Tartu Ülikool

Humanitaarteaduste ja kunstide valdkond

Eesti ja üldkeeleteaduse instituut

Laura Katrin Leman

Tehisnärvivõrgul põhinevate lemmatiseerijate võrdlev analüüs eesti keeles

Bakalaureusetöö

Juhendaja Kairit Sirts, PhD

Tartu 2019

SISUKORD

SISSEJUHATUS

Lemmatiseerimise all mõistetakse sõna algvormi ehk lemma leidmist. Eesti keeles on selleks nimisõnade puhul üldiselt ainsuse nominatiiv, tegusõnade puhul ma-infinitiiv. Lemmast võib mõelda ka kui sõnastikus esinevast sõnavormist. Näiteks lause *Õpilastel on koolis raskusi* oleks lemmatiseeritud kujul *Õpilane olema kool raskus*.

Leidmaks sõna lemmat, on vaja tuvastada mitmesugustest muutevormidest sõna algvorm ning seejuures arvestada homonüümiaga. Näiteks sõna *teid* lemma võiks olla nii *tee* kui ka *teie* – esimesel juhul esineb sõna *tee* mitmuse partitiivis, teisel juhul on mõeldud mitmuse 2. isikut *teie* osastavas käändes. Arusaadavalt esineb lemmatiseerimise juures raskusi just morfoloogiliselt rikaste keelte puhul, näiteks eesti keeles.

Miks on lemmatiseerimist tarvis? Lemmatiseerimisel on mitmeid rakendusi tekstianalüüsis. Eeltöötlusvahendina saab seda kasutada nii tekstide liigitamisel (näiteks teemade või autorite järgi) kui ka keelekasutuse uurimisel (Manjavacas, Kádár, Kestemont 2019). Eriti tähtis on lemmatiseerimine automatiseeritud infootsingute sooritamisel – see võimaldab lugeda ühe sõna erinevad morfoloogilised vormid kõik ühe algvormi juurde kuuluvaks (Muischnek, Kaalep, Sirel 2011). Ilma selleta loetaks infopäringut tehes näiteks muutevormid *tuba*, *toa* ja *tuppa* erinevateks sõnadeks. Lemma teadmine on abiks ka keeleõppijale, kes ei pruugi osata muutevormi järgi sõnastikust sõna leida. Lemmatiseerimine on ka eelprotsess süntaksianalüüsile, näiteks parsimisprogrammides MaltParser ja MSTParser (Habernal, Matousek 2013: 291)

Eesti keeles on lemmatiseerimiseks seni loodud ja kasutatud eelkõige statistilisi ja reeglipõhiseid mudeleid, näiteks OÜ Filosofti loodud lemmatiseerija Vabamorf [1] . Vabamorfi kasutatakse ka vabavaralises Pythoni teegis EstNLTKclemat[2], mis võimaldab muuhulgas etteantud teksti või sõna lemmatiseerimist.

[1] https://github.com/Filosoft/vabamorf

[2] https://github.com/estnltk)

Autorile teadaolevalt ei ole loodud ühtki neuromudelitel põhinevat lemmatiseerijat spetsiaalselt eesti keelt silmas pidades. Küll on mitmed uurijad loonud erinevatel tehisnärvivõrkude arhitektuuridel põhinevaid lemmatiseerijaid ning kasutanud neid kümnete erinevate keelte, sealhulgas eesti keele korpuste peal (Bergmanis ja Goldwater 2018; Kanerva, Ginter, Salakoski 2019; Makarov ja Clematide 2018; Manjavacas, Kádár, Kestemont 2019; Kondratyuk jt 2018).

Antud töö eesmärk on analüüsida erinevate tehisnärviarhitektuuridel põhinevate lemmatiseerijate täpsust mitmel eestikeelsel andmestikul ja teha tulemustest analüüs, mida saaks tulevikus kasutada eesti keele lemmatiseerimise parandamiseks, ning mida saaks võrrelda olemasolevate eesti keele automaatsete lemmatiseerijatega. Selleks võeti aluseks CoNLL 2018 Shared Task võistlusel lemmatiseerimise kategoorias eesti keeles vastavalt 1. ja 3. koha saavutanud TurkuNLP (Kanerva, Ginter, Salakoski 2019) ja Stanfordi (Qi jt 2018) mudelid ja morfoloogiliselt rikastele keeltele mõeldud LemmaTag (Kondratyuk jt 2018) mudeli. Kõiki mudeleid treeniti ja testiti Universal Dependencies 2.3 eesti keele puupanga peal (Nivre jt 2018), morfoloogiliselt ühestatud korpuse[3] ja eesti keele murdekorpuse peal. Iga korpuse peal treenitud mudelit kasutati sama korpuse ja teiste korpuste testhulgal ning lisaks Estonian Web Treebank (edaspidi EWTB) korpuse (Särg, Muischnek, Müürisep 2018) materjalidel, mõõdeti lemmatiseerimise täpsus ja teostati tulemuste analüüs.

Töö koosneb neljast peatükist. Esimeses antakse ülevaade lemmatiseerimise teoreetilisest käsitlusest ja olemasolevatest lemmatiseerimisprogrammidest: reeglipõhistest ning andme- ja tehisnärvivõrkudepõhistest.

Teises peatükis kirjeldatakse töös kasutatud tehnilisi vahendeid, nende ehitust ja kasutamist. Kolmandas peatükis käsitletakse lähemalt töös kasutatud korpuseid ja nende viimist mudelitele sobivale kujule. Neljandas peatükis on esitatud töö tulemused antud korpuste ja mudelite põhjal ning nende põhjal tehtud vigade ja probleemide analüüs.

[3] https://keeleressursid.ee/et/keeleressursid-cl-ut/korpused/83-article/clutee-lehed/143-morfkorpus

1. LEMMATISEERIMISE OLEMUS JA AUTOMAATNE LAHENDAMINE

1.1 Lemmatiseerimise olemus

Sõnad esinevad tekstis ja kõnes erinevates grammatilistes vormides, sealhulgas eri käänetes, arvus, isikus jne. Igat sellist grammatilist vormi, milles mõni sõna esineda võib, nimetatakse **sõnavorm**iks. Tekstis esinevat sõnavormi ehk igat eraldi sõna tekstis kutsutakse **sõne**ks. Igal sõnavormide kogumil on määratud esindaja – selle sõna **algvorm** ehk **lemma**: eesti keeles noomenitel ainsuse nimetava vorm, verbidel ma-infinitiiv. (Erelt, jt 2007)

Lemmatiseerimine tähendab sõna algvormi ehk lemma leidmist, analüüsides sõna morfoloogiliselt ja vajadusel morfeeme eemaldades või muutes (Manning, Raghavan, Schütze 2008: 32). Alljärgnevas näites 1 toodud väljendis tuleks tuvastada mitmuses seesütleva käände lõpp (mitmuses omastava käände tunnus -de/-te koos liitega -s) ja selle eemaldamisel tagastada sõnade algvormid.

(1) Sõnavormid: *suurtes majades*

Lemmad*: suur maja*

Ainult morfoloogiliste tunnuste vaatlemisel ja muutmisel võib aga lemmatiseerides saada ühele sõnale vasteks mitmese analüüsi. Näiteks sõna *võis* võib tähendada nii nimisõna *või* seesütlevas käändes kui ka tegusõna *võima* imperfekti ainsuse 3. pöördes. Selliste variantide seast peaks lemmatiseerides tagastama ainult ühe. Mitmese analüüsi seast ühe (õige) vaste valimist nimetatakse **ühestamiseks**. Ühese lemma leidmist on tarvis mitmetes keeletehnoloogilistes rakendustes, näiteks eelsammuna süntaksitöötluses (Habernal, Matousek 2013: 291), infopäringutes (Muischnek, Kaalep, Sirel 2011, teksti liigitamisel ja keelekasutuses esinevate sõnade uurimisel (Manjavacas, Kádár, Kestemont 2019) jpm.

Lemmatiseerimise automaatseks lahendamiseks on loodud mitut tüüpi programme, nii keelespetsiifilisi kui ka keeleüleseid. Järgnevas ülevaates on need jagatud kolmeks: reeglipõhisteks, andmepõhisteks ja tehisnärvivõrkudel põhinevateks.

1.2 Reeglipõhised lemmatiseerimisprogrammid

Reeglipõhistes programmides on tavaliselt kasutusel hulk keelespetsiifilistel grammatilistel teadmistel põhinevaid reegleid, mille alusel tuletada sõnavormist lemmat (Plisson, Lavrac, Mladenic 2004: 2). Lisaks tarvitatakse sõnastikku, millest vaadatakse järele, kas sõnavormi jaoks on lemma juba defineeritud.

Eesti keele automaatseks morfoloogiliseks analüüsiks on loodud laia funktsionaalsusega tarkvaraprogramm Vabamorf, mis sisaldab muuhulgas reeglipõhist lemmatiseerijat (EKT). Ka eesti keele töötluseks mõeldud Pythoni teegis EstNLTK toimub morfoloogiline analüüs ja süntees Pythoni rakendusliidessese mähitud Vabamorfi kaudu (Orasmaa jt 2016: 2461).

Vabamorf kasutab sõnede analüüsimiseks nii sõnastikku, milles on olemas põhilised lihtsõnad ja sagedasemad pärisnimed ning lühendid. Tuletisi ja liitsõnu analüüsitakse algortmiliselt, leides igale sõnale kõige tõenäolisema osadeks jagunemise viisi. (Kaalep 1998: 23) Lihtsõnade analüüsimiseks lõigatakse lõpp maha, otsitakse sõnastikust tüvi ja kontrollitakse lõpu ning tüve kokkusobivust. Tuletiste analüüsiks on kasutusel produktiivsed sufiksid koos nende lubatud kombinatsioonide loendiga. Liitsõnade analüüsil arvestatakse liitsõnade moodustamise vormilisi piiranguid ja tõenäolisemate tüvede loendeid. (ibid., 24–25)

Ka Eesti Keele Instituudis on reeglipõhise morfoloogiasüsteemi raames loodud morfoloogilise analüüsi tarkvara, mis sisaldab muuhulgas lemmatiseerijat, seda küll ilma mitmesuse ühestamiseta (Viks 2000).[4]

1.3 Andmepõhised lemmatiseerijad

Andmepõhistes lemmatiseerijates kasutatakse eeldefineeritud reeglite asemel statistilisi mudeleid, millega arvutatakse andmete põhjal välja kõige tõenäolisem variant, mis mingi sõnavormi lemmaks sobiks.

[4] http://kn.eki.ee/tool/?m=morfoloogia

Ka Vabamorfis on kasutusel andmepõhine ehk statistiline osa, nimelt on Vabamorfi ühestaja aluseks Markovi peitmudel. Selles tõenäosuslikus mudelis vaadeldakse lauset kui märgendite, mitte sõnade järjestust. Ühestamata lausel võib olla mitu potentsiaalset märgendite järjestust, ent kuna vaid üks neist on õige, valib mudel selle, mis on kõige tõenäolisem. Sealjuures oleneb iga üksiku märgendi tõenäosus lauses vaid sellele eelneva märgendi tõenäosusest, arvestamata laiemat konteksti. (Kaalep, Vaino 1998: 33)

Lemmatiseerimiseks võib edukalt kasutada ka log-lineaarset mudelit, nagu seda on teinud Müncheni ja Johns Hopkinsi ülikooli teadlased programmiga Lemming (Müller jt 2015). Selles leitakse sõnavormi lemma sõnavormi enda ja tema morfoloogiliste atribuutide kaudu. Selleks leitakse treenimiseks kasutatavatest andmetest sõnavormist ja sellele vastavast lemmast koosnevate paarida pikim ühisjada ja modelleeritakse selle afiksipaare. Ühisjada puudumisel esitatakse vajalikud asendused sõnavormi lemmaks teisendamisel. Nii saadakse töötluspuu, mis salvestab prefiksite ja sufiksite pikkused või vajalikult asendused ning oskab vastavalt sellele sõnavormi lemmat leida, näiteks üks ja sama puu suudab lemmatiseerida sõned *worked* ja *touched* vastavalt lemmadeks *work* ja *touch*. Kõik puud, mida saab kasutada rohkem kui ühe sõnavormi ja lemma paari jaoks, ekstraheeritakse. Lemmatiseerimisel kasutatakse iga sõne jaoks kõikvõimalikke puid koos mitmesuguste puudel, sõnavormidel ja lemmadel põhinevate tunnustega. (ibid., 1–2)

1.4 Tehisnärvivõrgud ja nendel põhinevad lemmatiseerijad

Tehisnärvivõrk on andmetöötlusmudel, mis on saanud oma nime ja tõuke nende loomiseks (inim)aju toimimisest (Goldberg 2017: 2). Tehisnärvivõrgud on sisuliselt kogumik andmetöötluselemente, mille toimimist võib võrreldada neuronitega ja mis on nagu neuronidki omavahel ühenduses. Need elemendid või neuronid õpivad treeningandmete mustrite põhjal ning annavad enda väljundi, millele on antud tugevushinnang või kaal, teistele neuronitele sisendina ette. (Gurney 2004: 13) Niisiis transfomeerib tehisnärvivõrk etteantud andmeid, kuni viimane transformatsioon ennustab väljundi (Goldberg 2017: 2). Tehisnärvivõrkudel põhinevate lemmatiseerijate erinevus

reeglipõhistest lemmatiseerijatest seisnebki selles, et närvivõrgud on võimelised etteantud andmete põhjal ise õppima, kuidas sõnu algvormiks taandada, ega sõltu vaid etteantud reeglitest või sõnastikest.

Tehisnärvivõrkudele antakse andmeid õppimiseks ette treenimis- ja arendushulga kujul. Esimene on mõeldud selleks, et võrk saaks andmete pealt mustreid ja seaduspärasusi õppida; arendushulka kasutatakse mudeli optimiseerimiseks. Kui mudel on treenitud, hinnatakse tema täpsust testhulgaga: mudel ennustab etteantud testhulgale soovitud väljundi ning seda võrreldakse testhulga esialgsete, õigete tunnustega. (ibid.) Olemasolevaid andmeid treenimis-, arendus- ja testimishulgaks jaotades peaks kõige suurem osa, ca 60%-80% jääma treeningandmeteks, arendus- ja testimisandmete saamiseks jagatakse järelejäänud andmed pooleks.

Tehisnärvivõrgu sisendina võib kasutada väga erinevas formaadis andmeid. Lemmatiseerimise puhul on võimalik anda ette näiteks lihtsalt sõnavormid ilma konteksti või märgenduseta. Lisaks võib närvivõrgu sisendiks olla kontekstipõhine liugaken: see sisaldab sõnavormi koos temast paremale ja vasakule jääva *n* sõnavormiga, *n* on vabalt valitav täisarv. Seda on kasutanud näiteks Bergmanis ja Goldwater (2018) oma programmis Lematus. Veel võib lemmatiseerimisel treeningandmetena kasutada sõnavormi, millele on lisatud morfoloogiline märgendus. Viimast kasutavad siin töös vaadeldavad mudelid (Kanerva, Ginter, Salakoski 2019; Kondratyuk jt 2018; Qi jt 2018).

Keeletöötluseks mõeldud närvivõrkude puhul on oluline roll **sõnavektor**itel ehk sõnade vektoresitusel. Sõnavektor ongi vektoriks teisendatud sõna: vektoriteks teisendamine tähendab diskreetsete üksikute märkide muutmist pidevateks vektoriteks. See teeb kirjasõnadest matemaatilised objektid, kusjuures vektorite omavaheline kaugus on seotud sõnadevahelise kaugusega. Selliseid vektoreid on mudelil võimalik hõlpsamini käitleda. (Goldberg 2017: 3)

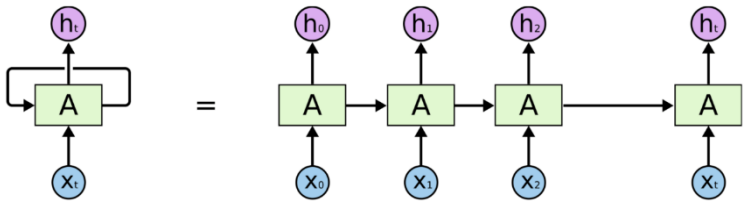
Tehisnärvivõrgud on väga lai termin, mis hõlmab mitmed erinevaid mudelitüüpe või arhitektuure. Järgnevalt on toodud välja käesolevas töös kasutatud mudelitüübid koos lühikese ülevaatega.

1.4.1 Rekurrentsed tehisnärvivõrgud

Rekurrentsed tehisnärvivõrgud (lühend **RTN)** koosnevad kolmest kihist: sisendkihist, rekurrentsest peitkihist ja väljundkihist. Nende eripära ja eelis seisneb selles, et nad on võimelised sisendit meeles pidama, sest peitkihid toimivad kui närvivõrgu mälu ja peitkihi seisund mingis ajahetkes on tingitud tema eelmisest seisust. (Salehinejad jt 2017: 1–2) Seega sobivad rekurrentsed närvivõrgud väga hästi keeletehnoloogilistesse rakendustesse, sest nii lemmatiseerides, tõlkides jms on vaja, et mudel hoiaks sisendit sekventsite kaupa meeles, st uut tähemärki sisendiks saades on eelnevad tähemärgid ehk sõne või lause eelmine osa mudeli mälus olemas.

Rekurrentse närvivõrgu sisendkiht võtab sisse vektorid. Nii sisendkihi kui ka väljundkihi ühikud on seotud peitkihi peitühikutega, kusjuures igale sidemele on omistatud teatud kaal. Peitühikud on omavahel rekurrentselt (korduvalt) seotud. Peitkihid annavad igal ajasammul väljundkihile sisendvektoril põhineva ennustuse ja sisaldavad infot närvivõrgu eelnevate seisundite kohta üle mitme ajasammu. Tänu sellele infole ongi võimalik teha täpseid ennustusi sekventsiaalse info kohta ja seda sekventsiaalselt väljastada. (ibid.) Joonisel 1 on kujutatud rekurrentse närvivõrgu struktuuri: sisend (x_t) läheb närvivõrku A, mis pärast rekurrentset töötlust väljastab lõpuks väljundi (h_t) (MLB)

Rekurrentne närvivõrk võib esineda ka kahesuunalises vormis, **bidirektsionaalse** rekurrentse närvivõrguna (lühend BRTN). Kui tavalised RTN-id kasutavad treenimiselt vaid sisendile eelnevat konteksti, siis kahesuunalised RTN-id arvestavad ka järgnevat konteksti. Selleks töötleb üks RTN sekventsi algusest lõpuni pärisuunas ja teine RTN töötleb andmeid vastupidi, lõpust alguseni. Niimoodi saab sisendi töötlemisel arvestada nii juba töötletud kui ka alles eesootavate andmetega. (ibid., 8)



Joonis 1. Rekurrentse närvivõrgu skeem (MLB)

1.4.2 LSTM närvivõrgud

Rekurrentsete närvivõrkude komplekssem eritüüp on *Long-Short Term Memory* ehk **LSTM**. Need on mõeldud lahendama lihtsate RTN-ide puhul esinevat probleemi: treeningfaasi ajal plahvatavat või kaduvat gradienti, mis tähendab, et närvivõrk ei suuda pikaajaliselt andmetest sekventsiaalseid seoseid õppida. LSTM võrkudes on peitühikute struktuur muudetud mälurakkudeks, milles sisendi ja väljundi infovoogu kontrollivad väravad. Selline mälurakk on võimeline infot talletama nii, et gradient püsib konstantsena ja seega suudab närvivõrk infot pikema aja jooksul meeles hoida ja sellest õppida. Ka LSTM-id võivad esineda eelnevalt kirjeldatud kahesuunalisel kujul, sellisel juhul nimetatakse neid bidirektsionaalseteks LSTM-ideks ehk biLSTM-ideks. (ibid., 9–10)

LSTM-i tööpõhimõttega on sarnased väravatega rekurrentsed ühikud ehk *Gated Recurrent Units* ehk **GRU**. Ka nendes ühikutes on väravad, mis kontrollivad infovoogu, kuid puuduvad mälurakud. Nii LSTM kui ka GRU põhimõttel töötavatel närvivõrkudel on oma eelised kindlate ülesannete lahendamisel. (ibid., 12–13)

1.4.3 Seq2seq mudeli põhimõte ja selle komponendid

Kuigi rekurrentsed närvivõrgud suudavad infot sekventsiaalselt sisse võtta ja väljastada, toimib see vaid siis, kui sisendi ja väljundi vaheline joondumine on ette teada. Kui sisendi ja väljundi sekventsidel on erinevad pikkused, kasutatakse nende ühtlustamiseks spetsiaalseid võtteid. Sellel põhinebki *sequence to sequence* ehk ***seq2seq*** lähenemine. (Sutskever, Vinyals, Le 2014: 3)

Selleks kasutatakse mingit RTN-i tüüpi, näiteks üht LSTM-i, mis loeb sisendit ja teeb sellest vektoresituse. Seda kihti nimetatakse **kodeerijaks**. Teist LSTM-i kasutatakse, et muuta saadud vektor lõpuks väljundjadaks. See kiht on **dekodeerija**. (ibid., 2)

Seq2seq mudelitega kasutatakse tihti **tähelepanumehhanism**e. See on mõeldud lahendamaks dekodeerimisel esinevat probleemi, kus dekodeerija tähelepanu on hajutatud kogu sekventsi peale laiali, ehkki sekventsi mõned osad on tegelikult tähtsamad

ja teised vähem olulised või ebavajalikud. Tähelepanumehhanismid arvutavad iga sõna jaoks kontekstivektori, mis annab dekodeerimisel igale sõnale kaalu vastavalt tema olulisusele. Niimoodi saab fookuse suunata sekventsi kõige tähtsamatele osadele. (Hu 2018: 1)

2. TÖÖS KASUTATUD TEHNILINE MATERJAL

Käesolevas peatükis tutvustatakse töös kasutatud kolme tehisnärvivõrgupõhist lemmatiseerijat – TurkuNLP, Stanfordi ja LemmaTagi mudel, antakse ülevaade nende toimemehhanismist ja sisendiks vajalikke andmeid.

2.1 TurkuNLP

Turu ülikooli loomuliku keele töötluse töögrupis TurkuNLP loodud lemmatiseerimismudel tagastab lemma sõnavormi tähemärkide ja morfoloogiliste kategooriate põhjal (Kanerva, Ginter, Salakoski 2019: 1). See saavutas CoNLL 2018 Shared Task võistlusel lemmatiseerimise kategoorias esikoha (CoNLL 2018).

Kanerva ja teiste mudeli erineb varasematest lemmatiseerijatest selle poolest, et kasutab sisendina sõnavormi koos morfoloogiliste märgenditega, mitte näiteks liugakent, mis hõlmab sõnavormi koos kontekstiga (Kanerva, Ginter, Salakoski 2019: 1) Vajalik info sõnaliikide ja morfoloogiliste kategooriate kohta saadakse Stanfordi sõnaliigiennustajaga CoNLL-17 Shared Task võistluselt. Sellega luuakse esmalt lauses esinevatest sõnedest kontekstiga arvestavad vektoresitused, mille peal treenitakse koos kaht klassifitseerimiskihti – nii universaalsete kui ka keelespetsifiiliste sõnaliikide leidmiseks. Ennustaja tagastab seega nii universaalse kui keelespetsiifilise sõnaliigi. Kuna Turku mudeli sisend vajab lisaks sellele veel morfoloogilisi kategooriaid, näiteks käändeid või verbi tegumoodi, ent Stanfordi ennustaja neid esialgsel kujul ei tagastanud, muutsid Kanerva ja teised ennustaja treenimisandmeid niimoodi, et keelespetsiifiline sõnaliik pandi morfoloogiliste kategooriatega kokku üheks sõneks. Seejärel tagastas Stanfordi ennustaja lisaks mõlemat tüüpi sõnaliikidele ka morfoloogilised kategooriad, kaotamata seejuures täpsuses. (Kanerva, Ginter, Salakoski 2019: 6)

Sellisel kujul (sõnavorm koos morfosüntaktilise märgendusega) esitatud andmete korral võib lemmatiseerimiseks kasutada igasugu *seq2seq* mudeleid. Kanerva ja teised valisid aluseks olemasoleva neuromasintõlke mudeli, nimelt OpenNMT: Open-Source Toolkit for Neural Machine Translation. (Kanerva, Ginter, Salakoski 2019: 4; Klein jt 2017)

See mudel on tähelepanumehhanismiga kodeerija-dekodeerija tüüpi närvivõrk. Kodeerija kasutab õpitud tähemärgi- ja märgendivektoreid ja kaht biLSTM kihti, et muuta sisendtähemärkide ja morfoloogiliste märgendite jada ühepikkusteks kodeeritud vektorite jadadeks. Dekodeerija genereerib väljundmärgijada kahe ühesuunalise LSTM kihiga, kus lisaks kodeerija väljundile kasutatakse söödetakse mudelile sisendina ette tähelepanuvektoreid, mis annavad mudelile uute otsuste tegemisel infot eelnevalt tehtud otsuste kohta. (Kanerva, Ginter, Salakoski 2019: 4; Luong, Pham, Manning 2015)

TurkuNLP uurimisgrupi poolt loodud lemmatiseerija lähtekood on vabalt kättesaadav Githubis[5] ning installeerimis- ja kasutusjuhised on leitavad uurimisgrupi kodulehel[6]. Lemmatiseerijat saab kasutada nii olemasolevate, Universal Dependencies puupankade peal treenitud mudelite allalaadimiseks ja nende peal testimiseks, kui ka enda andmestike peal uute mudelite treenimiseks ja salvestamiseks. Viimaseks on tarvis treenimis- ja arendusandmeid CoNLL-U formaadis, mis peab sisaldama vähemalt sõnavormi, lemmat, universaalset sõnaliiki (UPOS), keelespetsiifilist sõnaliiki (XPOS) ja morfoloogilisi kategooriaid (FEATS). Lisaks on tarvis eeltreenitud sõnavektoreid word2vec formaadis ja teksti tükeldamiseks on tarvis eraldi treenida tükeldaja (*tokenizer*) UDPipe v1.2.0 tarkvaraga. (TurkuNLP; Straka, Straková 2017). Nii treenimine kui testimine toimub käsurealt.

2.2 StanfordNLP

Stanfordi Ülikooli teadlaste Qi, Dozati, Zhangi ja Manningu poolt CoNLL-2018 võistlusele esitatud süsteem võimaldab tekstide tükeldamist, lausete ja sõnade segmenteerimist, lauseliikmete ja morfoloogiliste kategooriate märgendamist, lemmatiseerimist ja sõltuvuste parsimist (Qi jt 2018: 1).

[5] https://github.com/TurkuNLP/Turku-neural-parser-pipeline

[6] https://turkunlp.org/Turku-neural-parser-pipeline/

Lemmatiseerimisega tegelev mudeli osa ehitab treeninghulga põhjal kaks sõnastikku: üks seob omavahel paari, mis koosneb sõnavormist ja universaalsest keelemärgendist (sõnavorm, UPOS), ning lemma. Teises on otse vastavusse viidud sõnavorm ja tema lemma. Lemmade ennustamisel kasutatakse kõigepealt esimest sõnastikku koos UPOSiga, seejärel teist. Kui nendest lemma ennustamiseks ei piisa, minnakse üle närvivõrgule. Närvivõrk antud süsteemis kujutab endast seq2seq mudelit BiLSTM kodeerijaga. Lisaks tarvitab mudel 3-osalist klassifitseerijat, mis ennustab, kas lemma on sõnaga identne, või sõna väiketähestatud versioon, või on tarvis seda põhjalikumalt muuta mudeli abil. Treenimise ajal antakse igale sõna-lemma paarile vastav silt ning treenitakse klassifitseerijat koos seq-to-seq mudeliga. Ennustusi tehakse osade kaupa, see tähendab, et kõigepealt otsustab klassifitseerija, kas antud sõna puhul on võimalik kasutada mõnd esimesest kahest lihtsamast töötlusvariandist, seejärel kasutatakse vajaduse korral neuromudelit. (Qi jt 2018: 2–4) Stanfordi mudelit on võimalik kasutada Pythoni teegina, mis koosneb nii siinkirjeldatud mudelist. Enda andmete põhjal uute mudelite treenimiseks on tarvis kood kloonida ning kasutada treenimiseks vajalikke skripte[7]. Lähtekood on Githubis vabalt kättesaadav[8].

Ka Stanfordi lemmatiseerimismudel vajab sisendina ConLL-U formaadis andmeid koos sõna algvormi, lemma, universaalsete ning keelespetsiifiliste sõnaliikidega. Hindamine toimub pärast treenimist automaatselt, kuid hiljem on võimalik väljundfaili kasutada ka muu hindamisskripti sisendina.

2.2 LemmaTag

LemmaTag on Karli Ülikoolis välja töötatud neuroarhitektuur, mis genereerib igale sisendis esinevale sõnale koos nii lemma kui ka sõnaliigi. LemmaTag on näidanud väga häid tulemusi morfoloogiliselt rikastel keeltel, näiteks tšehhi, araabia ja saksa keel (Kondratyuk jt 2018: 1).

[7] https://stanfordnlp.github.io/stanfordnlp/training.html

[8] https://github.com/stanfordnlp/stanfordnlp

Mudel koosneb kolmest osast: esiteks kodeerijast, mis loob igale sõnale süsteemisisese esituse vastavalt sõna tähemärkidele ja kontekstile. See osa on nii lemmatiseerimiseks kui sõnaliikide märgendamiseks ühine. Teiseks on mudelis märgendaja-dekodeerija, mis eelmise kihi väljundi põhjal sõnaliike ennustab, ja kolmandaks lemmatiseerija-dekodeerija, mis kasutab kahe eelmainitud kihi väljundit, et ennustada sõnale vastav lemma. Selle süsteemi eeliseks on, et teksti pole vaja eeltöötleda või spetsiaalsele kujule viia, ning mõlemad dekodeerijad kasutavad samu parameetreid. (Kondratyuk jt 2018: 1–2)

LemmaTagi kodeerija moodustab iga sõna igast tähemärgist kindla pikkusega vektori, mis antakse edasi GRU-dest koosnevale kahesuunaliste rekurrentsete närvivõrkude (BRTN) kihile, kus väljundvektorid ühendatakse ja neist moodustatakse antud sõna tähemärgitasemel vektoresitus. Lisaks arvutatakse sõnast sõnatasemel vektor, mis koos eelnevaga moodustabki sõna lõpliku vektoresituse. See lõplik esitus antakse järgmisele BRTN kihile, mille väljund on sõnaesitus, mis sisaldab ka informatsiooni iga sõna kontekstist lauses. (Kondratyuk jt 2018: 2)

Märgendaja-dekodeerija saab sisendiks eelmainitud sõnaesituse ning ennustab selle jaoks märgendi suurima tõepära meetodil. Kui märgendil on alamkategooriaid, arvutatakse ka nende tõepära ning kogu arvutatud info viiakse üle vektorkujule, mis on sisendiks lemmatiseerijale. (Kondratyuk jt 2018: 2–3)

Lemmatiseerija koosneb rekurrentsest LSTM kihist, mille algseisund tuleb sõnaesitusest ning mille sisend koosneb kolmest osast. Esiteks on sisendiks eelneva väljundtähemärgi vektoresitus. Teiseks sisendiks on Luong, Pham, Manning 2015 loodud tähelepanumehhanism, mis arvutab tähelepanuvektori, kaaludes tähemärgitasemel informatsiooni LSTM kihi seisundi põhjal. Kolmandaks sisendiks on informatsioon sõnaesituse, lausepõhise konteksti ja märgendaja väljundi kohta. (Kondratyuk jt 2018: 3)

LemmaTagi lähtekood on vabalt kättesaadav Githubis[9]. Selle kasutamiseks on vaja programm alla laadida, paigaldada vajalikud teegid ning treenida mudelit käsurealt. Andmed peavad olema kas CoNLL-U kujul või LemmaTag formaadis. Viimases on 3 tabulaatoriga eraldatud tulpa: sõnavorm, lemma ja sõnaliik. Lausepiire eristab tühi rida.

[9] https://github.com/hyperparticle/LemmaTag

3. KASUTATUD KORPUSMATERJAL

Käesolevas töös on mudelite treenimiseks ja testimiseks kasutatud 3 eri korpusallikat: Universal Dependencies eesti keele puupank, morfoloogiliselt ühestatud eesti keele korpus, Estonian Web Treebank korpus. Selles peatükis antakse igast korpusest ülevaade, sealhulgas nende suurusest, nendes esinevate sõnaliikide sagedusest ja korpuste teisendamisest lemmatiseerimismudelitele sobivasse formaati. Korpusmaterjal tuleb tükeldada närvivõrkudele vajalikku kolme osasse: treeninghulgaks, arendushulgaks ja testhulgaks, niisiis kirjeldatakse siin peatükis ka selle jaotamist.

3.1 Universal Dependencies

Universal Dependencies (edaspidi UD) on projekt, mille käigus arendatakse keelteüleselt ühtse märgendusega puupankasid. Puupangad on vabalt kättesaadavad Universal Dependencies kodulehelt kümnetes keeltes, sealhulgas eesti keeles. (Nivre jt 2018) Käesolevas töös on kasutatud UD versiooni 2.2 ning UD versiooni 2.3 eesti keele puupankasid.

3.1.1 Universal Dependencies puupankade formaat

Puupankade vormistamiseks kasutatakse formaati nimega CoNLL-U[10], mis põhineb iga-aastase Conference on Computational Natural Language Learning (CoNLL) võistlusel kasutamiseks mõeldud andmeformaadil CONLL-X (Buchholz, Marsi: 2006). Selles formaadis on iga lause sõnad tõstetud eraldi ridadele ja lausepiire märgivad tühjad read.

Igal real, mis sisaldab sõna, on kümme tabulaatoriga eraldatud välja, millest käesolevas töös on vajalikud esimesed kuus:

1. ID: sõna indeks, mis iga uue lause puhul algab 1-st

[10] https://universaldependencies.org/format.html

2. FORM: lauses esinenud sõnavorm või punktuatsioonimärk
3. LEMMA: sõnavormi lemma
4. UPOS: universaalne sõnaliik
5. XPOS: keelespetsiifiline sõnaliik
6. FEATS: morfoloogilised kategooriad

Sellesse formaati tuli teisendada ka muud käesolevas töös kasutatud andmed, kuna CoNLL võistluseks ehitatud süsteemid aktsepteerivad just selles formaadis sisendit.

UPOS ehk universaalsed sõnaliigid on kasutusel kõigis UD puupankades keeleüleselt. Nende kirjeldus on kättesaadav UD kodulehel.[11]

Morfoloogilised kategooriad annavad täiendavat infot sõnade grammatiliste ja leksikaalsete omaduste kohta. Täpse ülevaate kõikidest UD-s kasutatavatest kategooriatest on antud nende veebilehel.[12]

3.1.2 Töös kasutatud Universal Dependencies puupankade versioonide ülevaade

Selles töös on mudelite treenimiseks kasutatud eesti keele UD versiooni 2.2 treening- ja arendushulka ning testimiseks eesti keele UD versiooni 2.3 testhulka. Mõlema puupanga andmed pärinevad eesti keele sõltuvuspuude pangast ja HamleDT 3.0 puupanga eestikeelsest osast ning sisaldavad 3 žanrit: ajakirjandus, teaduskirjandus ja ilukirjandus.[13] UD puupangad on juba jaotatud treening-, arendus- ja testhulkadeks.

Treenimiseks valiti UD 2.2 versioon, kuna selles oli kahe mudeli, TurkuNLP ja Stanfordi puhul võimalik kasutada autorite eeltreenitud versiooni, ning testimiseks UD 2.3

11 https://universaldependencies.org/u/pos/index.html

12 https://universaldependencies.org/u/feat/index.html

13 https://github.com/UniversalDependencies/UD_Estonian-EDT/blob/be5cb38b79682c90d27ff3ae5ef631801edb7e26/README.md

versiooni testhulk, kuna selles peaks sisalduma pisut rohkem materjali ning seega on võimalik paremini hinnata, kui hästi tuleb mudel toime seninägemata andmetega.

UD versioonis 2.2 on lausete ja sõnede jaotus tervikuna (kõigi hulkade peale kokku) järgmine[14]:

- ilukirjandustekstid: 5522 lauset, 67 741 sõne
- ajakirjandustekstid: 14 470 lauset, 205 177 sõne
- teadustekstid: 4928 lauset, 84 233 sõne
- HamleDT 3.0: 9200 sõne

UD versioonis 2.3 on lausete ja sõnede jaotus tervikuna (kõigi hulkade peale kokku) järgmine[15]:

- ilukirjandustekstid: 5522 lauset, 67 774 sõne
- ajakirjandustekstid: 18 411 lauset, 263 279 sõne
- teadustekstid: 5483 lauset, 94 022 sõne
- HamleDT 3.0: 1277 lauset, 9200 sõne

Tabelis 1 on toodud nii UD 2.2 versioonil põhinevate treening- ja arendushulga kui ka UD 2.3 versioonil põhineva testhulga suuruse info: sõnede ja lausete arv ning unikaalsete sõnede ja lemmade hulk. Testhulk on arendushulgast mõnevõrra suurem, tingituna sellest, et UD 2.3 versiooni on lisatud eelmise versiooniga võrreldes rohkem materjali.

[14]https://github.com/UniversalDependencies/UD_Estonian-EDT/blob/12bc71c07fc7ac64010bf73eea8f273fcaba57b4/README.md

[15] https://github.com/UniversalDependencies/UD_Estonian-EDT/blob/be5cb38b79682c90d27ff3ae5ef631801edb7e26/README.md

Tabel 1. **UD 2.2 treening- ja arendushulgas ning UD 2.3 testhulgas esinevate sõnede ja lausete sagedusinfo.**

Korpus	**UD 2.2**		**UD 2.3**	**Kokku**
Hulga tüüp	Treeninghulk	Arendushulk	Testhulk	
Sõnede arv	287 859	37 219	48 491	373 569
Lausete arv	20 827	2633	3214	26 674
Unikaalsete sõnede arv	41 908	10 042	11 199	63 149
Unikaalsete sõnede arv väiketähestades	37 844	9310	10 420	57 574
Unikaalsete lemmade arv	17 786	5021	5565	28 372
Unikaalsete lemmade arv väiketähestades	17 321	4967	5457	27 745

Tabelis 2 on antud UD 2.2 versiooni treening- ja arendushulgas ning UD 2.3 testhulgas esinevate universaalsete sõnaliikide sagedused eestikeelses puupangas. Kõige sagedasemad sõnaliigid on mõistagi nimisõnad, verbid, adverbid ja adjektiivid, lisaks ka punktuatsioon. Sõnaliikide jaotus arendus- ja testhulgas on küllalt sarnane.

Tabel 2. **UD versiooni 2.2 treening- ja arendushulgas ning UD 2.3 testhulgas esinevate sõnaliikide sagedused**

Korpus	**UD 2.2**		**UD 2.3**	
Sõnaliik	**Sagedus treeninghulgas**	**Sagedus arendushulgas**	**Sagedus testhulgas**	**Kokku**
ADJ	24 071	3317	4075	31 463
ADP	6325	761	910	7996
ADV	27 910	3535	4859	36 304
AUX	14 663	1881	2576	19 120
CCONJ	10 425	1359	2022	13 806
DET	4541	557	814	5912
INTJ	247	21	50	318
NOUN	75 892	9845	12 633	98 370
NUM	5688	947	858	7493
PRON	15 308	1976	2520	19 804
PROPN	16 989	1935	3064	21 988
PUNCT	42 786	5875	7207	55 868
SCONJ	5692	711	1123	7526
SYM	107	37	27	171
VERB	32 461	4070	5216	41 747
X	284	22	70	376
Z	4470	370	467	5307

3.2 Morfoloogiliselt ühestatud korpus

Morfoloogiliselt ühestatud korpus koosneb morfoloogiliselt märgendatud ja ühestatud eesti kirjakeelsetest tekstidest, täpsemalt

- George Orwelli romaan „1984“,
- eesti kirjakeele korpusest pärinevad ilukirjandustekstid eesti autoritelt,
- ajakirjandustekstid aastatest 1995-1999,
- seadustekstid,

- tekstid ajakirjast „Horisont“ aastatest 1996-2003 ning
- infotekstid eesti kirjakeele korpusest (teemadel hobid-harrastused ja entsüklopeediad).

Korpuse failid, v.a suulise kõne tekstid, on allalaadimiseks vabalt kättesaadavad korpuse kodulehel.[16] Allalaetud kaustas on 128 faili, mille nimi vastab žanrile. Tekstifailides on info esitatud järgneval kujul:

sõna tüvi+lõpp // analüüs //

kus sõna on tekstis algselt esinenud sõnavorm, tüvi on lemma (tegusõnade puhul on ma-infinitiiv antud ilma ma-lõputa) ja lõpp on sõna lõpp, sh liitunud mitmuse tunnusega ning partiklika -ki/-gi. Kui sõnal lõppu ei ole, lisatakse sellele null-lõpp (+0). Analüüsis on antud morfoloogilised kategooriad vastavas tabelis esineval kujul, sisaldades muuhulgas keelespetsiifilist sõnaliiki. (MÜK)

Kuna töös kasutatud neuromudelid nõuavad sisendiks teistsuguses, CoNLL-U formaadis faile, oli tarvis morfoloogiliselt ühestatud korpuse morfoloogilised andmed vastavale kujule viia.

3.2.1 Morfoloogiliselt ühestatud korpuse teisendamine CoNLL-U formaati

CoNLL-U formaati teisendamiseks tõsteti esmalt kõik tekstifailid kokku. Esmalt teisendati HTML-olemid *unicode* kujule vastavalt olemite tabelile[17]. Keelespetsiifilistelt sõnaliikidelt eemaldati ümbert alakriipsud ning kaldkriipsud. Universaalne sõnaliik määrati vastavalt keelespetsiifilisele sõnaliigile[18] ja lisati tabulaatoritega eraldatuna lemma ning keelespetsiifilise sõnaliigi vahele. Lemmadest eemaldati plussmärgid, et viia kokku sõnatüvi ja -lõpp, ning võrdusmärgid, sest nendega eraldatud sufiksite märkimine

[16] https://www.cl.ut.ee/korpused/morfkorpus/

[17] https://www.cl.ut.ee/abi/olemid/

[18] https://universaldependencies.org/u/pos/index.html

ei ole korpuse kodulehe andmetel järjekindel (ibid.). Morfoloogiliste kategooriate teisendamiseks kasutati Pythoni skripti koos CSV-formaadis tabeliga, milles CoNLL-U morfoloogilised kategooriad olid viidud vastavusse morfoloogiliselt ühestatud korpuse kategooriatega. Tabeli algkuju pärineb Tkachenkolt ja Sirtsilt (Tkachenko, Sirts 2018), seda kohendati käesoleva uurimuse tarbeks.

3.2.2 Morfoloogiliselt ühestatud korpuse jaotamine treening-, arendus- ja testhulgaks

Morfoloogiliselt ühestatud korpuse jaotamisel hulkadeks arvestati sellega, et eri žanrid oleksid igas hulgas esindatud. Selleks oli vaja korpusfail, kus kõik eri žanrist tekstifailid olid kokku tõstetud ja CoNLL-U formaati teisendatud, segada läbi lausepiire arvestades. Vastasel juhul oleks näiteks treeningfail sisaldanud kogu romaani „1984“, aga ei ühtegi juriidilist ega infotekst. Segatud fail jagati treeninghulgaks, mille suurus oli 70% esialgsest failist, arendushulgaks (suurus 15%) ja testhulgaks (suurus 15%). Nii segamiseks kui jagamiseks kasutati Pythoni skripte.

3.2.3 Morfoloogiliselt ühestatud korpuse suuruse ning sõnaliikide info

Tabelis 3 on toodud morfoloogiliselt ühestatud korpuse treening-, arendus- ja testhulga suuruse info sõnede ja lausete ning unikaalsete sõnede ja lemmade kaupa. Korpuses on umbes veerand miljoni jagu rohkem sõnu kui UD kahest versioonist pärit treening-, arendus- ja testhulgas kokku.

Tabel 3. **Morfoloogiliselt ühestatud korpuse treening-, arendus- ja testhulgas esinevate sõnede ja lausete ülevaade.**

Hulk	**Treeninghulk**	**Arendushulk**	**Testhulk**	**Kokku**
Sõnede arv	437 011	93 721	93 772	624 504
Lausete arv	29 684	6281	6234	42 199
Unikaalsete sõnede arv	48 846	19 424	19 329	87 599

Unikaalsete sõnede arv väiketähestades	44 082	17 968	17 928	79 978
Unikaalsete lemmade arv	19 596	9159	9120	37 875
Unikaalsete lemmade arv väiketähestades	19 181	8989	8959	37 129

Tabelis 4 on toodud morfoloogiliselt ühestatud korpuses esinevate universaalsete sõnaliikide sagedus nii treening-, arendus- ja testhulgas eraldi kui ka kogu korpusmaterjali peale kokku. Sagedasemad sõnaliigid on nimisõnad, verbid, adverbid, pronoomenid ja adjektiivid ning punktuatsioon. Ilmne on ka, et morfoloogiliselt ühestatud korpus on UD versioonidest suurem.

Tabel 4. **Sõnaliikide sagedus morfoloogilises korpuses**

Sõnaliik	**Sagedus treeninghulgas**	**Sagedus arendushulgas**	**Sagedus testhulgas**	**Kokku**
ADJ	32 101	6949	6995	46 045
ADP	9260	1957	1937	13 154
ADV	35 815	7604	7887	51 306
AUX	9181	1923	1980	13 084
CCONJ	17 876	3946	3873	25 695
INTJ	226	45	49	320
NOUN	118 557	25 722	25 420	169 699
NUM	18164	3846	3904	25 914
PRON	34186	7411	7261	48 858
PROPN	16 390	3581	3400	23 371
PUNCT	72 153	15 173	15 356	102 682
SCONJ	9165	1876	2034	13 075
VERB	58 426	12 498	12 551	83 475
X	473	108	117	698
Y	5038	1082	1008	7128

3.3 Estonian Web Treebank

Estonian Web Treebank (lühend EWTB) on osa EtTenTeni korpusest, andmed pärinevad Särgi, Muischneki ja Müürisepa 2018. aasta publikatsioonist, ehkki korpus oli artikli avaldamise hetkel väiksem (Särg, Muischnek, Müürisep 2018).

EWTB korpust kasutati ainult olemasolevate mudelite testimiseks, sest selle mahust ei piisaks uute mudelite treenimiseks. Tabelis 5 on toodud EWTB korpuse suuruse info lausete ja sõnede ning unikaalsete sõnavormide ja lemmade järgi. Korpus on tõepoolest väike, lauseid on vaid 1662 ning sõnesid veidi üle 27 000, millest unikaalseid on umbes 6500 jagu.

Tabel 5. **EWTB korpuse lausete ja sõnede sageduse ülevaade.**

Sõnede arv	27 153
Lausete arv	1662
Unikaalsete sõnede arv	6547
Unikaalsete sõnede arv väiketähestades	6079
Unikaalsete lemmade arv	3204
Unikaalsete lemmade arv väiketähestades	3148

Tabelis 6 on antud ülevaade EWTB korpuses esinevate sõnaliikide sagedusest. Sõnaliike on loetud Universal Dependencies universaalsete sõnaliigikategooriate järgi. Andmestikus esines üks L-iga märgendatud sõna (*olla* sõnast *võib olla*), mida kategooriate teisendamisel ei tuvastatud ja mis jäi lõppandmetesse sisse.

Tabel 6. **Estonian Web Treebank korpuses esinevate sõnaliikide sagedus.**

Sõnaliigi lühend	**Sagedus**
ADJ	1657
ADP	513
ADV	3373

AUX	955
CCONJ	1128
INTJ	108
L	1
NOUN	5582
NUM	330
PRON	2978
PROPN	858
PUNCT	4150
SCONJ	919
SYM	14
VERB	4286
X	301

3.3.1 EWTB teisendamine CoNLL-U formaati

Ka EWTB korpus tuli teisendada mudelite jaoks sobivale CoNNL-U kujule. Universaalse sõnaliigi saamiseks teisendati korpuses olemasolnud eestikeelne sõnaliik vastavaks universaalseks variandiks. Mõningad märgendid olid sellised, mida ei esinenud morfoloogiliselt ühestatud korpuses, nt märgend *B*, mille universaalseks vasteks valiti märgendi esinemise põhjal *INTJ* ehk interjektsioon. Morfoloogiliste kategooriate teisendamiseks kasutati eelmainitud tabelit (Tkachenko, Sirts 2018), mida antud korpuse jaoks täiendati.

Kategooriate teisendamisel esines probleem universaalse sõnaliigi DET (determineerija) märgendamisega. Korpuse algmaterjalis oli osade pronoomenite morfoloogiliste kategooriate alla lisatud märge *det*. Ka CoNLL-U formaadis on olemas sõnaliik DET, kuid selle kasutus eestikeelses puupangas erines mõneti EWTB märgendusest. Seetõttu

kasutati EWTB lemmatiseerimise testimisel kahte varianti: ühes oli algmaterjali kategooria *det* teisendatud CoNLL-U formaadi universaalseks sõnaliigiks DET; teises variandis oli kategooria *det* jäetud teisendamata. See ei andnud aga lemmatiseerimisel mingit olulist vahet: vaid Stanfordi mudelil paranes *det*-kategooriat teisendamata jättes täpsus 0,02% võrra. Niisiis on lõpptulemustes esitatud vaid selline EWTB versioon, kus morfoloogiline kategooria *det* on jäetud teisendamata, ehk siis korpuses ei leidu universaalset sõnaliiki DET.

4. EKSPERIMENTAALSED TULEMUSED JA NENDE ANALÜÜS

Käesolevas, viimases peatükis kirjeldatakse töös kasutatud kolme mudeli ning Vabamorfi tarkvaraga erinevatel korpustel tehtud katseid, antakse ülevaade lemmatiseerimise täpsustest ja analüüsitakse põhilisi vigu.

4.1 Treenimise ja testimise kirjeldus

Mudelite treenimiseks ja testimiseks on tarvis andmetest luua 3 osa: treenimis-, arendus- ja testhulk. Esimesed kaks on vajalikud mudeli treenimiseks soovitud andmetel, viimane aga olemasoleva mudeli testimiseks. Universal Dependencies puhul olid failid juba treenimis-, arendus- ja testhulgaks jagatud. Morfoloogiliselt ühestatud korpuse jaotamist hulkadeks on kirjeldatud 3. peatükis. EWTB korpust kasutati mahu tõttu vaid testimiseks.

Nii TurkuNLP kui ka Stanfordi mudeli puhul kasutati mudeli autorite poolt eeltreenitud ja allalaetavat Universal Dependencies 2.2 peal treenitud mudelit. LemmaTagi puhul treeniti mudel vastavalt Githubis toodud juhendile samuti valmis Universal Dependencies 2.2 puupanga peal. Treenimisel etteantud parameetreid ei muudetud. Morfoloogiliselt ühestatud korpusel põhineva mudeli treenimiseks tuli kõigile kolmele mudelile anda ette vastaval korpusel põhinev treening- ja arendushulk. Selleks kasutati mudelite kodulehel ja Githubis kättesaadavaid juhiseid. Treenimisel etteantud parameetreid ei muudetud.

Mudelite täpsuse testimiseks kasutati Universal Dependencies 2.3 testhulka, morfoloogiliselt ühestatud korpusel põhinevat testhulka ja EWTB korpust.

Lisaks töös kirjeldatud kolmele mudelile katsetati testhulkasid ka Vabamorfi tarkvarale, et võrrelda selle reeglipõhise lemmatiseerija täpsust närvivõrkudel põhinevate lemmatiseerijatega. Vabamorfi kasutati EstNLTK teegi versiooni 1.4.1 kaudu.

Mudelite täpsuse hindamisel ei arvestatud järelliidet eraldavat võrdusmärgi sümbolit, sest see esines treening- ja testfailides ebaühtlaselt. Kui lemma on sisuliselt õige, ei peetud vajalikuks produktiivsite sufiksite märkimist võrdusmärgiga täpsuse hindamisel arvesse võtta.

4.4 Vabamorfi väljund

Võrdlemaks tehisnärvivõrkudel põhinevate lemmatiseerijate täpsust reeglipõhiste vahenditega, anti EstNLTK teegi kaudu Vabamorfile ette nii Universal Dependencies 2.3, morfoloogiliselt ühestatud korpuse kui ka EWTB testfailide sõnavormid ja arvutati selle täpsus. Täpsuse arvutamisel ei arvestatud alakriipsu ega võrdusmärki, mida korpuste lemmades esineb, kuid Vabamorfi tagastatavates lemmades mitte. Kuna Vabamorf jätab mõningad analüüsid mitmeseks, hinnati väljundis eraldi üheste lemmade täpsust, lisaks kogu korpuse täpsust, kui mitmeselt analüüsitud variantidest valiti esimene, ning seda, kui paljudel mitmeselt analüüsitud sõnadel oli õige vaste variantide seas olemas.

Üheselt analüüsitud sõnade täpsus on toodud tabelis 7. Siit ilmneb, et Vabamorf toimis nii väiketähestades kui väiketähti arvestamata kõige paremini morfoloogiliselt ühestatud korpusel, saades mõlemal juhul tulemuseks üle 98%. EWTB materjali lemmatiseerimine jääb täpsuselt teistele kuni paari protsendipunkti võrra alla, 95% ja 96% vahele. UD 2.3 peal on tulemused pisut paremad kui EWTB korpusel, ületades kolmandiku jagu 96% piiri, väiketähestades on täpsus üle 97%. Väiketähestamine parandab morfoloogiliselt ühestatud korpusel saadud tulemust väga väiksel määral, alla 0,2%, kuid UD 2.3 puhul tõstis see täpsust peaaegu 0,9% võrra ja EWTB korpusel umbes poole protsendi jagu.

Tabel 7. **Vabamorfi täpsus etteantud testkorpustel %-des.**

Testkorpus	**Üheste sõnade täpsus**	**Üheste sõnade täpsus väiketähestades**
UD 2.3	96.31	97.19
Morfoloogiliselt ühestatud korpus	98.10	98.27
EWTB	95.31	95.89

Mitmeselt analüüsitud sõnade arvestamiseks lisaks ühestele lemmadele hinnati kogu korpuse analüüsi täpsust, kui mitmestest variantidest oli valitud esimene. Need tulemused on toodud allpool tabelis 8.

Esimese variandi valimine mitmestes analüüsides annab küllaltki head tulemused, vähemalt 94,38% ulatuses on väiketähestamata lemmad õiged. Kõige täpsemini analüüsib Vabamorf morfoloogiliselt ühestatud korpust, saades peaaegu 97% jagu õigeid lemmasid. Täpsus UD 2.3 testhulgal jääb 95% kuni ligi 96% vahele, halvim on tulemus EWTB korpusel, kus see ei ületa isegi väiketähestades 95%.

Tabel 8. **Vabamorfiga analüüsitud testkorpuste täpsus %-des, kui mitmestest analüüsidest valiti esimene variant.**

Testkorpus	**Täpsus väiketähestamata**	**Täpsus väiketähestades**
UD 2.3	95.17	96.06
Morfoloogiliselt ühestatud korpus	96.61	96.79
EWTB	94.38	94.98

Viimaks arvutati, kui paljudel mitmeselt analüüsitud sõnadel oli õige lemma analüüsivariantide seas üldse olemas. Need osakaalud on toodud tabelis 9. Siit ilmneb, et Vabamorf leiab väga hea täpsusega üles õige lemma, kuigi ei või teada, mitmendana see mitmeses analüüsis esineb. Tulemus on kõige parem morfoloogiliselt ühestatud korpusel, kus väiketähestamata on täpsus peaaegu 97,5% ja väiketähestades oli koguni peaaegu 99% mitmestest analüüsidest õige lemma variantide seas. Väiketähestades ei jää palju alla ka tulemus UD 2.3 testhulgal, kus õige lemma oli olemas ligi 98% juhtudest, väiketähestamata aga oli tulemus 2% võrra madalam. Kõige madalam tulemus oli jälle EWTB korpusel, kus mitmeste analüüside seas esines õige lemmavariant väiketähestamata vaid umbes 93% juhtudest, väiketähestades aga pooleteise protsendi võrra rohkem.

Tabel 9. **Vabamorfiga testkorpusi lemmatiseerides saadud mitmeste lemmaanalüüside ükskõik mitmenda variandi õigeks osutumise osakaal %-des kõigist mitmestest analüüsidest.**

Testkorpus	**Osakaal väiketähestamata**	**Osakaal väiketähestades**
UD 2.3	95.62	97.92
Morfoloogiliselt ühestatud korpus	97.45	98.86
EWTB	93.37	94.92

4.2 Tehisnärvivõrkudel põhinevate lemmatiseerijate tulemused

Töös kasutatud kolme mudeli täpsused kõigis kolmes testkorpuses on toodud tabelis 10. Tabelis on toodud nii UD 2.2 puupangal kui ka morfoloogiliselt ühestatud korpusel treenitud mudelite täpsused testkorpuste kaupa. Täpsust arvutati kolmel viisil: esiteks esialgsel kujul (tabelis iga mudeli nimele vastav lahter), teiseks väiketähestatud kujul (tabelis iga mudeli juures teine rida märkega *väiketähestatud*), kolmandaks väiketähestatud kujul, kus lisaks oli lemmadest eemaldatud sidekriips ja alakriips (tabelis iga mudeli juures kolmas rida märkega *lisamärkideta*).

Väiketähestatud lemmade täpsus on arvutatud eraldi, sest esialgsete tulemuste vaatlemine näitas, et mudelid analüüsivad lemma küll sisuliselt õigesti, ent jätavad sellele mõnikord suure algustähe ka siis, kui seda vaja poleks, näiteks õigele lemmale *väike* asemel on mudel pakkunud *Väike*.

Pöördel: Tabel 10. **Töös kasutatud mudelite tulemused kolmel testkorpusel esialgsel, väiketähestatud ja väiketähestatud ning ilma side- ja alakriipsudeta kujul.**

UD 2.2 puupangal treenitud mudelid			
Testhulk / Mudel	UD 2.3 testhulk	Morfoloogiliselt ühestatud korpuse testhulk	EWBT testhulk
TurkuNLP	96.84	96.55	95.48
- väiketähestatuna	97.69	97.25	96.36
- lisamärkideta	98.20	97.7	96.84
Stanford	95.37	94.88	94.25
- väiketähestatuna	96.17	95.66	95.01
- lisamärkideta	96.95	96.45	95.61
LemmaTag	94.71	93.15	93.11
- väiketähestatuna	95.05	94.04	93.71
- lisamärkideta	96.37	95.38	94.47
Morfoloogiliselt ühestatud korpusel treenitud mudelid			
Testhulk / Mudel	UD 2.3 testhulk	Morfoloogiliselt ühestatud korpuse testhulk	EWBT testhulk
TurkuNLP	95.87	97.14	94.81
- väiketähestatuna	96.88	97.56	95.69
- lisamärkideta	97.56	97.81	96.11
Stanford	95.24	97.45	94.32
- väiketähestatuna	96.11	97.77	95.04
- lisamärkideta	96.98	98.12	95.57
LemmaTag	93.61	96.74	93.09
- väiketähestatuna	94.32	97.07	93.70
- lisamärkideta	95.58	97.69	94.46

Kuna treening-ja testhulkades oli märgendatud ka liitsõnade osi eraldav alakriips ja sidekriipsu esines eraldiseisvalt, näiteks leidus testkorpustes sõna ’--’, mida lemmatiseeriti ühe sidekriipsuna ’-’, otsustasti arvutada täpsust ka pärast nende faktorite eemaldamist ja lemmade väiketähestamist. Nii näeb, kas mudelid oskavad liitsõnu tegelikult õigesti lemmatiseerida, isegi kui liitsõna komponentide vahelist piiri täpselt määrata ei suudeta.

4.2.1 UD 2.2 versiooni puupangal treenitud mudelite tulemused

Tabelist 10 ilmneb, et UD 2.2 puupangal treenitud mudelitest saavutas kõige parema tulemuse iga testhulgaga TurkuNLP mudel, mille täpsus jäi minimaalselt ~95.5% ja maksimaalselt 96.84% juurde. Teiseks jäi Stanfordi mudel, mis oli igas testhulgas TurkuNLP mudelist enam kui protsendi võrra ebatäpsem. Kõige madalama täpsusega oli LemmaTagi analüüs, mis jäi TurkuNLP-le alla kuni 3.5%-ga. Kõik mudelid saavutasid aga parima tulemuse UD 2.3 testhulgal ning halvima EWTB korpusel.

Väiketähestamine kasvatas täpsusi kuni umbes protsendi võrra, paremusjärjestus jäi endiseks. Pärast ala- ja sidekriipsu eemaldamist kasvas kõigi tulemuste täpsus vahemikus 0,48% kuni 1,34%. Kõige rohkem paranes LemmaTagi mudeli täpsus, kõige vähem muutus TurkuNLP oma. Sellest võib oletada, et TurkuNLP mudel suudab liitsõna osi kõige täpsemini eristada, LemmaTagil esineb selles aga teistest rohkem probleeeme.

4.2.2 Tulemused morfoloogiliselt ühestatud korpuse peal treenitud mudelitel

Morfoloogiliselt ühestatud korpusel treenitud mudelite tulemused testhulkades on toodud tabelis X. Võrreldes UD 2.2 treenitud korpusega, on tulemused pisut ühtlasemad: TurkuNLP saavutas parima tulemus UD 2.3 testhulgal ja EWTB korpusel, Stanfordi mudel aga morfoloogiliselt ühestatud korpuse testhulgal. Taas saavutasid kõik mudelid oma parima tulemuse just morfoloogiliselt ühestatud korpuse testhulgal, selle suurim täpsus on 97,45%. Kõige madalama tulemuse igas testhulgas sai LemmaTag: kui võrrelda seda UD 2.2 peal treenitud LemmaTagi tulemustega, on täpsus halvem nii UD 2.3

testhulgas kui ka EWTB korpuses. See-eest on ka LemmaTag saanud morfoloogiliselt ühestatud korpuse testhulgal suurepärase tulemuse 96,74% täpsusega.

Väiketähestades paranesid jälle kõik tulemused maksimaalselt 1.01% võrra, kuid paremusjärjestus iga testhulga korral jäi ikkagi samaks. Ka side- ja alakriipsude eemaldamisel väiketähestatud lemmadest paranesid kõik tulemused, ilma et paremusjärjestus muutuks.

4.2.3 Mudelite täpsuse võrdlemine

Kui vaadelda EWTB lemmatiseerimise täpsust, on nii TurkuNLP kui LemmaTagi mudeli puhul parem UD 2.2 peal treenitud versioon; Stanfordi mudelil on keskmiselt võttes morfoloogiliselt ühestatud korpusel treenitud variant väga väiksel määral täpsem.

Morfoloogiliselt ühestatud korpuse testkorpust lemmatiseerivad iga mudeli puhul mitme protsendi võrra paremini morfoloogiliselt ühestatud korpusel treenitud variandid.

UD 2.3 testhulga lemmatiseerimisel on kõigi mudelite puhul parem UD 2.2 peal treenitud variant, ehkki Stanfordi mudelil saavutab morfoloogiliselt ühestatud korpusel treenitud variant lisamärkideta arvestuses 0,03% võrra kõrgema tulemuse.

UD 2.3 testhulgas on UD 2.2 peal treenitud mudelid Vabamorfist üldiselt täpsemad, morfoloogiliselt ühestatud korpusel treenitud mudelitest jääb Vabamorfile alla LemmaTag. Morfoloogiliselt ühestatud testkorpuse lemmatiseerimisel on UD 2.2 peal treenitud mudelitest Vabamorfi väljundist parem vaid TurkuNLP, morfoloogiliselt ühestatud korpusel treenitud mudelid aga ületavad kõik Vabamorfi täpsust. EWTB puhul on TurkuNLP ja Stanfordi mudelis mõlemas treeningvariandis täpsemad kui Vabamorfi väljund, LemmaTag aga halvema tulemusega.

4.3 Analüüs vigadest ja probleemkohtadest

Järgnevates alapeatükkides on esitatud mudelitel saadud tulemuste põhivigade ja probleemkohtade ülevaade ja analüüs.

4.3.1 Analüüs Vabamorfi tulemustest

Vabamorf esitab mineviku kesksõnade vasteks nii kesksõna enda kui ka ma-infinitiivi, ehkki korpustes on õigeks lemmaks märgitud ainult ma-infinitiiv või ainult nud/tud-partitsiip. Näites 2 on toodud variandid sellistest kesksõnadest ja infinitiividest.

(2)	**Vabamorfi väljund**	**Õige lemma**
	toimuma|toimunud	*toimunud*
	esitama|esitanud	*esitama*

Pärisnimede puhul väljastas Vabamorf nime mitmes erinevas käändes. Eriti probleemsed olid võõrapärased nimed. Näites 3 on toodud üks eestipärane ja kaks võõrapärast nime, millele Vabamorf ühest analüüsi pakkuda ei osanud.

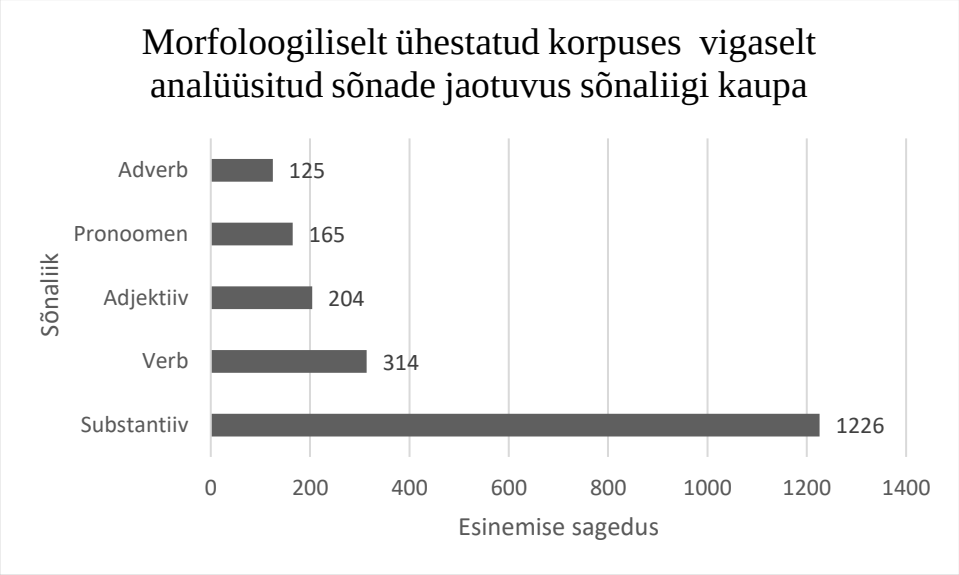
(3)	**Vabamorfi väljund**	**Õige lemma**
	Pung|Punga	*Pung*
	Terek|Tereki|Terekk	*Terek*
	Tours|Toursi|Tourss	*Tours*

Uurides Vabamorfi pakutud mitmeseid analüüse, millest ükski variant õigeks lemmaks ei osutunud, nähtub, et väga tihti seisnes viga sõna *kohal* lemmatiseerimisega. Korpustes on seda tõlgendatud kui muutumatut sõna, mistõttu lemma langeb sõnavormiga ühte, Vabamorf aga pakkus analüüsiks sõnu *koha* või *koht*. Mõnikord tõlgendati tegijanimesid kui tegusõnu, näiteks õige lemma *sündinu* või *lõpetanu* asemel pakuti vastavalt *sündima|sündinud* ja *lõpetama|lõpetanud*.

4.3.2 LemmaTagi vigade analüüs

Morfoloogiliselt ühestatud korpusel treenides oli LemmaTagi mudelil iga testkorpuse puhul kõige rohkem vigu viies sõnaliigis: nimisõnades, tegusõnades, omadussõnades, asesõnades ja määrsõnades. Näited selle jaotuvusest morfoloogiliselt ühestatud korpuse testhulgal on toodud joonisel 2. Esiviisiku täpne järjekord varieerus mõneti, kuid igas korpuses oli kõige tihedamini valesti lemmatiseeritud nimisõnu. Nende vigaste analüüside sagedus ületas teiste sõnaliikide oma vähemalt neljakordselt. Nimisõnad on küll igas korpuses kõige sagedasem sõnaliik, kuid nagu korpuste peatükis ilmneb, on neid verbidest tavaliselt mitte rohkem kui kaks korda enam. Seega esineb nimisõnade lemmatiseerimisel ebaproportsionaalselt palju vigu.

Joonis 2. **Morfoloogiliselt ühestatud korpuse peal treenitud LemmaTagi mudeli enim vigaselt analüüsitud sõnaliikide jaotus sama korpuse testhulgas.**



Näited vigastest analüüsidest on toodud tabelis 11. Nimisõnade puhul ilmneb, et LemmaTag eemaldab või asendab sõnalõpu isegi siis, kui see pole tegelikult käändelõpp. Näiteks sõna *seadusesäte* lemmatiseeritakse kui *seadusesä*, *õdus* kui *õdu*, *musträstas* kui *musträst*. Selliste sõnade puhul võib oletada, et mudel on õppinud treeningandmete peal kindlaid sõnalõppe, nagu mitmuse omastava tunnus *-te* või sisseütleva tunnus *-s* kääneteks liigitama ja eraldab need siis ka sõnadest, kus need on tegelikult tüve osa.

Tabel 11. **LemmaTagi mudeli poolt valesti analüüsitud sõnaliikide näiteid.**

Sõnaliik	**LemmaTagi ennustatud lemma**	**Õige lemma**
Nimisõna	seadusesä	seadusesäte
	õdu	õdus
	muträst	muträstas
	Daamilt	daam
	roa	roog
	toidupoe	toidupood
	kandma	kant
	saatma	saatan
Verb	sündima	sündinud
	andma	antud
	viitma	viitsima
	liigelema	liiklema
	juurelema	juurdlema
Adjektiiv	ülivõimne	ülivõimas
	küpse	küps
	peenik	peenike
Adverb	paljutõutavalt	paljutõotavalt
	samoodi	samamoodi
Pronoomen	tema	see
	see	tema

Teisest küljest aga ei tunne mudel mingeid käändelõppe mõnikord ära ja jätabki nimisõnad käänatud vormi, näiteks sõna *daam* asemel antakse lemmaks *Daamilt, roog*

asemel *roa*, *toidupood* asemel *toidupoe*. Lisaks lemmatiseerib mudel mõned nimisõnad ekslikult tegusõnadeks: näiteks sõnast *kant* on tuletatud *kandma*, sõnast *saatan* aga *saatma*. Kuna sõna saatan lõpp langeb kokku ainsuse esimese pöörde tunnusega, on selline analüüs mõneti mõistetav.

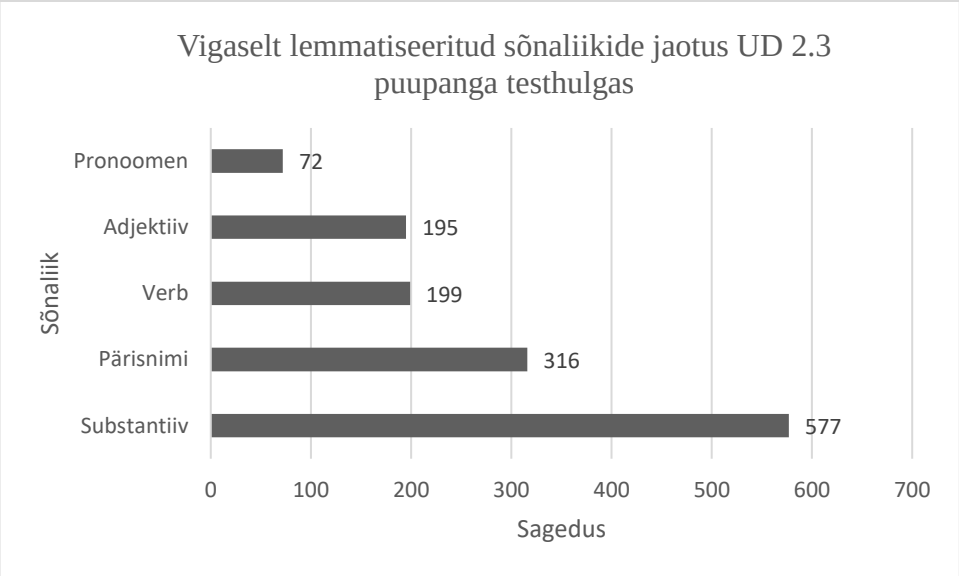
Teine sõnaliik, mida tihedamini vigaselt lemmatiseeritakse, on verb. Siin ilmneb jälle probleem mineviku kesksõna ja ma-infinitiiviga: LemmaTag lemmatiseerib ma-infinitiiviks sõnad, mille lemma õige vorm algfailis on nud-partitsiip, näiteks *sündima* vs *sündinud*, *andma* vs *antud*. Teine probleemitüüp seisneb sisuliselt vigases astmevahelduses ja tuletamises või tähtede ärajäämises: näiteks sõna *viitsima* asemel on lemmaks valitud *viitma*, *liiklema* asemel *liigelema*, *juurdlema* asemel *juurelema*. Mudel on võimeline astmevaheldust õppima, kuid ei oska seda õigesti rakendada, samuti ei määrata nulltuletist verbides alati õigesti.

Adjektiivide puhul oli probleemiks eelkõige sõnalõpu muutmine, käändelõpu allesjätmine või sõnalõpu kaotamine. Adverbide analüüsil esines ootamatuid ja ebareeglipäraseid muudatusi sõnatüves. Asesõnade puhul oli väga levinud *see* ja *tema* lemmade äravahetamine.

4.5.3 Stanfordi mudeli vigade analüüs

Stanfordi mudeli mõlemas – nii UD 2.2 peal treenitud kui morfoloogiliselt ühestatud korpuse peal treenitud – variandis olid märkimisväärselt kõige sagedamini valesti lemmatiseeritud sõnaliigid igas testkorpuses üldiselt järgnevad: substantiivid, pärisnimed, verbid, adjektiivid, pronoomenid. Joonisel 3 on toodud morfoloogiliselt ühestatud korpusel treenitud Stanfordi mudeli viie vigaseima sõnaliigi jaotus. Nagu LemmaTagi mudeli puhulgi, on kaugelt kõige rohkem valesti analüüsitud nimisõnu, eraldi on välja toodud pärisnimed, mida LemmaTag ei märgendanud. Verbe ja adjektiive on valesti analüüsitud peaaegu võrdsel määral, vähem on valesti lemmatiseeritud pronoomeneid.

Joonis 3. **Morfoloogiliselt ühestatud korpusel treenitud Stanfordi mudeli enim vigaselt lemmatiseeritud sõnaliikide jaotus UD 2.3 testhulgas.**



Vigaselt analüüsitud sõnu uurides leiti, et veatüübid sarnanevad oluliselt LemmaTagi vigadele. Näited on toodud tabelis 12. Ka Stanfordi mudelil on probleeme sõnalõppude ärajätmisega: *riigihange* asemel on lemmana antud *riigihang*, *seksikus* asemel *seksik*, mida võiks tõlgendada käändelõppude ärajätmisega analoogselt sõnadega *hang* ja *padrik*. Mõnede sõnade puhul mudel aga käändelõppe üldse ei tuvastanud, näiteks jäeti sõnad *viinamarjaaias* ja *valenoodi* esialgsele kujule. Esineb ka tõlgendab mõned nimisõnad tegusõnadeks, näiteks lemma *dinosaurus* asemel on vasteks pakutud *dinosauruma*, tõenäoliselt meenutab noomeni viimane täht ainsuse 3. isiku imperfekti tunnust. Pisut raskem on seletada, miks *küünal* asemel on mudel lemmaks valinud *küündama*.

Ka verbide lemmatiseerimisel esineb LemmaTagiga sarnaseid probleeme, näiteks on palju vigaseks peetud lemmasid seotud kesksõnade ja ma-infinitiivi äravahetamisega või vigase astmevaheldusega, nagu *teipima* asemel *teivima*. Mudel jätab huvitaval kombel mõne verbi puhul ma-infinitiivi lõpu täiesti ära, näiteks *lisama* asemel pakutakse vasteks *lisa*, mis on tõesti vähemalt omaette käändsõna, kuid *edenema* lemmaks oletatud vormi *edene* küll levinud sõnavormiks pidada ei saa.

Tabel 12. **Stanfordi mudeli poolt valesti analüüsitud sõnaliikide näiteid.**

Sõnaliik	**Stanfordi mudeli poolt ennustatud lemma**	**Õige lemma**
Nimisõna	riigihang	riigihange
	seksik	seksikus
	Lehtsa	lehtsalat
	viinamarjaaias	viinamarjaaed
	valenoodi	valenoot
	dinosauruma	dinosaurus
	küündama	küünal
	ürs	ürt
Verb	teivima	teipima
	tõugama	tõukama
	lisa	lisama
	edene	edenema
Adjektiiv	nauditama	nauditav
	harima	hariv
	visuaalselt	visuaalne
	meetripikkus	meetripikkune
Pärisnimed	meeline	Meelis
	Xina	Xu
	tenerifelt	Tenerife
Pronoomen	tema	see
	see	tema

Adjektiivide puhul erinesid silmatorkavad murekohad LemmaTagi probleemidest. Stanfordi mudel lemmatiseerib v-kesksõna verbiks, isegi kui sellist verbi tegelikult ei eksisteeri: näiteks lemma *nauditama* on igal juhul vale. Mõned sõnad, mis korpuses olid märgendatud adjektiivideks, tõlgendas Stanfordi mudel adverbideks, näiteks korrektse lemma *visuaalne* vasteks pakuti hoopis *visuaalselt*. Vigu esines ka käändelõppude teisendamisel, näiteks pakuti *meetripikkune* asemel lemmaks *meetripikkus*.

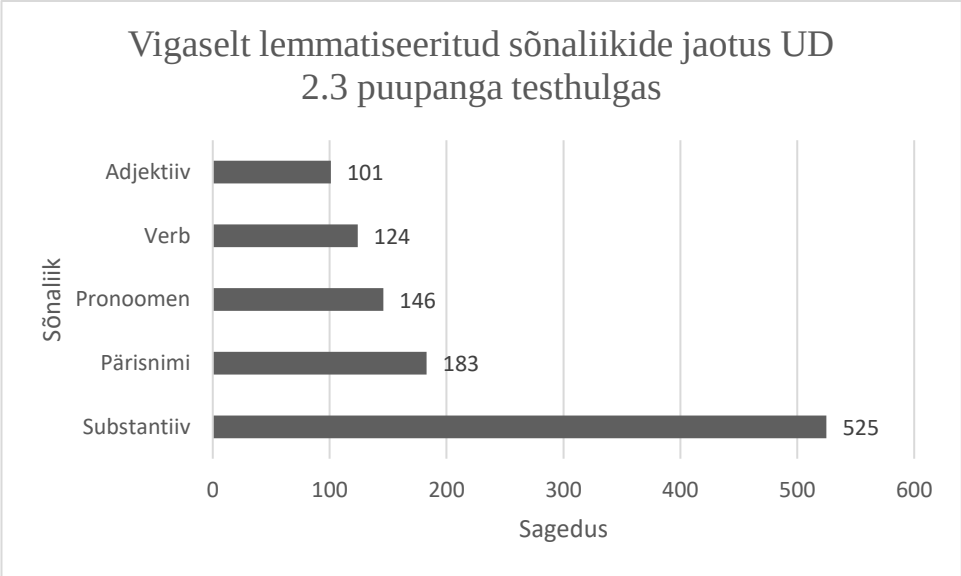
Pärisnimesid on mudel samuti valesti analüüsinud, kas leides neile täiesti uue vormi, nagu *meeline Meelise* vasteks või *Xina* ja *Xu*, või siis käändelõpu alles jättes, näiteks *Tenerifelt* vasteks on pakutud *tenerifelt*.

Pronoomenite puhul on probleem mõlemas neuromudelis identne: lemmad *see* ja *tema* on vahetusse läinud.

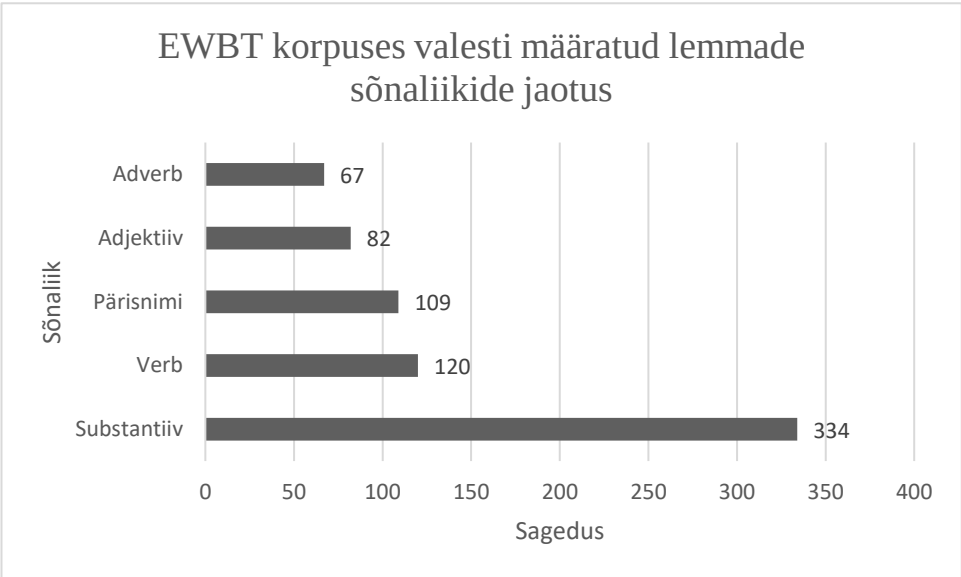
4.5.4 TurkuNLP vigade analüüs

TurkuNLP mudelite poolt ennustatud lemmasid analüüsides ilmnes, et ka siin olid kõige sagedamini valesti lemmatiseeritud sõnaliiki testkorpusteüleselt üldiselt samad: esikohal alati ülekaalukalt substantiivid, ülejäänud esiviisiku moodustasid verbid, pärisnimed, adjektiivid ning pronoomenid või adverbid. Joonisel 4 on esitatud morfoloogiliselt ühestatud korpusel treenitud TurkuNLP mudeli viis kõige sagedamini vigaselt analüüsitud sõnaliiki UD 2.3 testhulgas. Substantiive on valesti analüüsitud ebaproportsionaalselt palju, kui arvestada, et neid on korpuses verbidest veidi üle 2 korra rohkem, aga valesti lemmatiseeritud substantiive on verbidest neli korda enam. Teisel kohal on nagu Stanfordi mudelilgi samas testkorpuses pärisnimed, vigaselt analüüsitud pronoomeneid on TurkuNLP mudelil aga rohkem ning verbe ja adjektiive oluliselt vähem kui Stanfordi mudelil.

Joonis 4. **Morfoloogiliselt ühestatud korpusel treenitud TurkuNLP mudeli enim vigaselt lemmatiseeritud sõnaliikide jaotus UD 2.3 testhulgas.**



Joonis 5. **UD 2.2 puupangal treenitud TurkuNLP mudeli enim vigaselt lemmatiseeritud sõnaliikide jaotus EWBT korpuses.**



Joonisel 5 on võrdluseks toodud UD 2.2 puupangal treenitud TurkuNLP mudeli viis kõige sagedamini valesti lemmatiseeritud sõnaliiki EWBT korpuses. Esiviisik erineb eelnevast näitest selle poolest, et pronoomeneid selles pole, kuid leiduvad adverbid. Kõige sagedamini on valesti analüüsitud substantiive, peaaegu kolm korda vähem on valesti lemmatiseeritud verbe, ehkki testkorpuses oli verbide hulk umbes 80% substantiivide hulgast. Pärisnimesid ja adjektiive on harvemini valesti määratud kui UD 2.3 testhulgas teisel korpusel treenitud mudeliga.

Tabelis 13 on toodud mõlemast korpusest näiteid TurkuNLP mudelite poolt valesti analüüsitud sõnadest. Veatüübid langevad osalt kokku eelneva kahe mudeli probleemidega: ka TurkuNLP ei analüüsinud partitsiipe õigesti, tõlgendades adjektiive verbideks ja vastupidi. Nimisõnade puhul esines palju probleeme sõnalõppude ärajätmisega või muutmisega, näiteks kaotati *rahasumma* lõpust a-täht, *südametuli* asemel pakuti analüüsiks *südametu*, *näolihas*t tõlgendati kui ne-lõpulist sõna. Ka TurkuNLP pidas mõningaid noomeneid verbideks, näiteks *bassikeerd* asemel pakuti vasteks *bassikeerduma*. Seda viga esines nii keerukate ja haruldaste kui ka levinumate nimisõnade puhul.

Mõned vead on aga teist tüüpi, näiteks esines TurkuNLP mudelil palju rohkem probleeme õigekirjaga: eesti keele täpitähed asendati mõnikord muu tähega, näiteks *lühike* asemel *lühike* või *põhjendus* asemel *pöhjendus*. Samuti esines kirjavigu, mida võib tihti leida eestikeelsetes tekstides, näiteks *mõttetu* asemel pakuti lemmaks *mõtetu*, *variant* asemel *varjant*.

Verbide puhul esines probleeme sarnase tüvega verbide lemmatiseerimisel, näiteks saatma asemel pakuti saama ning kuulama asemel kuulma. Mõnel juhul tuletisliiteid muudeti, näiteks *mõtlema* asemel pakkus mudel välja hoopis *mõtelema*. Lisaks ei suudetud mõningaid verbe üldse lemmatiseerida, vaid peeti neid käändsõnadeks, näiteks *keema* asemel pakuti vasteks *kees*, mis võib olla nii sõnavorm tegusõnast kui ka nimisõna *kee* seesütleva vorm..

Pöördel: Tabel 13. **TurkuNLP mudelite poolt valesti analüüsitud sõnaliikide näiteid.**

Sõnaliik	TurkuNLP ennustatud lemma	Õige lemma
Nimisõna	näolihane	näolihas
	rahasumm	rahasumma
	südametu	südametuli
	põhjendus	põhjendus
	varjant	variant
	bassikeerduma	bassikeerd
Verb	mõtelema	mõtlema
	kuulma	kuulama
	saama	saatma
	kees	keetma
Adjektiiv	värkne	värske
	mõtetu	mõttetu
	lyhike	lühike
	rikane	rikas
	turval	turvaline
Pärisnimed	meeline	Meelis
	Mard	Mart
	Hennos	Hennoste
Pronoomen	tema	see
	see	tema
	kee	keegi
Adverb	muie	muideks
	ymber	ümber

Adjektiividel jäeti lisaks õigekirjavigadele ära sõnalõppe, näiteks turvaline asemel pakuti lemmaks turval, lisaks leidus täiesti valesti analüüsitud sõnu, nagu värkne või rikane. Pärisnimedest lemmatiseeriti ka siin valesti näiteks nimi *Meelis*, lisaks tõlgendati nimede viimaseid tähti kui käändelõppe, mistõttu neid lemmatiseerides kas kaotati (*Hennos*) või asendati teise tähega (*Mard*). Pronoomenite põhiviga olid taas lemmad see ja tema, lisaks mõned üksikud muud valesti analüüsitud sõnad, näiteks tõlgendati asesõna *keegi* ilmselt kui gi-partikliga nimisõna *kee*. Adverbide puhul esines probleeme nii õigekirjaga kui ka vale sõnaliigina tõlgendamisega, näiteks pakuti *muideks* vasteks hoopis nimisõna *muie*.

KOKKUVÕTE

Bakalaureusetöö eesmärk oli analüüsida tehisnärvivõrgul põhinevate lemmatiseerijate täpsust eesti keelsetel andmestikel, analüüsida probleemkohti ning võrrelda tulemusi eesti keele põhilise automaatse lemmatiseerimisvahendi Vabamorfi tulemustega. Autori hinnangul töö eesmärk täideti.

Eesmärgist lähtuvalt anti ülevaade nii reeglipäraste ja andmepõhiste lemmatiseerimisvahendite kui ka tehisnärvivõrkude ehitusest ja toimimisest. Täpsemalt kirjeldati kolme selles töös eesti keele lemmatiseerimiseks kasutatud mudelit, TurkuNLP, Stanford ja LemmaTag, nende struktuuri ja vajalikku sisendit. Töös kasutatud korpusmaterjalist, mis koosnes Universal Dependencies eesti keele puupanga 2.2 ja 2.3 versioonidest, morfoloogiliselt ühestatud korpusest ja Estonian Web Treebank korpusest, anti põhjalik ülevaade, lähtudes nii korpuste suurusest lausete, sõnede ja lemmade järgi kui ka sõnaliikide jaotumisest, korpuste vajalikku formaati teisendamisest ja treening-, arendus- ning testhulgaks jaotamisest.

Kõiki kolme lemmatiseerimismudelit treeniti UD 2.2 versiooni puupanga ning morfoloogiliselt ühestatud korpuse peal. Mudelite testimiseks kasutati UD 2.3 versiooni testhulka, morfoloogiliselt ühestatud korpuse testhulka ning EWTB korpust täismahus. Täpsust arvutati nii esialgsel kujul lemmadest, väiketähestatud lemmadest kui ka sellistest väiketähestatud lemmadest, millest oli eemaldatud side- ja alakriipsud. Lisaks anti kõikide testhulkade sõned ette Vabamorfile ning arvutati Vabamorfi lemmade täpsus nii üheste sõnade kaupa kui ka kogu korpuse peal siis, kui mitmestest analüüsidest valiti esimene variant. Lisaks arvutati, kui tihti leidus Vabamorfi mitmese analüüsi seas üldse õige lemmavariant.

Analüüsist selgus, et kõige täpsemad eesti keele lemmatiseerijad on TurkuNLP ja Stanfordi mudelid, LemmaTagi mudeli täpsus jäi iga testhulga puhul teistele alla. Mudelite täpsus testhulgal sõltus mõningal määral sellest, millise korpuse peal mudelit treenitud oli: morfoloogiliselt ühestatud korpusel treenitud mudelid saavutasid sama korpuse testhulgal täpsuse 97% ringis, UD 2.2 peal treenitud mudelid said UD 2.3

testhulgal tulemusi vahemikus 94-98%. Kõige halvemini lemmatiseerisid kõik mudelid EWBT korpust.

Neuromudelitel põhinevate lemmatiseerijate täpsus on väga lähedane Vabamorfi täpsusega samal korpusel, kui mitmestest analüüsidest valida esimene variant. Näiteks olid UD 2.2 peal treenitud neuromudelid Vabamorfist väiketähestatud ja lisamärkideta analüüsis täpsemad UD 2.3 testhulga lemmatiseerimisel 0,3%-2% võrra täpsemad, morfoloogiliselt ühestatud korpuse testhulgal aga jäid Stanfordi ja LemmaTagi neuromudelid Vabamorfile alla. LemmaTag sai ka EWBT testkorpusel Vabamorfist halvema tulemuse, Stanford ja TurkuNLP aga täpsema.

Valesti lemmatiseeritud sõnu analüüsides ilmnes, et Vabamorfi mitmeste lemmavariantide põhjuseks on sageli pärisnimed, millele programm mitmeid vasteid pakub. Lisaks analüüsiti mitmeselt kindlat tüüpi sõnu, näiteks sõna *kohal* ja tegijanimesid. Vabamorfi pakutud mitmesed analüüsid on aga tihti tegelikult grammatiliselt korrektsed ja kuni peaaegu 99% juhtudest leidub mitmese analüüsi seas ka õige lemma.

Neuromudelitel põhinevate lemmatiseerijate veatüüpidest olid enamik omased kõigile mudelitele: näiteks analüüsiti valesti partitsiipe; jäeti ära või transformeeriti sõnalõppe, mis on tegelikult tüve osa; ei tuvastatud käänatud või pööratud sõnast üldse algvormi; mõned nimisõnad analüüsiti tegusõnadeks; pronoomenite puhul ei tehtud vahel lemmadel *see* ja *tema*. Esines ka mudelispetsiifilisi probleeme, näiteks TurkuNLP kaotas ära õige täpitähe ja asendas selle mõne muu tähemärgiga. Neuromudelid pakuvad alati vaid ühe analüüsi ning see võis olla ka täiesti vigane ning eesti keeles üldse mitte esinev sõnakuju.

Eesti keeletehnoloogias ei ole veel neuromudelitel põhinevaid analüüsivahendeid laialdaselt kasutusele võetud. Arvestades, et käesolevas töös kasutati vaid kolme neuromudelit ja need saavutasid kohati paremaid tulemusi kui reeglipõhine lemmatiseerija Vabamorf, võiks lähemalt uurida erinevate neuroarhitektuuride toimimist eesti keele lemmatiseerimisel ning katsetada neid enamatel andmestikel. Selle töö tulemusi saab edaspidi tehisnärvivõrgul põhineva eesti keele lemmatiseerimise arendamisel arvesse võtta.

KIRJANDUS

Bergmanis, Toms, Goldwater, Sharon 2018. Context Sensitive Neural Lemmatization with Lematus. – Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics. lk 1391–1400. Veebiaadress: *https://www.aclweb.org/anthology/N18-1126*. Vaadatud: 02.06.2019.

Buchholz, Sabine, Marsi, Erwin 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. – Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X). New York City: Association for Computational Linguistics. lk 149–164. Veebiaadress: *https://www.aclweb.org/anthology/W06-2920*. Vaadatud: 30.05.2019.

CoNLL 2018. CoNLL 2018. – CoNLL 2018 Shared Task Results. Veebiaadress: *https://universaldependencies.org/conll18/results-lemmas.html*. Vaadatud: 30.05.2019.

EKT = Eesti Keeletehnoloogia koduleht. Veebiaadress: *https://www.keeletehnoloogia.ee/et/ekt-projektid/vabavaraline-morfoloogiatarkvara*.[Vaadatud: 02.06.2019].

Erelt, Mati, Erelt, Tiiu, Ross, Kristiina 2007. Eesti keele käsiraamat. Eesti Keele Instituut. Veebiaadress: https://www.eki.ee/books/ekk09/index.php?p=3&p1=2. Vaadatud: 28.05.2019

Goldberg, Yoav 2017. Neural Network Methods for Natural Language Processing. Morgan & Claypool. (Synthesis Lectures on Human Language Technologies 37.).

Gurney, Kevin 2004. An Introduction to Neural Networks. CRC Press.

Habernal, Ivan, Matousek, Vaclav 2013. Text, Speech, and Dialogue: 16th International Conference, TSD 2013, Pilsen, Czech Republic, September 1-5, 2013, Proceedings. Springer.

Hu, Dichao 2018. An Introductory Survey on Attention Mechanisms in NLP Problems. – arXiv:1811.05544.

Kaalep, Heiki-Jaan 1998. Tekstikorpuse abil loodud eesti keele morfoloogiaanalüsaator. – Keel ja Kirjandus, kd 1, lk 22–29.

Kaalep, Heiki-Jaan, Vaino, Tarmo 1998. Kas vale meetodiga õiged tulemused? Statistkale tuginev eesti keele morfoloogiline ühestamine. – Keel ja Kirjandus, nr 1, lk 30–38.

Kanerva, Jenna, Ginter, Filip, Salakoski, Tapio 2019. Universal Lemmatizer: A Sequence to Sequence Model for Lemmatizing Universal Dependencies Treebanks. – arXiv:1902.00972.

Klein, Guillaume, Kim, Yoon, Deng, Yuntian, Senellart, Jean, Rush, Alexander M. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. – arXiv:1701.02810.

Kondratyuk, Daniel, Gavenčiak, Tomáš, Straka, Milan, Hajič, Jan 2018. LemmaTag: Jointly Tagging and Lemmatizing for Morphologically Rich Languages with BRNNs. – Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics. lk 4921–4928. Veebiaadress: *https://www.aclweb.org/anthology/D18-1532*. Vaadatud: 30.05.2019.

Luong, Minh-Thang, Pham, Hieu, Manning, Christopher D. 2015. Effective Approaches to Attention-based Neural Machine Translation. – arXiv:1508.04025 [cs], .

Makarov, Peter, Clematide, Simon 2018. Imitation Learning for Neural Morphological String Transduction. – arXiv:1808.10701.

Manjavacas, Enrique, Kádár, Ákos, Kestemont, Mike 2019. Improving Lemmatization of Non-Standard Languages with Joint Learning. – arXiv:1903.06939.

Manning, Christopher D., Raghavan, Prabhakar, Schütze, Hinrich 2008. Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press.

Muischnek, Kadri, Kaalep, Heiki-Jaan, Sirel, Raul 2011. Korpuslingvistiline lähenemine eesti internetikeele automaatsele morfoloogilisele analüüsile. – Eesti Rakenduslingvistika Ühingu aastaraamat, kd 7, lk 111–127.

MÜK = Morfoloogiliselt ühestatud korpus. Veebiaadress: *https://keeleressursid.ee/et/keeleressursid-cl-ut/korpused/83-article/clutee-lehed/143-morfkorpus*. Vaadatud: 31.05.2019.

Müller, Thomas, Cotterell, Ryan, Fraser, Alexander, Schütze, Hinrich 2015. Joint Lemmatization and Morphological Tagging with Lemming. lk 2268–2274. Veebiaadress: *https://aclweb.org/anthology/papers/D/D15/D15-1272/*. Vaadatud: 03.06.2019.

Nivre, Joakim, Abrams, Mitchell, Agić, Željko, Ahrenberg, Lars, Antonsen, Lene, Aplonova, Katya, Aranzabe, Maria Jesus, Arutie, Gashaw, Asahara, Masayuki, Ateyah, Luma, Attia, Mohammed, Atutxa, Aitziber, Augustinus, Liesbeth, Badmaeva, Elena, jt 2018. Universal Dependencies 2.3. – http://universaldependencies.org/. Vaadatud 30.05.2019

Orasmaa, Siim, Petmanson, Timo, Tkachenko, Alexander, Laur, Sven, Kaalep, Heiki-Jaan 2016. ESTNLTK - NLP Toolkit for Estonian. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). Toimetanud Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaar, Joseph Mariani, Asuncion Moreno, Jan Odijk, Stelios Piperidis. Portorož, Sloveenia. ELRA, lk 2460–2466.

Plisson, Joël, Lavrac, Nada, Mladenic, Dunja 2004. A Rule based Approach to Word Lemmatization. Veebiaadress: https://pdfs.semanticscholar.org/5319/539616e81b02637b1bf90fb667ca2066cf14.pdf. Vaadatud 03.06.2019

Qi, Peng, Dozat, Timothy, Zhang, Yuhao, Manning, Christopher D. 2018. Universal Dependency Parsing from Scratch. – Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Brussels, Belgium: Association for Computational Linguistics. lk 160–170. Veebiaadress: *https://nlp.stanford.edu/pubs/qi2018universal.pdf*. Vaadatud: 25.05.2019.

Salehinejad, Hojjat, Sankar, Sharan, Barfett, Joseph, Colak, Errol, Valaee, Shahrokh 2017. Recent Advances in Recurrent Neural Networks. – arXiv:1801.01078.

Straka, Milan, Straková, Jana 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. – Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Vancouver, Canada: Association for Computational Linguistics. lk 88–99. Veebiaadress: *http://www.aclweb.org/anthology/K/K17/K17-3009.pdf*. Vaadatud: 26.05.2019.

Sutskever, Ilya, Vinyals, Oriol, Le, Quoc V 2014. Sequence to Sequence Learning with Neural Networks. – Advances in Neural Information Processing Systems 27. Toim Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger. Curran Associates, Inc., lk 3104–3112.

Särg, Dage, Muischnek, Kadri, Müürisep, Kaili 2018. Annotated Clause Boundaries' Influence on Parsing Results. – Text, Speech, and Dialogue. Toim Petr Sojka, Aleš Horák, Ivan Kopeček, Karel Pala. Cham: Springer International Publishing, lk 171–179.

Tkachenko, Alexander, Sirts, Kairit 2018. Neural Morphological Tagging for Estonian. – arXiv:1810.06908.

TurkuNLP = TurkuNLP, Turku neural parser pipeline. Veebiaadress: *http://turkunlp.org/Turku-neural-parser-pipeline/training.html*.[Vaadatud: 22.05.2019].

Viks, Ülle 2000. Eesti keele avatud morfoloogiamudel. – Arvutuslingvistikalt inimesele. Toim Tiit Hennoste. (Tartu Ülikooli üldkeeleteaduse õppetooli toimetised) Tartu: Tartu Ülikooli Kirjastus, lk 9–36.

MLB = Machine Learning Blog. Veebiaadress: *https://machinelearning-blog.com/2018/02/21/recurrent-neural-networks/*. Vaadatud: 08.06.2019.

COMPARATIVE ANALYSIS OF NEURAL NETWORK BASED LEMMATIZERS IN THE ESTONIAN LANGUAGE

Summary

Currently, most tools used in text analysis for the Estonian language are rule-based or statistical programs, such as the open-source morphological analysis software Vabamorf. Although neural networks have proven to be powerful tools for various tasks in natural language processing, no neural network based models have been developed nor are in use specifically in Estonian text analysis. The aim of this bachelor's thesis was to analyse the precision and results of neural network based lemmatizers in the Estonian language.

To this end, three different neural models for lemmatizing – TurkuNLP, Stanford and LemmaTag were used on various Estonian data. The data consisted firstly of two versions of the Universal Dependencies Estonian treebanks, namely version 2.2 for training and version 2.3 for testing the models; secondly of the Estonian morphologically disambiguated corpora (EMDC for short), split into training, development and test sets; and thirdly, the Estonian Web Treebank corpus, used only for testing purposes. Each of the three models was trained on both UD 2.2 and EMDC, and tested on the test sets of UD 2.3 and EMDC as well as the entire Treebank corpus. In addition, the test sets were used on the rule-based Estonian lemmatizer Vabamorf.

An analysis of the results showed that while LemmaTag lagged behind TurkuNLP and Stanford in precision, and Vabamorf outperformed the models on specific combinations of train and test data, e.g when trained on UD 2.2 and tested on EMDC, the models were often more precise or at least comparable with Vabamorf. Examining the incorrectly lemmatized words shows that the neural models tend to make similar mistakes, e.g mixing up certain pronouns or mistakenly interpreting the final characters of a word as inflected.

These neural models, especially TurkuNLP and Stanford, showed fairly good results in lemmatizing Estonian texts, never dropping below 93%. As this thesis was only concerned with three specific models, further research on the efficiency of different neural architectures on various Estonian-language data would certainly yield beneficial results.

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Laura Katrin Leman,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

„Tehisnärvivõrgul põhinevate lemmatiseerijate võrdlev analüüs eesti keeles“ ,

mille juhendaja on Kairit Sirts,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Laura Katrin Leman

11.06.2019