisto.

– (toukokuu 2015). "Omorfi — Free and open source morphological lexical database for Finnish". Teoksessa: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*. Vilnius, Lithuania: Linköping University Electronic Press, Sweden, s. 313–315. URL: https://aclanthology.org/W15-1844.

– (tammikuu 2019). "Neural and rule-based Finnish NLP models—expectations, experiments and experiences". Teoksessa: *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*. Tartu, Estonia: Association for Computational Linguistics, s. 104–114. DOI: 10.18653/v1/W19-0309. URL: https://aclanthology.org/W19-0309.

Pyysalo, S., Kanerva, J., Missilä, A., Laippala, V. ja Ginter, F. (2015). "Universal Dependencies for Finnish". Teoksessa: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*. Vilnius, Lithuania: Linköping University Electronic Presss, s. 163–172.

Rathod, S. ja Govilkar, S. (2015). "Survey of various POS tagging techniques for Indian regional languages". Teoksessa.

Reddy, S. ja Sharoff, S. (2011). "Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources". Teoksessa: *Proceedings of the fifth international workshop on cross lingual information access*, s. 11–19.

Rijkhoff, J. (tammikuu 2001). "Verbs and nouns from a cross-linguistic perspective". *Italian Journal of Linguistics (Rivista di Linguistica)* 14, s. 115–157.

Rudin, C. ja Radin, J. (22. marraskuuta 2019). "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition". *Harvard Data Science Review* 1.2. https://hdsr.mitpress.mit.edu/pub/f9kuryi8. DOI: 10.1162/99608f92.5a8a3a3d. URL: https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

Sadredini, E., Guo, D., Bo, C., Rahimi, R., Skadron, K. ja Wang, H. (2018). "A Scalable Solution for Rule-Based Part-of-Speech Tagging on Novel Hardware Accelerators". Teoksessa: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, s. 665–674. ISBN: 9781450355520. DOI: 10.1145/3219819.3219889. URL: https://doi.org/10.1145/3219819.3219889.

Samuel, A. (1959). "Some Studies in Machine Learning Using the Game of Checkers". *IBM Journal of Research and Development* 3.3, s. 210–229. DOI: 10.1147/rd.33.0210.

Samuelsson, C. ja Voutilainen, A. (1997). "Universal Dependencies for Finnish". Teoksessa: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, Spain: Association for Computational Linguistics, s. 246–253.

Sapir, E. (1921). *Language: An Introduction to the Study of Speech*. New York: Harcourt, Brace & Co.

Soria, C., Del Gratta, R., Khan, F., Mariani, J. ja Frontini, F. (toukokuu 2014). "The LREMap for Under-Resourced Languages". Teoksessa.

Taylor, A., Marcus, M. ja Santorini, B. (2003). "The Penn Treebank: An Overview". Teoksessa: *Treebanks: Building and Using Parsed Corpora*. Toim. A. Abeillé. Text, Speech and Language Technology Vol. 20. Dordrecht: Springer. DOI: https://doi.org/10.1007/978-94-010-0201-1_1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. ja Polosukhin, I. (kesäkuu 2017). ”Attention Is All You Need”. Teoksessa: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. Toim. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan ja R. Garnett.

Voutilainen, A. (1997). ”EngCG tagger, Version 2”. Teoksessa: *Sprog og Multimedier*. Toim. T. Brøndsted ja I. Lytje. Aalborg: Aalborg Universitetsforlag, s. 20–38.

– (2003). ”Part-of-speech tagging”. *The Oxford handbook of computational linguistics*, s. 219–232.

Liite A Annotaatioskeema

`DET` = determinatiivi
`N` = substantiivi
`V` = verbi

`PUNCT` = välimerkki

`DEF` = määräinen
`INDF` = epämääräinen

`FIN` = finiittinen
`NFIN` = infiniittinen

`SG` = yksikkö
`PL` = monikko

`PST` = mennyt aika
`PRF` = perfekti

`PTCP` = partisiippi

`NOM` = nominatiivi
`GEN` = genetiivi
`INESS` = inessiivi

`PSS1S` = possessiivi, yksikön 1. persoona

`EOS` = virkkeen loppu