TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kaarel Randorg 176810IAPM

EESTIKEELSE KÕNEABITARKVARA TEKSTISÜNTESAATORI LOOMINE REKURRENTSETE NÄRVIVÕRKUDE ABIL

magistritöö

Juhendajad: Kairit Sirts
PhD
Martin Rebane
MSc

Tallinn 2019

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kaarel Randorg

09.05.2019

Annotatsioon

Töö eesmärgiks oli luua tekstisüntesaator, mida saaks kasutada kõneabitarkvaras, mis laseb kasutajal valida järjestikku pilte, mille tulemusena moodustatakse piltidest lause ning loetakse see välja. Tekstisüntesaatori sisendiks on jada lemmadest, mis moodustavad lause ning väljundiks korrektne eestikeelne lause. Selle teostamiseks jagati ülesanne kahte ossa. Esiteks tuli lemmadele määrata lauseosa märgend ja morfoloogilised märgendid, mis näitavad, mis tüüpi sõna on ja mis käändes või pöördes sõna on. Teiseks tuli määratud märgendite abil sünteesida lemma õiges vormis sõnaks.

Märgendite määramiseks katsetati kahte rekurrentsetel närvivõrkudel töötavat mudelit: Stanfordi jadamärgendamise mudelit ning standardset BiLSTM'il põhinevat jadamärgendamise mudelit. Lemma sünteesimiseks kasutati Stanfordi lemmatiseerijat, mis kohandati selliseks, et õiges vormis sõnast lemma ennustamise asemel ennustati lemmast õiges vormis sõna. Alternatiivse lahendusena kasutati lemma sünteesimiseks ka EstNLTK teeki. Mudelite treenimiseks kasutati kahte tekstikorpust: Universal Dependencies ja CoNLL-U 2017.

Parima terviksüsteemi leidmiseks leiti iga mudeli parim tulemus ja parimad mudelid kombineeriti omavahel. Terviksüsteemi parimaks tulemuseks saadi 74,39. Parim jadamärgendaja tulemus oli 75,88 ning sünteesija mudeli täpsus oli 88,15.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 6 peatükki, 13 joonist, 16 tabelit.

Abstract

Creating a Text Synthesizer for Estonian Speech Aid Using Recurrent Neural Networks

The goal of this thesis was to create a text synthesizer, that could be used in a speech aid software, which allows user to select sequential images, based on the images user selects, a sentence is formed using text synthesizer and read our using speech synthesizer. The input for the text synthesizer is a sequence of lemmas, which form a sentence. The output of the text synthesizer is a correct Estonian sentence. To achieve this, the task was divided into two subtasks. First of all, part-of-speech tag and morphological tags were predicted to each lemma. This tells us what kind of noun case or verb form should the lemma be in. Secondly, the predicted tags were used to synthesize lemma into the correct form.

To predict the part-of-speech tag and morphological tags, two recurrent neural networks were tested: Stanford's sequence tagger and a standard sequence tagger based on BiLSTM. For lemma synthesizing, Stanford's lemmatizer was used, which was modified to predict a word in the correct form from lemma, instead of predicting lemma from the correct word. Alternatively, the EstNLTK library was used to synthesize lemma. Two text corpora was used to train the models: Universal Dependencies and CoNLL-U 2017.

To find the best complete system, the best result for each model was found and the best models were combined. The highest accuracy for the whole system was 74,39. The highest accuracy for sequence taggers was 75,88 and the accuracy of the synthesizer model was 88,15.

The thesis is in Estonian and contains 33 pages of text, 6 chapters, 13 figures, 16 tables.

Lühendite ja mõistete sõnastik

AAC	*Augmentative and alternative communication, a*ugmentatiivne ja alternatiivne kommunikatsioon
BiLSTM	*Bidirectional Long Short-Term Memory*, kahesuunaline pikaajalise lühimäluga närvivõrk
BiRNN	*Bidirectional recurrent neural network*, kahesuunaline rekurrentne närvivõrk
CoNLL-U	Tekstikorpuse formaat
EstNLTK	*Estonian Natural Language Toolkit*, Pythonis kirjutatud kogumik teeke eestikeelsete tekstide töötluseks
Lemma	Sõna selle algvormis
LSTM	*Long Short-Term Memory*, pikaajalise lühimäluga närvivõrk
Morfoloogia	Vormiõpetus, käsitleb sõnamuutmist ja -moodustust
RNN	*Recurrent neural network*, rekurrentne närvivõrk
Tekstikorpus	Struktureeritud tekstide kogumik
UD	*Universal Dependencies*, mitmekeelne grammatika annoteerimise raamistik

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Inimestel võib esineda erinevaid kõnehäireid, mis võivad olla tingitud mõnest haigusest või kaasasündinud probleemist. Kõnehäiretega inimeste eneseväljendamiseks on loodud abivahendid, mis neid aitavad. Neid vahendeid nimetatakse augmentatiivse ja alternatiivse kommunikatsiooni vahenditeks [1]. Siin töös keskendutakse vahendile, mis laseb inimestel järjestikku valida erinevaid pilte, mille tulemusena luuakse tekstisüntesaatori abil piltidest tekst ning öeldakse kõnesüntesaatori poolt loodud lause välja. Iga valitud pilt vastab ühele algvormis sõnale ehk lemmale. Näiteks valib inimene pildid, mis kujutavad sõnu „mina", „tahtma" ja „õun", seejärel muudab tekstisüntesaator need sõnadeks „mina", „tahan" ja „õuna" ning kõnesüntesaator loeb loodud lause välja.

Hetkel ei ole autorile teadaolevalt eesti keelele kohandatud kõrgtehnoloogilist lahendust. On olemas madaltehnoloogilised lahendused ehk raamatud ja kaardid, kus on ikoonid ja nende tähendus. Kõrgtehnoloogiline lahendus oleks kiirem ja efektiivsem lahendus inimestele, kellel puudub kõnevõime.

Kuna iga pilt väljendab sõna selle algvormis ei ole valitud lause moodustamine triviaalne, kuna peab algvormides sõnadest moodustama korrektse eestikeelse lause. Algvormidest lauseid moodustava tarkvara on võimalik luua näiteks kas reeglisüsteemi või treenitava närvivõrgumudeli abil. Siin töös luuakse lahendus rekurrentsete närvivõrkude (RNN) abil. See võimaldab katta palju suurema osa keelest, kuid üksikuid leitud vigu korrigeerida on keerulisem [2].

Töö eesmärgiks on lausest, kus on sõnad lemma kujul, moodustada eestikeelselt korrektne lause. Korrektse lause koostamine koosneb kahest osast. Esiteks leitakse igale lemmale lauseosa märgend ja morfoloogilised märgendid, mis näitavad sõnatüüpi ning mis käändes või pöördes sõna olema peab. Kui märgendid on leitud sünteesitakse lemma märgenditele vastavaks sõnaks ehk siis pööratakse või käänatakse sõna. Esimese sammu ehk lauseosa ja morfoloogiliste märgendite ennustamiseks võrreldi kahte rekurrentsetel närvivõrkudel põhinevat arhitektuuri: Stanfordi biafiinset märgendajat [3] ning tavapärast BiLSTM jadamärgendajat. Teise sammu realiseerimiseks treeniti

jadagenereerimise närvivõrk, mis baseerub Stanfordi lemmatiseerija mudelil [3] ning alternatiivse lahendusena kasutatakse eesti keele morfoloogilist süntesaatorit, mis on kasutatav EstNLTK [4] teegi vahendusel.

Kõiki mudeleid hinnatakse individuaalselt ning ka koos ehk siis esiteks ennustatakse märgendid ning nende märgendite põhjal sõna. Hindamiseks kasutatakse ennustuste õigsust ehk mitu märgendit või sõna õigesti ennustati kogu märgendite või sõnade hulgast. Mudelite treenimiseks ja hindamiseks kasutatakse Universal Dependecies [5] ja CoNLL-U 2017 [6] tekstikorpuseid. Parimaks tulemuseks lemmadest koosneva lause korrektseks eestikeelseks lauseks sünteesimisel oli 74,39.

Töö on jaotatud nelja ossa: taust, seotud tööd, metoodika ja eksperimendid. Tausta osas tehakse ülevaade kasutatavatest närvivõrkude arhitektuuridest, EstNLTK teegist ning seletatakse, mis on augmentatiivne ja alternatiivne kommunikatsioon. Metoodika osas kirjeldatakse andmestikku, mille peal mudeleid treeniti ning tehakse ka ülevaade mudelitest, mida kasutatakse. Eksperimentide osas presenteeritakse ja analüüsitakse tulemusi.

2 Taust

Siin peatükis selgitatakse lähemalt töös kasutatavaid termineid ja tehnoloogiaid. Alguses antakse ülevaaade, mis on augmentatiivne ja alternatiivne kommunikatsioon ning seejärel selgitatakse närvivõrkude mudelite tööpõhimõtteid. Lõpuks tehakse ka ülevaade töös kasutatava EstNLTK teegi kohta.

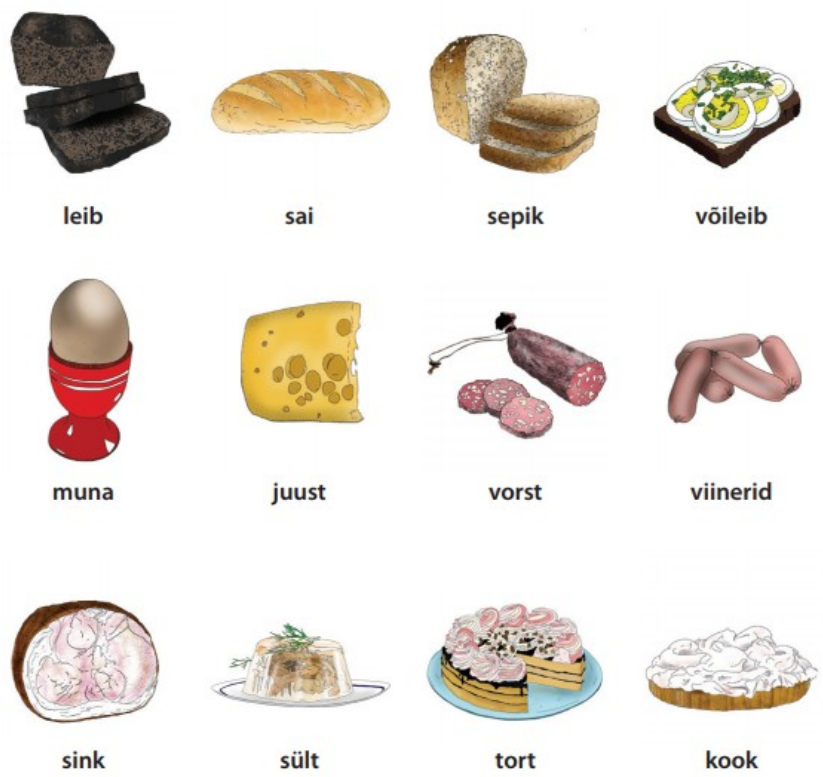
2.1 Augmentatiivne ja alternatiivne kommunikatsioon

Nii lastel kui ka täiskasvanutel võivad esineda raskeid kõneprobleeme, mille tõttu nad ei saa ennast väljendada tavapärase rääkimise abil. „Augementatiivne ja alternatiivne kommunikatsioon (ingl k *Augementative and alternative commucation* e AAC) on erinevate tööriistade ja strateegiate kogum, mis aitab inimesel lahendada igapäevaseid kommunikatsiooniga tekkivaid probleeme” [7]. Suhelda on võimalik mitmetel erinevatel viisidel, näiteks kõne, teksti, kehakeele vahendusel – neid viise kasutavad enamus inimesi igapäevaselt, kuid ka näiteks viipekeelega, sümbolite, piltide ja kõnet genereerivate vahenditega [7]. AAC alla kuuluvad neist sümbolite ja piltidega suhtlemine, kõnet genereerivad vahendid ja viipekeel [8].

AAC vahendid jaotuvad kahte eraldi kategooriasse: abistamata vahendid ja abistatud vahendid. Abistamata vahenditeks liigitatakse vahendeid, mille jaoks ei ole vaja muud, kui inimese keha ning abistatud vahenditeks on vahendid, kus läheb vaja mõnda tööriista või seadet, näiteks paberit ja pliiatsit [8]. Abistatud vahendid jaotuvad omakorda veel kaheks: madaltehnoloogilised ja kõrgtehnoloogilised [8]. Kõrgtehnoloogilisteks vahenditeks on sellised, mis kasutavad töötamiseks ka arvutit, näiteks puutetundliku ekraaniga vahend, kus saab inimene järjest sõnu või pilte valida ning arvuti väljastab sisestatud lause helina [8].

Siin töös keskendutakse kõrgtehnoloogilisele AAC vahendile, kus inimene saab valida pilte, millega end väljendada. Iga pilt sümboliseerib ühte sõna, näiteks saab valida inimene järjest „mina”, „tahtma”, „sööma” ning sellega väljendab inimene, et ta tahab

süüa. Eesti keeles puudub praegu kõrgtehnoloogiline lahendus piltidega suhtlemiseks. On olemas madaltehnoloogilised vahendid ehk pilditahvlid Joonis 2, kuid ka need pole veel koos arvutile arusaadavate metaandmetega digitaliseeritud. Hetkel on autorit konsulteerinud spetsialistidel plaanis koostada piltide kogumik, mis katab kogu eesti keele põhisõnavara sõnastiku. Põhisõnavara sõnastik on EKI poolt koostatud sõnastik, kus on eesti keele 5000 olulisemat sõnad, see on koostatud eelkõige, et aidata algajaid ja edasijõudnuid keeleõppijaid [9].

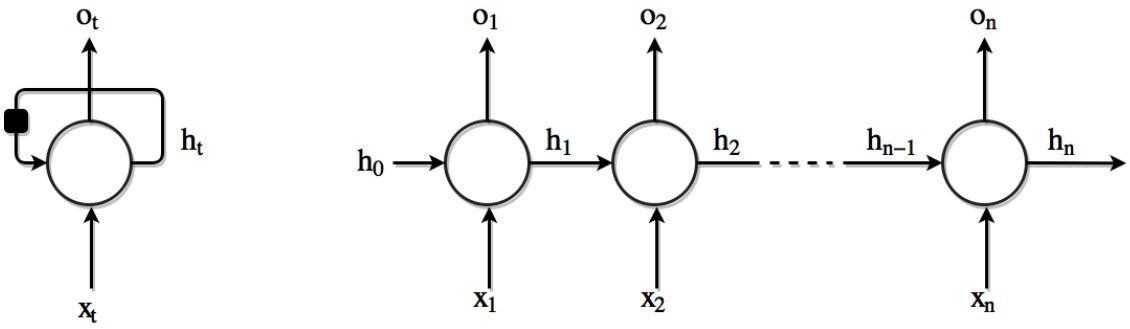


Joonis 2: Näide EKI pildilehest [26]

2.2 Rekurrentsed närvivõrgud

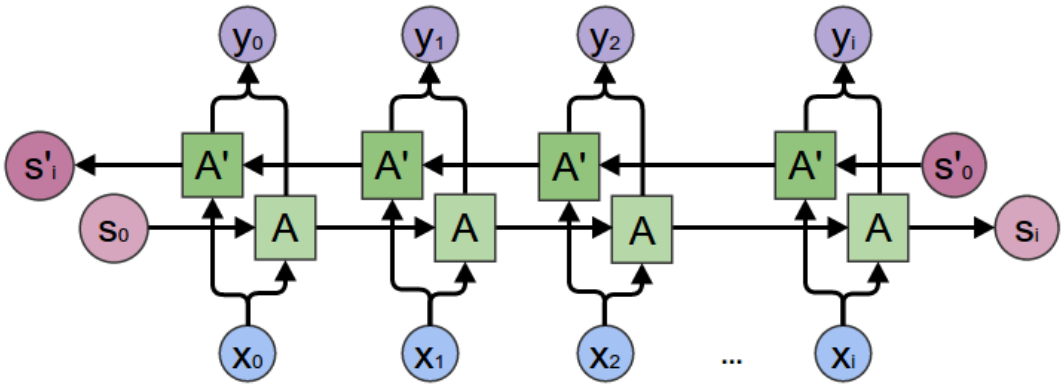
Pärilevi närvivõrkude üheks miinuseks on see, et nad on liiga piiritletud, kuna pärilevi närvivõrgud saavad sisendiks kindla pikkusega vektor ning ka väljastavad kindla pikkusega vektori. Rekurrentsete närvivõrkude (RNN) sisendiks ja väljendiks võib olla jada vektoritest. RNN’e kasutatakse näiteks kõnetuvastuses [10], pildituvastuses [11] ja

masintõlkimises [12]. Masintõlkimise kontekstis võib mõelda sisendina jada sõnadest, mis moodustavad lause ühes keeles ning väljundiks jada sõnadest teises keeles.



Joonis 3: RNN närvivõrk.Vasakul pool on närvivõrk kujutatud tsüklilisena ning paremal pool on sama närvivõrk, kuid lahtirullitult [13]

RNN närvivõrgus töödeldakse iga jada element läbi ning iga elemendi peal sooritatakse sama operatsioon. Operatsiooni väljund sõltub praegusest elemendist ning eelnevatest operatsioonide tulemustest. Joonisel (Joonis 3) on kujutatud RNN närvivõrk ning on näidatud, et igal ajahetkel saab närvivõrk sisendiks x_t ning h_t, mis hoiab informatsiooni, mis toimus närvivõrgus enne ning igal sammul on väljundiks o_t [13].



Joonis 4: Kahesuunaline RNN võrk [14]

Kahesuunaline RNN (BiRNN) võimaldab saada ka igal ajahetkel informatsiooni nii eelneva kui ka järgneva jada elementide kohta. See saavutatakse kui panna kaks RNN võrku kokku ning ühele anda sisendiks õiges järjekorras jada ning teisele võrgule anda vastupidises järjestuses jada. See on kasulik näiteks lausele lauseosa märgendi ennustamisel, kus tuleb kasuks teada nii eelnevaid kui ka järgnevaid sõnu [14].

Tavaline RNN võrk ei pruugi toimida väga hästi, kuna RNN võrgud üritavad seost luua paljude eelnevate operatsioonide tulemustega ning on raske hinnata kui olulist rolli eelnevad tulemused mängivad. RNN'is on igal sammul samad kaalud ning seetõttu võib gradient kas väga suureks või väga väikeseks minna. Selle vältimiseks on olemas lülitusrakkudega RNN'id – GRU (Gated recurrent unit) ja LSTM (Long short-term memory). Lülitusrakk määrab iga sisendi oleku juures kui palju sellest sisendist uut infot salvestatakse ja kui palju teadaolevast infost peaks unustama [15, lk 204–206].

2.3 Jadamärgendamine

Üheks rekurrentsete võrkude kasutusalaks on jadamärgendamine ning üheks levinuks jadamärgendamise ülesandeks on lauseosamärgendamine, millega tegeletakse ka käesolevas töös. Lauseosamärgendamisel määratakse igale sisendsõnale märgend [15, lk 189–190].

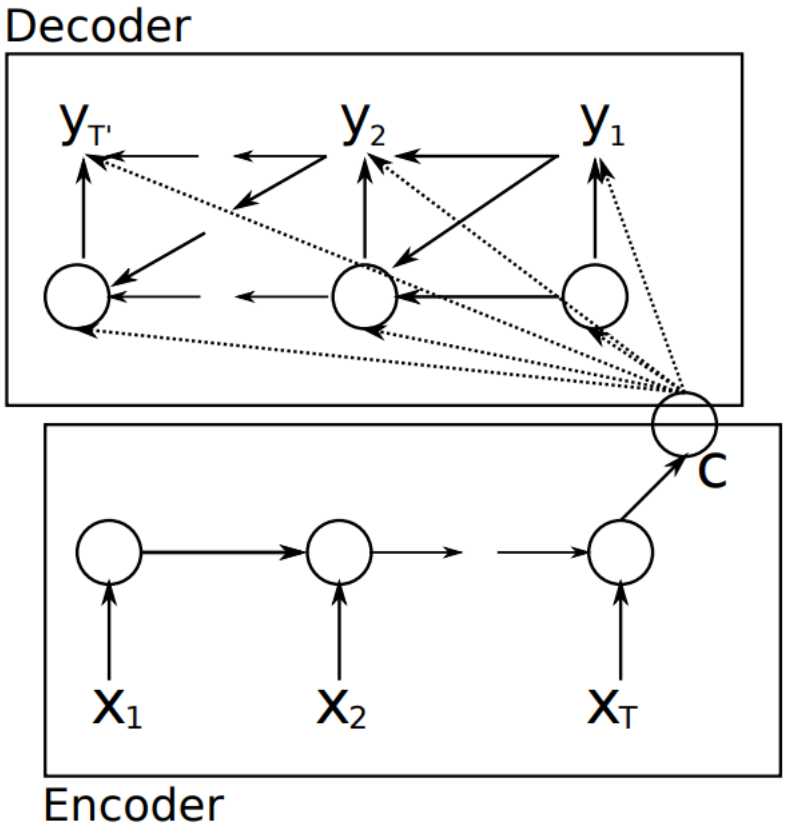
Kui on antud lause, mis koosneb sõnadest kujul $s = w_{1:n}$, siis muundatakse nad sisendvektoriteks $x_{1:n}$ kasutades tunnusfunktsiooni. Saadud sisendvektoreid kasutatakse kahesuunalisse RNN'i ja saadakse väljundvektorid $y_{1:n} = biRNN(x_{1:n})$. Iga väljundvektor y_i on sisendiks mitmekihilisele närvivõrgule, mis ennustab ühe märgendi igale sõnale (Joonis 4). Iga väljundvektor y_i on fokuseeritud positsioonile i , kuid omavad informatsiooni ka terve jada kohta. Treenimise ajal õpib närvivõrk ära tundma jada aspekte ning see aitab märgendeid ennustada [15, lk 189–190].

Sisendvektoris on sõnad sõnavektori kujul. Sõnavektori võib initsialiseerida juhuslikult või kasutada mõnda eeltreenitud sõnavektorit, näiteks FastText või Word2Vec. Ehk siis sisendiks on sõnade positsioon sõnavektoris [15, lk 189–190].

2.4 Kooder-dekooder mudel tähelepanu mehhanismiga

Kooder-dekooder mudel võeti kasutusel masintõlkimisel [16], kus oli probleemiks, et sisendjada ja väljundjada pikkused võivad erineda. Kooder-dekooder mudelis on kaks komponenti, kooder ja dekooder. Kooderi ülesandeks on käia läbi kogu sisendjada ja kodeerib selle kindla pikkusega vektoriks, mida kutsutakse kontekstvektoriks. Kontekstvektorisse saab kodeerida informatsiooni, mis treenimise käigus kasutatusel ja

arvatakse et on kasulik. Dekooder ennustab väljundjada kasutades kontekstvektorit ning eelnevaid ennustusi [15, lk 196–200].



Joonis 5: Kooder-dekooder mudel, kus x'id tähistavad sisendeid, C kontekstvektorit ja y'id väljundeid [12]

Probleem kooder-dekooder mudeliga tuleb välja kui on vaja ennustada pikki jadasid, sest närvivõrk peab kogu sisendjada kodeerima üheks fikseeritud suurusega vektoriks. Dekodeerimise faasis peab siis närvivõrk arvestama kogu lause infoga ning RNN võrgud ei toimi hästi pikkade jadadega. Selle probleemi lahendab tähelepanu mehhanism. Tähelepanu mehhanism loob kontekstivektori, mis on filtreeritud iga väljundjada elemendi jaoks [15, lk 196–200].

2.5 Eesti keele automaatne töötlus: EstNLTK

EstNLTK (NLTK ehk Natural Language ToolKit) on peamiselt Pythonis kirjutatud kogumik teeke eestikeelsete tekstide töötluseks [4]. EstNLTK teek kasutab tuumas juba olemasolevaid lahendusi naturaalsete keeletöötluse ülesannete sooritamiseks ning selle eesmärk on kõik erinevad olemasolevad tööriistad kokku koguda ühe teegi alla, et neile oleks kergem ligipääs. Keeletöötluse ülesannete alla kuuluvad näiteks lemmatiseerimine, tokeniseerimine, morfoloogiline analüüs ja süntees, lauseosamärgendamine, nimeüksuste tuvastamine [17].

Lemmatiseerimisel muudetakse mõnes käändes või pöördes olev sõna lemma ehk sõna algvormi kujule, näiteks sõna „kassile” muudetakse sõnaks „kass”. Morfoloogiline analüüs annab sõnast selle morfoloogilised märgendid (kääne, pööre jm.) ning sünteesimisel rakendatakse neid märgendeid lemmale ning moodustatakse õiges vormis sõna. Tokeniseerimisel saadakse tekstist teada millised osad on laused ning lausete järgi on võimalik jagada tekst ka paragrahvideks. Lauseosa märgendamisel määratakse sõnale tema lauseosamärgend (nt tegusõna, nimisõna, määrsõna jm.). Nimeüksuste tuvastamisel tuvastatakse, kas sõna on nimi, organisatsioon või asukoht [17]. Siin töös kasutatakse morfoloogilist analüüsi ja lauseosamärgendamist algandmete eeltöötlemiseks ning morfoloogilist sünteesi eksperimentide osas.

Morfoloogilise analüüsi ja sünteesi teostamiseks kasutab EstNLTK C++’is kirjutatud Vabamorfi teeki [17], [18]. Vabamorfi teek teostab morfoloogia analüüsi ja sünteesi kasutades sõnastikupõhist lähenemist. Umbes 98% sõnadest on võimalik analüüsida kasutades sõnastikke, kuid ülejäänute sõnade jaoks kasutab Vabamorf ennustamist. Ennustatakse kasutades sõna lõppe ning silpide arvu. Kõige keerulisem on Vabamorfi teegil ennustada võõrsõnast nimesid [18].

3 Seotud tööd

AAC teemal on 2018. aastal Shiran Dudy ja Steven Bedrick poolt tehtud töö "Compositional Language Modeling for Icon-Based Augmentative and Alternative Communication" [19]. Selles töös uuriti, kuidas sünteesida algandmestikku, mille põhjal luua keelemudel. Probleem, mida lahendati seisnes selles, et AAC süsteemidel ei ole head keelemudelit, mille põhjal ennustada, mida kasutaja öelda tahab. Kui AAC tarkvaras olevate ikoonide arv suureneb, siis on kasutajal raske ennast väljendada.

Morfoloogia ennustamist lemmade baasil on autorile teadaolevalt vähe uuritud. Ainuke teadaolev töö, mis on morfoloogiliste sümbolite ennustamist lemmade baasil uurinud on C. Conforti, M. Huck ja A. Fraser valminud "Neural Morphological Tagging of Lemma Sequences for Machine Translation" [20] ning selleks kasutatakse masinõpet. Nemad uurisid seda masintõlkimise kontekstis. Idee oli tõlkida algkeelest lõppkeelde nii, et tekiks jada algvormidest. Seejärel saab algvormidest ennustada morfoloogilisi sümboleid. Selle töö eesmärk kattub ühe välja pakutud mudeliga, kuid meil on tarvis ka ennustatud morfoloogilistest sümbolitest sünteesida korrektses vormis lause ning seda eelmainitud töö ei kata. Antud töös oli mudel koostatud kolmekihiliselt kahesuunalistest GRU (Gated recurrent unit) rakust, mille peal on *softmax*-kiht, mis teisaldab numbrilist sisendit normaliseeritud tõenäosuste jaotuseks. Töös saadi morfoloogiliste väljendite ennustamisel F1 skooriks 91,22 [20]. F1 skoor näitab harmoonilist keskmist täpsusest ja saagisest. Täpsust näitab suhet õigesti positiivselt ennustatud tulemuste ja kogu positiivselt ennustatud tulemuste vahel ja saak näitab suhet õigesti positiivselt ennustatud tulemuste ja tegeliku positiivsete tulemuste vahel [21].

Hea tulemus annab kinnitust, et morfoloogilisi märgendeid on võimalik edukalt ennustada. Kuid suurim erinevus käesoleva töö ühe väljapakutud lahenduse ja eelmainitud töö lahenduse vahel on see, et meil on teada vähem erinevat liiki morfoloogilisi tunnuseid. EstNLTK teek võimaldab 74 [4] erinevat morfoloogilist väljendit, kuid eelmainitud töös on neid kokku 678 [20]. See annab lootust, et väiksema märgendite arvuga võib siin töös seatud ülesanne sama hästi või isegi paremini õnnestuda.

Kui töös lahendatavat probleemi natuke lahti mõtestada, siis võib seda käsitleda kui ühest keelest teksti teise keelde tõlkimist. Üks keel on siis ainult algvormidest ehk lemmadest koosnev ning keel, kuhu tõlkida tahetakse on korrektne eesti keel. Masintõlkimise probleemi on palju uuritud ning masinõppe arenemisel on seda hakatud lahendama närvivõrke kasutades. Üks esimene töö, mis masintõlkimises rekurrentseid närvivõrke kasutas on I. Sutskeveri, O. Vinyalsi ja Q. V. Le'i poolt koostatud "Sequence to Sequence Learning with Neural Networks" [16]. Töös kasutati nii kooderi kui dekooderi faasis mitmekihiliselt LSTM närvirakke ning saadi statistilisest masintõlkest paremad tulemused, kusjuures LSTM'idel koosnev närvivõrk suutis tõlkida ka väga pikki lauseid [16].

Stanfordi ülikool avaldas oma lähtekoodi "CoNLL 2018 UD Shared Task" raames valminud süsteemile [3], mille eesmärk oli teha täielik närvivõrkudel baseeruv süsteem, mis võtab sisendiks töötlemata teksti ning suudab teostada kõik ülesanded, mis olid "CoNLL 2018 UD Shared Taski" raames vajalikud teksti süntaktiliseks analüüsiks. Nende hulka kuulusid näiteks sõnade lemmatiseerimine, lausa osa määratlemine ja sõltuvuste analüüs. Seda ülesannet lahendasid mitmeid uurimisgrupid, kuid Stanfordi ülikooli tulemus oli igas arvestuses vähemalt kolme parima seas [22].

Käesoleva töö raames huvitab meid nende lahendus, kuidas nad sõnu lemmatiseerisid, sest käesoleva töö üks osa on lemmadest sõnad õigesse vormi sünteesida ehk siis vastupidine protsess Stanfordi omale. Nende lahendus koosneb kolmest osast. Esmalt koostatakse sõnastik, mis sisaldas paari sõnast ja lause osa märgendist, mis viitas lemmale. Järgnevalt tehti sõnastik kus oli ainult sõna ning viide lemmale. Kui sõnastikust vastet ei leita, siis tuleb kasutusele kooder-dekooder arhitektuuri kasutav mudel, mis koosneb kahesuunalisest LSTM rakkudest, millele on lisatud ka tähelepanu mehhanism. Enne kooder-dekooder mudeli kasutamist ennustatakse veel, kas sõna ise on lemma ning kui on, siis mudelit ei kasutata [22].

Stanfordi mudel sai lemmatiseerimise ülesandes F1 skooriks 94.22 üle kõikide keelte, mis on võrreldes teiste organisatsioonide tulemusega märgatavalt parem [23]. See annab lootust, et kui mudelisse teha muudatusi, siis on võimalik ka hea tulemus saada lemmadest õigesse vormi lausete ennustamisel [22].

4 Metoodika

Siin peatükis kirjeldatakse metoodikat, kuidas lahenduseni jõutakse ning tuuakse põhjused, miks valiti selline lahendusviis.

Kuna sisendinfoks on lemma kujul sõnad ning väljundiks peab olema õiges vormis sõnad, siis jaotati ülesanne kahte ossa. Esimeses pooles ennustatakse lemmale lauseosa märgend ja morfoloogilised väljendid. Teises osas kasutatakse lemmat ning esimese osa väljundeid ning ennustatakse õiges vormis sõnad.

Esimese osa lahendamiseks treenitakse kaks alternatiivset erineva arhitektuuriga närvivõrku. Teises osas treenitakse üks närvivõrk ning alternatiivina kasutatakse EstNLTK teeki. Lõpuks kombineeritakse esimese ja teise osa lahendeid, et leida parim tulemus.

4.1 Andmestik

Andmestikuna kasutatakse Universal Dependencies (UD) [5] keelekorpust. Keelekorpus on struktureeritud kogumik tekstist, tihtipeale sisaldab see ka sõnade lauseosamärgendeid ja morfoloogilisi märgendeid. Algselt sai valitud UD keelekorpus, kuna seal on korrektsed lauseosa märgend ja morfoloogilised märgendind juba olemas. Eesti keele UD keelekorpuses on tekstid võetud ilukirjandusest, ajalehtedest, teadustekstidest ja HamleDT 3.0 keelepangast [5]. Andmestik koosneb 30723 lausest ning 434245 sõnast. Treeningandmestikus on 24384 lauset ja 341122 sõnast, testandmestikus 3214 lauset ja 48491 sõna ning valideerimisandmestikus 3125 lauset ja 44632 sõna. Töös nimetatakse seda andmestikku UD andmestikuks.

Töö käigus tundus, et ainult UD keelekorpuses olevatest andmetest võib väheks jääda, seetõttu võeti kasutusele ka osa UD *CoNLL 2017 Shared Task*'i andmestikust [6], kus eesti keele kohta oli 25162496 lauset ning 328307176 sõna. Töötlemata tekstis on võetud Common Crawl ja Wikipedia andmestikust ning töödeldud UDPipe'iga CoNLL-U formaati [6]. CoNLL 2017 koosneb 32'st Common Crawli failist ja kolmest

Wikipedia failist, kuid tagada, et mudelite treenimine toimuks mõistliku ajaga, võeti töös kasutusele üks Common Crawli fail. Siin töös kasutatakse sellest andmestikust 750780 lauset ning 10800889 sõna ning seda kasutatakse treeningandmetena. Seda andmestikku nimetatakse töös CoNLL-U 2017 andmestikuks.

UD keelekorpus on CoNLL-U formaadis. CoNLL-U formaadis tähistab „#" kommentaari ja tühi rida lause lõppu. Kõik lauses olevad sõnad on erineval real ning igal sõnal on 10 välja, kus on selle sõna kohta info. Iga välja tähendus on toodud tabelis Tabel 1. CoNLL-U failid näide on toodud alloleval joonisel (Joonis 6).

Tabel 1: CoNLL-U formaadi seletus [24]

Välja number	Välja tähis	Välja tähendus
1	ID	Mitmes sõnas lauses, algab igal lausel ühest.
2	FORM	Õiges vormis sõna või kirjavahemärk.
3	LEMMA	Sõna algvorm ehk lemma.
4	UPOS	Universaalne lauseosa märgend.
5	XPOS	Keelespetsiifiline lauseosa märgend.
6	FEATS	Loend morfoloogilistest märgenditest, eraldatud „|"-ga.
7	HEAD	Sõna ID, millest praegune sõna sõltub.
8	DEPREL	UD poolt kasutatav seos HEAD'ile
9	DEPS	Tõhustatud sõltuvusgraaf, mis on HEAD-DEPREL paari kujul.
10	MISC	Muu annotatsioon.

```
# sent_id = aja_ee199920_1477
# text = Aga mulle tundub, et kogu maailm ootab muusikamaailmalt midagi erutavalt uut minimalismi kõrvale.
1   Aga aga CCONJ   J   _   3   cc  _   _
2   mulle   mina    PRON    P   Case=All|Number=Sing|Person=1|PronType=Prs  3   obl _   _
3   tundub  tunduma VERB    V   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act 0   root    _   SpaceAfter=No
4   ,   ,   PUNCT   Z   _   8   punct   _   _
5   et  et  SCONJ   J   _   8   mark    _   _
6   kogu    kogu    DET A   PronType=Tot    7   det _   _
7   maailm  maa_ilm NOUN    S   Case=Nom|Number=Sing    8   nsubj   _   _
8   ootab   ootama  VERB    V   Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act 3   csubj   _   _
9   muusikamaailmalt    muusika_maa_ilm NOUN    S   Case=Abl|Number=Sing    8   obl _   _
10  midagi  miski   PRON    P   Case=Par|Number=Sing|PronType=Ind   8   obj _   _
11  erutavalt   erutavalt   ADV D   _   12  advmod  _   _
12  uut uus ADJ A   Case=Par|Degree=Pos|Number=Sing 10  amod    _   _
13  minimalismi minimalism  NOUN    S   Case=Gen|Number=Sing    8   obl _   _
14  kõrvale kõrvale ADP K   AdpType=Post    13  case    _   SpaceAfter=No
15  .   .   PUNCT   Z   _   3   punct   _   _
```

Joonis 6: näide CoNLL-U failist

4.1.1 Andmete uuesti annoteerimine

Töö tegemise käigus selgus, et UD korpuses olevaid märgendeid ei ole võimalik täpselt muundada sellisteks, et EstNLTK teegis neid kasutada saaks sünteesimiseks. Näiteks on UD korpuses sõna „kasutan" morfoloogilised märgendid „Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin|Voice=Act", kuid EstNLTK teegis on morfoloogiliseks märgendiks sellel sõnal „n". Sama probleem oli ka lauseosa märgenditega. Seetõttu annoteeriti käesolevas töös UD korpus uuesti, kasutades EstNLTK teeki.

Uuesti annoteerimisel asendati LEMMA, XPOS ja FEATS väljad CoNLL-U failis. EstNLTK'le anti sisendiks terve lause ning väljundiks saab objekti, kus on iga sõna lemma, morfoloogilised märgendid ja lauseosa märgend. LEMMA väli asendati EstNLTK poolt tagastava lemmaga, XPOS väli tagastatud lauseosa märgendiga ja FEATS tagastatud morfoloogiliste väljenditega.

Veendumaks, et uuesti annoteeritud faili täpsus ei langeks märkimisväärselt, kontrolliti uue faili täpsust. Kuna uuesti annoteeritud failis ei muudeta õiges vormis sõna, siis koostati programm, mis kontrollib kas uuest lemmast ja morfoloogilistest ning lauseosa märgenditest tuleb kokku õiges vormis sõna. Selleks kasutati taas EstNLTK teeki, kus on olemas sünteesimise võimalus. Sünteesimiseks läheb vaja lause lemmat ning morfoloogilisi väljendeid. Parema täpsuse saamiseks on võimalik ka lauseosa märgend lisada, mida ka tehti. Lauseosa märgendi puudumisel ei saa teek aru, kas näiteks sõna „täidetud" on tegusõna või omadussõna.

Programm tagastas uuesti annoteeritud faili täpsuseks 98,22%. Probleeme tekitasid gi- ja ki-liidetega sõnad ning nimed, mida EstNLTK sünteesida ei suutnud. Kui eeldada et sõnad, mida EstNLTK sünteesida ei suutnud, kuid morfoloogilised märgendid olid olemas, on korrektsed on täpsuseks 99,67%. Sellest võib järeldada, et faili annoteerimisel kriitilisi vigu ei tehtud ning neid saab kasutada närvivõrkude treenimiseks.

4.2 Mudelid

Töös treenitakse välja kolm mudelit. Kaks jadamärgendamiseks ja üks lemmadest õigesse vormi sõnade muutmiseks. Jadamärgendamiseks kasutati Stanfordi mudelit ning

standardset BiLSTM jadamärgendamise mudelit, lemmadest õiges vormides sõnadesse ennustamisel ehk sünteesimiseks Stanfordi mudelit. Siin peatükis kirjeldatakse täpsemalt mudelite arhitektuuri ning parameetreid, millega neid treeniti. Kõik mudelid treeniti Tallinna Tehnikaülikooli serveris, graafikakaardi peal. Graafikakaardiks oli Nvidia Tesla P100-PCIE-16GB.

4.2.1 Stanfordi jadamärgendamise mudel

Üheks jadamärgendamise mudeliks kasutatakse Stanfordi poolt implementeeritud mudelit. Stanfordi mudel on osa *„CoNLL 2018 UD Shared Task”* esitatud süsteemist [22]. *„CoNLL 2018 UD Shared Taski”* eesmärgiks oli luua süsteem, mis suudab õppida süntaktilisi sõltuvusi tüpoloogiliselt erinevates keeltes ja tehes seda reaalse elu kontekstis, andes sisendiks töötlemata teksti [25].

Üks osa, mida Stanfordi süsteemist siin töös kasutatakse on lauseosa ja morfoloogiliste märgendite ennustaja. Stanfordi töös ennustatakse lauseosa ja morfoloogilised väljendid juba õiges vormis olnud sõnast, kuid kuna siin töös on sisendiks lemma, siis modifitseeriti mudeli sisendit selliselt, et sisendsõnaks võetakse lemma.

Mudeli tuumaks on BiLSTM närvivõrk, mille sisend on kolmest allikast kokku pandud. Esimene allikas on eeltreenitud sõnavektorid ja nendest kasutatakse FastText sõnavektoreid. Teine allikas on treenitav sõnavektor, kus on sõnad, mis on vähemalt 7 korda esinenud treeningandmestikus. Kolmandaks allikaks on tähtede representatsioon, mis on treenitud ühesuunalise LSTM võrguga sõnas olevate tähtede järgi [22].

Lauseosa (CoNLL-u failis XPOS väli) ennustamiseks transformeeritakse BiLSTM’i iga sõna olekud täielikult ühendatud (FC) kihiga. Seejärel rakendatakse biafiinset klassifitseerijat. Biafiinne klassifitseerija tagab kokkusobivuse erinevate märgendite vahel, nimelt XPOS ja UPOS märgendite vahel. Morfoloogilisi märgendeid ennustatakse sarnaselt. Mudelit treenitakse minimiseerima ristentroopiakahju [22].

Mudel treeniti kasutades enamjaolt samu parameetreid (Tabel 2), mida kasutati ka Stanfordi poolt. Muudeti plokkide suurust, et maksimaalselt ära kasutada graafikakaardi ressurssi. Sõnade ja lemmade sõnavektoris hoiustamiseks kasutatakse 75-dimensioonilist vektorit, lauseosa ja morfoloogiliste märgendite hoiustamiseks 50 dimensioonilist vektorit. Eeltreenitud sõnavektorid transformeeritakse 125-

dimensiooniliseks vektoriks, sama tehakse ka tähepõhise sõna esituse andmetega. Treenimise ajal muudetakse kõikides sõnavektorites 33% tõenäosusega sõna „<*drop*>"'iks. Jadamärgendamiseks kasutatakse 2-kihilist ja 200-dimensioonilist BiLSTM võrku. Kõigis närvivõrkudes kasutatakse väljajätumeetodit tõenäosusega 50% [22].

Tabel 2: Stanfordi jadamärgendaja parameetrid

Parameetri tähis	**Parameetri tähendus**	**Parameetri väärtus**
hidden_dim	Varjatud kihi suurus	200
char_hidden_dim	Tähtede varjatud kihi suurus	400
deep_biaff_hidden_dim	Biafiinse klassifitseerija suurus	400
composite_deep_biaff_hidden_dim	Biafiinse klassifitseerija suurus	100
word_emb_dim	Sõnavektori suurus	75
char_emb_dim	Tähtede representatsiooni suurus	100
tag_emb_dim	Märgendite suurus	50
transformed_dim	Transformeeritud kihi suurus	125
num_layers	Kihtide arv BiLSTM võrgus	2
char_num_layers	Kihtide arv tähtede	1
word_dropout	Sõnamaatriksi dropout määr	0.33
dropout	Dropouti määr	0.5
rec_dropout	Rekurrentsete ühenduste *dropout* määr	0
char_rec_dropout	Tähtede representatsiooni *dropout* määr	0
no_char	Ei kasutata täherepresentatsiooni	False
no_pretrain	Ei kasutata eeltreenitud sõnavektoreid	False
optim	Optimeerija tüüp	adam
lr	Õpisammu suurus	$3*10^{-3}$
beta2	Adami optimeerija *beta2* väärtus	0.95
batch_size	Ploki suurus	5000

4.2.2 BiLSTM jadamärgendaja

Teise jadamärgendamise mudelina kasutatakse lihtsama arhitektuuriga BiLSTM võrku, mida siin töös nimetatakse BiLSTM jadamärgendajaks. Mudel pärineb Kairit Sirtsi GIT'i repositooriumist[1]. Mudel vastab arhitektuurile, mida on selgitatud jaotises 2.3.

Võrgu sisendiks on eeltreenitud sõnavektorid, mida treenimise käigus täpsustatakse (ingl k *fine-tuning*) ja tähtedest BiLSTM representatsioon. Närvivõrk on ühekihiline ning selle otsas on *soft-max* kiht. Mudeli parameetrid on toodud välja allolevas tabelis (Tabel 3).

Tabel 3: BiLSTM jadamärgendaja parameetrid

Parameetri tähis	Parameetri tähendus	Parameetri väärtus
batch-size	Ploki suurus	300
word-emb	Sõnavektori suurus	300
hidden-dim	Varjatud oleku kihi suurus	300
char-emb	Tähe representatsiooni suurus	75
char-hidden	Tähe representatsiooni varjatud oleku suurus	75

4.2.3 Stanfordi lemmast õiges vormis sõna ennustaja

Teine mudel, mida Stanfordi süsteemist kasutatakse on lemmatiseerija (ingl k *lemmatizer*). Stanfordi süsteemis kasutati seda muutmaks õiges vormis sõna lemmaks. Sisendiks on õiges vormis sõna ning selle lauseosa märgend ning väljundiks lemma. Siin töös muudeti mudelit selliselt, et sisendiks on lemma ning morfoloogilised märgendid ning väljundiks õiges vormis sõna. Lauseosa märgendi muutmine morfoloogilisteks väljendiks tulenes sellest, et eesti keeles on raske õiges vormis sõna ennustada puhtalt lauseosa teades.

Mudel on kooder-dekooder tüüpi. Kooder on BiLSTM võrk, millel on tähelepanu mehhanism. Mudeli sisendiks on jada tähtedest ning väljund on samuti jada tähtedest. RNN'ide sisendid on kodeeritud jagatud maatriksiga tähtede representatsioonist (E). Kui kooderi varjatud kiht (h^{enc}) on kätte saadud kasutades ühekihilist BiLSTM võrku, siis iga dekooderi samm näeb välja selline.

[1] Tegemist on privaatse GIT'i repositooriumiga, mis asub aadressil https://github.com/kairit/embeddings

$$h_j^{dec} = LSTM_{dec}\left(E_{y_{j-1}}, h_{j-1}^{dec}\right),$$

$$\alpha_{ij} \propto \exp\left(u_\alpha^T \tanh\left(w_\alpha [h_j^{dec}, h_i^{enc}]\right)\right)$$

$$c_j = \sum_i \alpha_{ij} h_i^{enc}$$

Ennustamisel kasutatakse enne närvivõrku ka sõnastikku, mis koostatakse treeningandmetest. Sõnastikus on lemma ja morfoloogiliste märgendite paar, mis vastab õiges vormis sõnale. Kui sõnastikust vastet ei leita, kasutatakse närvivõrku õige sõna ennustamiseks [22].

Mudel treeniti kasutades enamjaolt samu parameetreid (Tabel 4), mida kasutati ka Stanfordi poolt. Muudeti plokkide suurust, et saada maksimaalne ressurssi kasutus graafikakaardist. Kooderis kasutatakse BiLSTM võrke, mille igas suunas on 200-dimensioonilised varjatud olekud. Sisendiks on 50-dimensiooniline tähepõhine representatsioon ning väljajätu määraks on 0,5. Dekoooderiks on ühesuunaline LSTM, millel on 200-dimensiooniline varjatud olekute kiht. Mudelit treenitakse standardsete Adam'i hüperparameetritega.

Tabel 4: Stanfordi sünteesija parameetrid

Parameetri tähis	Parameetri tähendus	Parameetri väärtus
hidden_dim	Varjatud kihi suurus	200
emb_dim	Embeddingu kihi suurus	150
num_layers	Kihtide arv	1
emb_dropout	Embeddingu dropout	0.5
dropout	Mudeli dropout	0.5
max_dec_len	Maksimaalne dekoooderi väljundi pikkus	50
beam_size	Beamide arv	1

4.2.4 Tulemuste hindamine

Tulemuste hindamiseks kasutatakse õigsust. Vaadatakse mitu sõna kogu sõnade hulgast on õigesti ennustatud. Stanfordi töös öeldakse, et kasutatakse F1 skoori, kuid kui implementatsioonist järele vaadata, on Stanfordi F1 valem selline, nagu näidatud Joonis

7. Kuna õigete sõnade arv ja ennustatud sõnade arv on võrdsed, siis taandub valem õigesti ennustatud sõnade arvu ja kogu sõnade arvu jagatiseks.

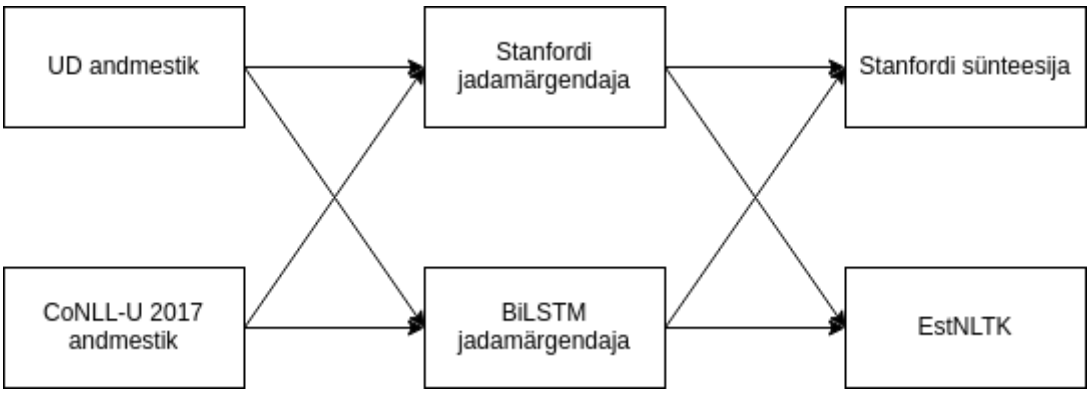
$$F_1 = \frac{2 * õigest\ ennustatud\ sõnade\ arv}{õigete\ sõnade\ kogu\ arv + ennustatud\ sõnad\ koguarv}$$

Joonis 7: Stanfordi F1 valem

F1 skoor näitab harmoonilist keskmist täpsusest ja saagisest. Täpsust näitab suhet õigesti positiivselt ennustatud tulemuste ja kogu positiivselt ennustatud tulemuste vahel ja saak näitab suhet õigesti positiivselt ennustatud tulemuste ja tegeliku positiivsete tulemuste vahel [21]. F1 skoori kasutatakse binaarsete klassifitseerijatega, kuid kuna siin töös ega Stanfordi töös binaarseid klassifitseerijaid ei kasutatud, ei teki arusaama, miks Stanfordi teadlased oma töös F1 skoori kasutasid või et miks nad nimetasid seda F1 skooriks.

5 Eksperimendid

Siin peatükis antakse ülevaade tulemustest ning analüüsitakse neid. Mõlemat jadamärgendajat treeniti nii UD, kui ka CoNLL-U 2017 andmestikuga ning sünteesijat ainult UD andmestikuga. Andmestiku ülevaade on peatükis 4.1. Alguses treeniti ja hinnati kõiki mudeleid treening- ja valideerimise andmestiku peal. Seejärel võetakse jadamärgendamise mudelite ennustused testandmestiku pealt ja ennustatakse nende järgi sõnu õigesse vormi, et saada kogu süsteemi toimimise kohta ülevaade. Joonisel (Joonis 8) on kujutatud parima terviksüsteemi leidmise meetod. Kõikidest mudelitest võeti parima tulemuse saanud mudel ning proovitakse kõik võimalikud terviksüsteemi kombinatsioonid läbi ning leitakse parim tulemus.



Joonis 8: Parima terviksüsteemi leidmine

5.1 Jadamärgendamise tulemused

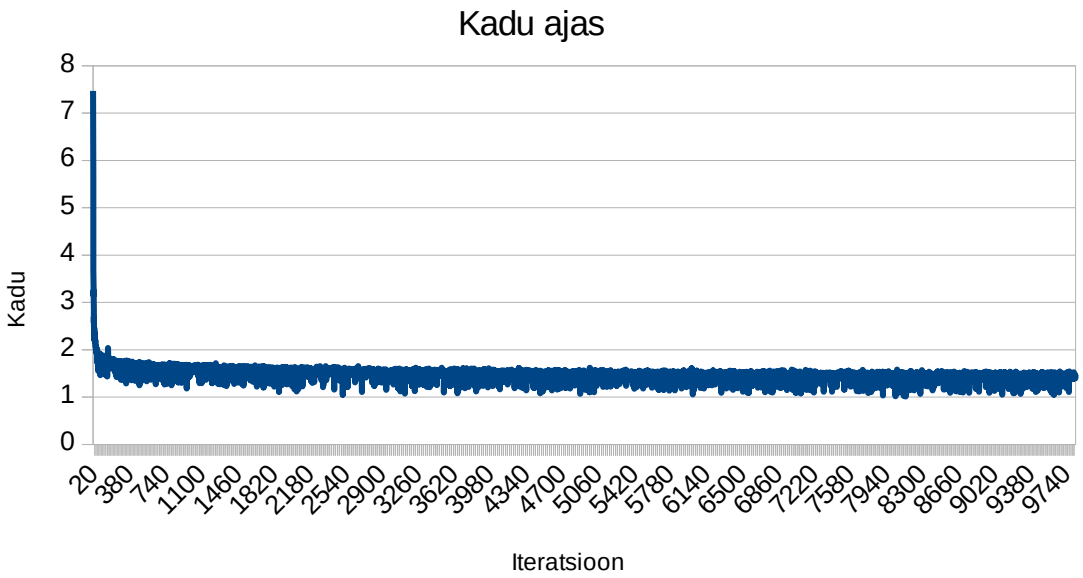
Mõlemat jadamärgendamise mudelit treeniti nii UD, kui ka CoNLL-U 2017 andmestikuga. Allolevas tabelis (Tabel 5) on välja toodud nii testandmestikus (tabelis „*test*"), kui ka valideerimisandmestikus (tabelis „*dev*") saavutatud tulemused.

Tabel 5: Jadamärgendamise tulemused

Andmestik	Stanfordi mudel		BiLSTM mudel	
	dev	test	dev	test
UD	72,51	73,80	70,10	71,71
CoNLL-U 2017	69,22	70,85	73,51	75,88

Parima tulemuse testandmestiku peal saavutas BiLSTM mudel, mis oli treenitud CoNLL-U 2017 andmestiku peal. Tabelist on näha ka, et kui BiLSTM mudeli puhul treeningandmestiku suurenemine mõjutas tulemust positiivselt, siis Stanfordi mudeli täpsus langes.

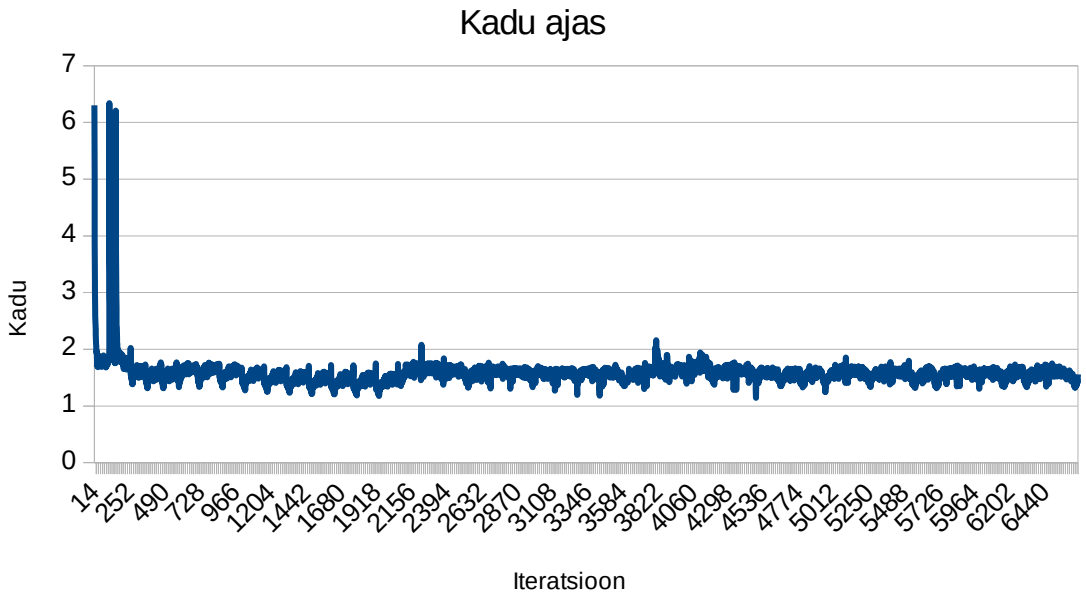
Stanfordi jadamärgendamise mudelit treeniti UD andmestikuga 1365 epohhi ehk kogu treeningandmestikust käidi 1365 korda üle. Tulemusi valideerimisandmestiku peal hinnati pärast igat 22 epohhi. Parim tulemus oli 72,51, mis saavutati pärast 1232. epohhi. Joonisel (Joonis 9) on näidatud kadu treeningu jooksul, kus iga iteratsioon tähistab kümne ploki treenimist.



Joonis 9: Stanfordi jadamärgendamise mudeli kadu treenimise jooksul UD andmestikuga.

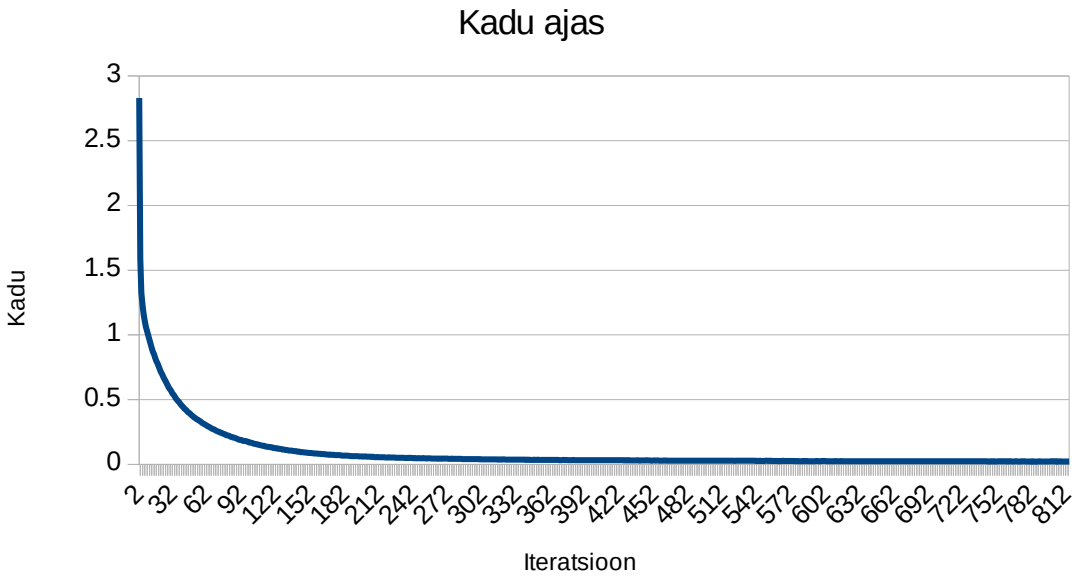
CoNLL-U 2017 andmestiku peal treeniti mudelit 42 epohhi. Tulemusi hinnati valideerimisandmestiku peal pärast igat epohhi. Parim tulemus oli 69.22, mis saavutati

pärast 40. epohhi. Joonisel (Joonis 10) on näidatud kadu treenimise jooksul, kus iga iteratsioon tähistab kümne ploki treenimist.



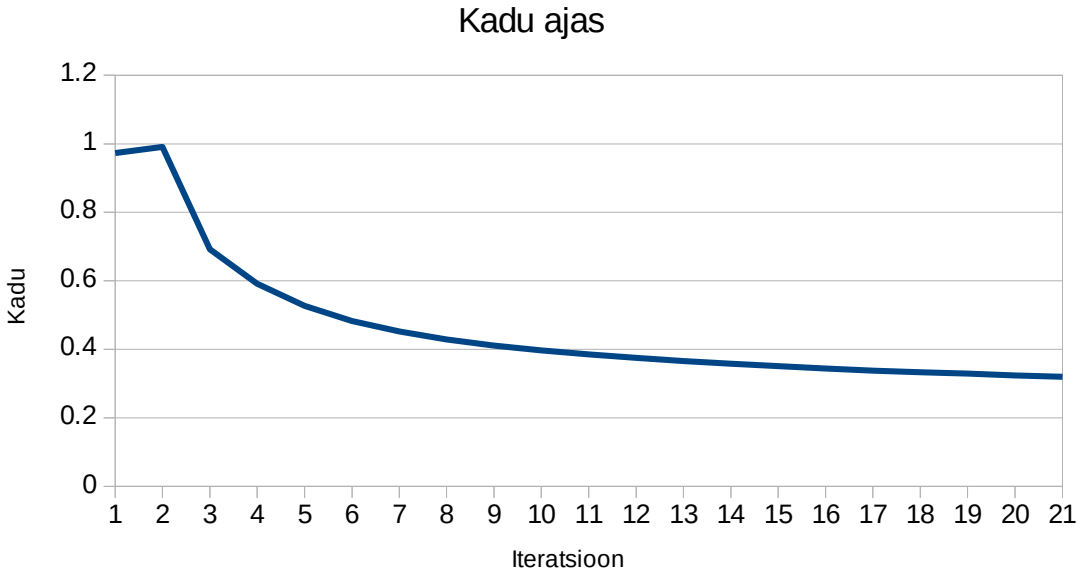
Joonis 10: Stanfordi jadamärgendamise mudeli kadu treenimise jooksul CoNLL-U 2017 andmestikuga

BiLSTM jadamärgendamise mudelit treeniti UD andmestiku peal 813 epohhi. Tulemusi valideerimisandmestiku peal hinnati pärast igat epohhi. Parim tulemus oli 70.10, mis saavutati pärast 87. epohhi. Joonisel (Joonis 11) on näidatud kadu treenimise jooksul, kus iga iteratsioon tähistab 300 ploki treenimist.



Joonis 11: BiLSTM jadamärgendamise mudeli kadu treenimise jooksul UD andmestikuga

CoNLL-U 2017 andmestikuga treeniti mudelit 21 epohhi. Tulemusi valideerimisandmestiku peal hinnati pärast igat epohhi. Parim tulemus oli 73.51, mis saavutati pärast 9. epohhi. Joonisel (Joonis 12) on näidatud kadu treenimise jooksul, kus iga iteratsioon tähistab 300 ploki treenimist.



Joonis 12: BiLSTM jadamärgendamise mudeli kadu treenimise jooksul CoNLL-U 2017 andmestikuga

5.1.1 Tulemuste analüüs

Siin peatükis vaadatakse, milliseid morfoloogilisi ja lauseosa märgendeid mudel ennustas ning kui neid märgendeid rakendada lemmale, kas tulemuseks on korrektne eestikeelne lause või mitte. Siin jaotises rakendatakse märgendid lemmale autori poolt, mitte töös loodud sünteesija ega EstNLTK poolt. Stanfordi jadamärgendaja puhul analüüsitakse UD andmestikuga treenitud mudeli tulemusi, kuna see andis parema tulemuse testandmestiku peal ning BiLSTM jadamärgendaja puhul CoNLL-U 2017 andmestikuga treenitud mudeli järgi, kuna see andis parema tulemuse.

Tulemuste illustreerimiseks võetakse vaatluse alla kaks lauset valideerimisandmestikust. Esimene lause on „Aga mulle tundub, et kogu maailm ootab muusikamaailmalt midagi erutavalt uut minimalismi kõrvale”, mis on üsnagi keeruline liitlause ning „Hommik algas taas ettekannetega”, mis on lihtlause. Esimene lause lemmade kujul on „aga mina tunduma, et kogu maailm ootama muusikamaailm miski erutavalt uus minimalism kõrvale.” ning teine lause on „hommik algama taas ettekanne.”

Kõigepealt vaadatakse, millised on korrektsed märgendid, mida rakendades tuleb korrektne eestikeelne lause, need on toodud allolevates tabelites (Tabel 6 ja Tabel 7).

Tabel 6: Esimese lause korrektsed märgendid

Lemma	Lauseosa	Morfoloogia	Sõna
aga	J	_	aga
mina	P	Number=sg| Case=all	mulle
tunduma	V	VerbForm=b	tundub
,	Z	_	,
et	J	_	et
kogu	A	_	kogu
maailm	S	Number=sg|Case=n	maailm
ootama	V	VerbForm=b	ootab
muusikamaailm	S	Number=sg| Case=abl	muusikamaailmalt
miski	P	Number=sg|Case=p	midagi
erutavalt	D	_	erutavalt
uus	A	Number=sg|Case=p	uut
minimalism	S	Number=sg|Case=p	minimalismi

Lemma	Lauseosa	Morfoloogia	Sõna
kõrvale	K	_	kõrvale
.	Z	_	.

Tabel 7: Teise lause korrektsed märgendid

Lemma	Lauseosa	Morfoloogia	Sõna
hommik	S	Number=sg|Case=n	hommik
algama	V	VerbForm=s	algas
taas	D	_	taas
ettekanne	S	Number=pl| Case=kom	ettekannetega
.	Z	_	.

Nii Stanfordi mudel, kui ka BiLSTM mudel ennustas kõik lauseosa märgendid korrektselt, seetõttu jäeti see veerg ennustuste tabelitest välja. Stanfordi mudeli morfoloogiliste märgendite ennustamise tulemused on toodud tabelis (Tabel 8). On näha, et kõik sõnad, millel pole morfoloogilisi märgendeid määratud ehk sõnad mida ei pea muutma on õigesti ennustatud, kuid ainult kahel juhul kaheksast on märgendid õigesti ennustatud, kui sõnu peab muutma. Kui märgendid rakendada lemmadele, ei tule kokku loogiline lause.

Tabel 8: Stanfordi mudeli poolt ennustatud esimese lause morfoloogilised märgendid

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
aga	_	_	aga
mina	Number=sg| Case=all	Case=ad|Number=sg	mul
tunduma	VerbForm=b	VerbForm=s	tundus
,	_	_	,
et	_	_	et
kogu	_	_	kogu
maailm	Number=sg|Case=n	Case=in|Number=sg	maailmas
ootama	VerbForm=b	VerbForm=b	ootab
muusikamaailm	Number=sg| Case=abl	Case=n|Number=sg	muusikamaailm
miski	Number=sg|Case=p	Number=sg|Case=p	midagi
erutavalt	_	_	erutavalt

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
uus	Number=sg|Case=p	Case=g|Number=sg	uue
minimalism	Number=sg|Case=p	Case=g|Number=sg	minimalismi
kõrvale	_	_	kõrvale
.	_	_	.

Teise lause ennustused (Tabel 9) näitavad, et kaks kolmest morfoloogilistest märgenditest, kus sõna peab muutma, on valed. Erinevus esimese lausega on see, et kokku tuleb täiesti loogiline lause. Sellest võib järeldada, et ainult lemmade järgi ei suuda mudel samamõttelist lauset moodustada, kuid suudab moodustada lause.

Tabel 9: Stanfordi mudeli poolt ennustatud teise lause morfoloogilised märgendid

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
hommik	Number=sg|Case=n	Number=sg|Case=ad	hommikul
algama	VerbForm=s	VerbForm=s	algas
taas	_	_	taas
ettekanne	Number=pl| Case=kom	Number=sg|Case=n	ettekanne
.	_	_	.

Võrreldes Stanfordi jadamärgendamise mudeliga on näha, et BiLSTM mudel on morfoloogiliste märgendite ennustamisel on õigesti ennustatud ühe märgendi rohkem (Tabel 10). Kui vaadata kogu lauset, siis ei ole see loogiline ning on vähem loogiline kui Stanfordi poolt ennustatud lause.

Tabel 10: BiLSTM mudeli poolt ennustatud morfoloogilised märgendid esimesele lausele

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
aga	_	_	aga
mina	Number=sg| Case=all	Case=all| Number=sg	mulle
tunduma	VerbForm=b	VerbForm=s	tundus
,	_	_	,
et	_	_	et

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
kogu	_	_	kogu
maailm	Number=sg|Case=n	Case=in|Number=sg	maailmas
ootama	VerbForm=b	VerbForm=b	ootab
muusikamaailm	Number=sg| Case=abl	Case=in|Number=sg	muusikamaailmas
miski	Number=sg|Case=p	Number=sg|Case=p	midagi
erutavalt	_	_	erutavalt
uus	Number=sg|Case=p	Case=g|Number=sg	uue
minimalism	Number=sg|Case=p	Case=g|Number=sg	minimalismi
kõrvale	_	_	kõrvale
.	_	_	.

Tabelist (Tabel 11) on näha, et kõik ennustused BiLSTM'i mudeli poolt on samad, mis Stanfordi jadamärgendajaga saadi. Lause on loogiliselt korrektne, kuid ei vasta tahetule.

Tabel 11: BiLSTM mudeli poolt ennustatud morfoloogilised märgendid teisele lausele

Lemma	Õiged morf. märgendid	Ennustatud morf. märgendid	Sõna
hommik	Number=sg|Case=n	Number=sg|Case=ad	hommikul
algama	VerbForm=s	VerbForm=s	algas
taas	_	_	taas
ettekanne	Number=pl| Case=kom	Number=sg|Case=n	ettekanne
.	_	_	.

5.2 Morfoloogilise sünteesimise tulemused

Stanfordi lemmast õige sõna ennustaja mudelit treeniti töö suure mahu tõttu ainult UD andmestikuga ning esialgsed katsetused CoNLL-U 2017 andmestikuga näitasid, et märkimisväärset tõusu täpsuses pole oodata. Mudeli täpsust vaadati nii juhul, kui kasutusel oli koostatud sõnastik, mis koosnes lemma ja morfoloogiliste märgendite paarist, mis olid vastavuses õiges vormis sõnaga ning ka seda sõnastikku mitte

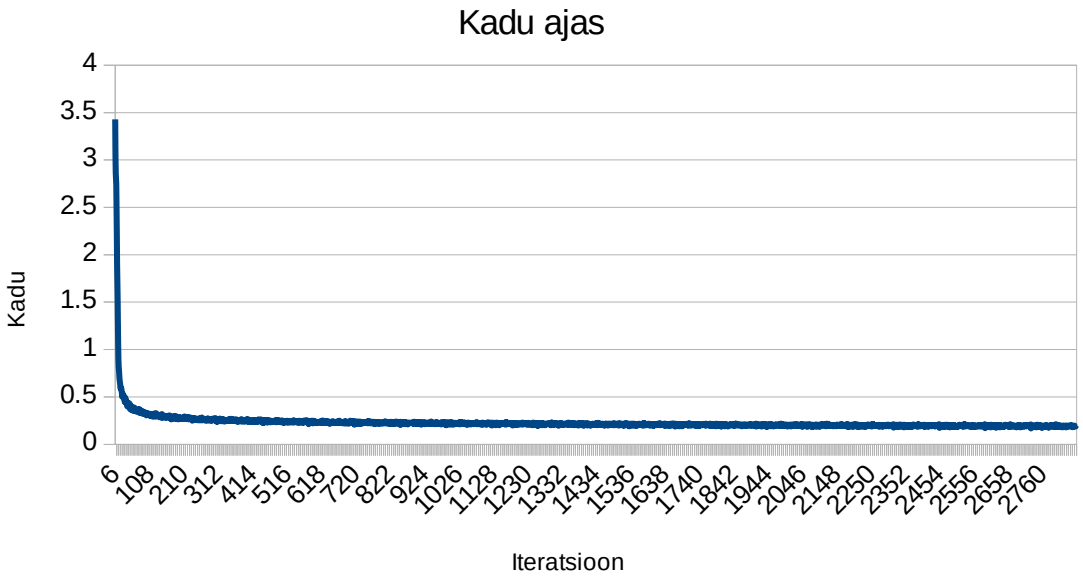
kasutades. Lisaks vaadatakse, mis mõju annab kui võtta kõige tõenäolisema ennustuse asemel suurem arv tõenäolisemaid ennustusi. EstNLTK paketi puhul mõõdeti täpsust kasutades sama loogikat, mis jaotises 4.1.1.

Tabel 12: Morfoloogilise sünteesimise tulemused

Andmestik	**Stanfordi sünteesija**				**EstNLTK**	
	Sõnastik + Närvivõrk		Närvivõrk			
	dev	test	dev	test	dev	test
UD	86,64	88,15	58,46	59,49	98,22	98,46

Morfoloogilise sünteesimise tulemused on toodud tabelis (Tabel 12). Parima tulemuse saavutas EstNLTK teegi sünteesija, mis on ka oodatud tulemus, kuna andmed on annoteeritud kasutades sama teeki. See tõestab, et teek töötab nagu oodatud. Stanfordi sünteesija sai testandmestiku peal täpsuseks 88,15, mis on oodatust parem tulemus, kuna originaalne lemmatiseerija sai tulemuseks 91,24 [22] ning võis arvata, et selle teistpidi tööle panemine (lemmast õiges vormis sõna sünteesimine, mitte õigest sõnast lemma sünteesimine) langetab täpsust rohkem.

UD andmestikuga treeniti mudelit 84 epohhi ning mudelit hinnati pärast igat epohhi. Parim tulemus valideerimisandmestiku peal ilma sõnastikuta saavutati pärast 51. epohhi ning selleks oli 58.46. Koos sõnastikuga oli valideerimisandmestiku peal täpsus 86,64. Joonisel (Joonis 13) on näidatud kadu treenimise jooksul, kus iga iteratsioon tähistab 10 ploki treenimist.



Joonis 13: Stanfordi sünteesija kadu treenimisel UD andmestiku peal

5.2.1 Tulemuse analüüs

Lauseid millega illustreeritakse tulemusi on samad, mis peatükis 5.1.1. Morfoloogilisteks ja lauseosa märgenditeks kasutati korrektseid märgendeid, et saada teada maksimaalselt hea tulemus. Samuti kasutatakse tulemusi, mille ennustas mudel, mis kasutas sõnastikku.

Tabel 13: Ennustatud sõnad esimeses lauses

Lemma	Morfoloogia	Sõna
aga	_	aga
mina	Number=sg|Case=all	mulle
tunduma	VerbForm=b	tundub
,	_	,
et	_	et
kogu	_	kogu
maailm	Number=sg|Case=n	maailm
ootama	VerbForm=b	ootab
muusikamaailm	Number=sg|Case=abl	muusikamaailma
miski	Number=sg|Case=p	midagi

Lemma	Morfoloogia	Sõna
erutavalt	_	erutavalt
uus	Number=sg|Case=p	uut
minimalism	Number=sg|Case=p	minimalism
kõrvale	_	kõrvale
.	_	.

Tabelist (Tabel 13) on näha, et ainukesed vead tehti sõnades, mida ei esine testandmestikus tihti ning mida sõnastik ei kata. Sõna „muusikamaailm" testandmestikus mis iganes vormis ei esine ning sõna minimalismi esineb testandmestikus ainult 30 korda.

Tabel 14: Ennustatud sõnad teises lauses

Lemma	Morfoloogia	Sõna
hommik	Number=sg|Case=n	hommik
algama	VerbForm=s	algas
taas	_	taas
ettekanne	Number=pl|Case=kom	ettekannetega
.	_	.

Teise lause ennustused (Tabel 14) on kõik korrektsed. See tuleneb sellest, et kõik sõnad selles vormis on esindatud ka koostatud sõnastikus.

5.2.2 Suurem arv tõenäolisemaid ennustusi

Siin peatükis vaadatakse, kuidas muutub täpsus, kui võtta kõige tõenäolisema sõna asemel suurem hulk tõenäolisemaid sõnu ning kontrollida, kas nende seas on ka õige sõna. See annab aimu, kas lisaarendus olemasolevale mudelile võiks tõsta mudeli täpsust.

Tabel 15: Mudeli täpsus, kui suurendada tõenäolisemate sõnade arvu

Sõnade arv	Valideerimisandmestik	Testandmestik
2	88,77	90,10
5	91,30	92,24
10	92,51	93,26

Tabel (Tabel 15) näitab, et antud mudelit on võimalik parendada, kui teha lisaarendusi. Lisaarendusteks võiks olla näiteks mudeli täiendamine, et võtta paremini arvesse morfoloogilisi märgendeid. Teiseks lisaarenduseks võiks olla uue keelemudeli treenimine, mis suudaks mitme ennustatud sõnavormi seast välja valida õiges vormis sõna.

5.3 Terviksüsteemi tulemused

Siin peatükis ühildatakse mudelite ennustused terviksüsteemiks ja vaadatakse, kui täpsed on mudelid koos töötades. Mudeleid hinnatakse testandmestiku peal. Esiteks võeti Stanfordi jadamärgendaja mudeli ennustused ja ennustati Stanfordi lemmast õiges vormis sõna ennustajaga õige sõna. Seejärel võeti uuesti Stanfordi jadamärgendaja mudeli ennustused ning rakendati need kasutades EstNLTK teeki. Lõpuks võeti lihtsama BiLSTM jadamärgendaja mudeli ennustused ning rakendati neid EstNLTK teegiga ja Stanfordi lemma sünteesijaga.

Tabel 16: Terviksüsteemi tulemused testandmestikul

Stanfordi jadamärgendaja ja Stanfordi lemma sünteesija	Stanfordi jadamärgendaja ja EstNLTK	BiLSTM jadamärgendaja ja Stanfordi sünteesija	BiLSTM jadamärgendaja ja EstNLTK
63,19	73,60	72,42	74,39

Tabelist (Tabel 16) on näha, et parima tulemuse saavutas BiLSTM jadamärgendaja ja EstNLTK teegist koosnev terviksüsteem. See tulemus oli oodatav, sest mõlemad mudelid said ka individuaalselt parimad tulemused. Üllatavam tulemus oli BiLSTM jadamärgendaja ja Stanfordi sünteesija kõrge tulemus. Kui Stanfordi jadamärgendaja juures oli EstNLTK teegiga sünteesides tulemus parem 10,41 võrra võrreldes Stanfordi sünteesijaga, siis BiLSTM puhul oli see vaid 1,97 võrra parem.

Tulemuste illustreerimiseks valitakse kaks lauset andmestikust ja võrreldakse kuidas neid sünteesitud on. Esimene lause on „Palju olulisi komponente, nagu liha ja kala, hangime siiski Eestist.” ning teine lause „Loomulikult kuuluvad meie kohalikku ostusedelisse ka aedviljad.”. BiLSTM märgendajat ja EstNLTK teeki kasutav süsteem

sünteesis esimese lause „Paljud olulisi komponente, nagu liha ja kala, hangitakse siiski Eesti.” ja teise lause „Loomulikult kuulub me kohalikku ostusedel ka aedviljad.”. BiLSTM märgendaja ja Stanfordi sünteesija sai esimeseks lauseks „Paljud olulisi komponente, nagu liha ja kala, hangitakse siiski Eestis.” ja teiseks lauseks „Loomulikult kuulub meie kohalikku ostusedel ka aedvili.”. Stanfordi märgendaja ja EstNLTK esimese lause tulemuseks oli „Paljud olulisi komponente, nagu liha ja kalad, hangivad siiski Eestis.” ning teise lause tulemuseks oli „loomulikult kuulub me kohalikud ostusedelid ka aedvilju.”. Stanfordi märgendaja ja Stanfordi sünteesija esimese lause tulemuseks oli „Palju olulise komponent, nagu liha ja kala, hankis siiski Eesti.” ja teise lause tulemuseks „Loomulikult kuulub meie kohalik ostusedel ka aedvili.”. Tulemustest on näha, et BiLSTM jadamärgendajat kasutanud süsteemid on täpsemad ning loogilisemad.

6 Kokkuvõte

Töö eesmärgiks oli luua tekstisüntesaator, mille sisendiks on jada lemmadest, mis moodustavad lause ning see jada sünteesida korrektseks eestikeelseks lauseks. Selle teostamiseks jagati ülesanne kahte ossa. Esiteks tuli lemmadele määrata lauseosa märgend ja morfoloogilised märgendid, mis näitavad, mis tüüpi sõna on ja mis käändes või pöördes sõna on. Teiseks tuli määratud märgendite abil sünteesida lemma õiges vormis sõnaks.

Märgendite määramiseks katsetati kahte rekurrentsetel närvivõrkudel töötavat mudelit: Stanfordi jadamärgendamise mudelit ning standardset BiLSTM'il põhinevat jadamärgendamise mudelit. Lemma sünteesimiseks kasutati Stanfordi lemmatiseerijat, mis kohandati selliseks, et õiges vormis sõnast lemma ennustamise asemel ennustati lemmast õiges vormis sõna. Alternatiivse lahendusena kasutati lemma sünteesimiseks ka EstNLTK teeki.

Treenimiseks kasutati kahte tekstikorpust, üks neist, Universal Dependencies'i korpus, oli väiksema mahuga ja CoNLL-U 2017 korpus oli suurema mahuga. Mõlemat jadamärgendamise mudelit treeniti nii UD, kui ka CoNLL-U 2017 andmetega. Stanfordi lemma sünteesijat treeniti töö suure mahu tõttu ainult UD andmestikuga. Parimaks jada märgendajaks oli CoNLL-U andmestikuga treenitud standardne BiLSTM'il põhinev jada märgendaja, mille täpsus valideerimisandmestiku peal oli 75,88. Stanfordi lemma sünteesija täpsuseks valideerimisandmestiku peal oli 86,64, kui täpsust arvestada kõige tõenäolisema sõna järgi. Kui vaadata kas õige sõna on kümne kõige tõenäolisema sõna hulgas, kasvas täpsus 92,51'ni. EstNLTK teegi täpsus oli 98,46. Kokkuvõttes saavutati tekstisüntesaatori parimaks täpsuseks valideerimisandmestiku peal 74,39.

Kõneabitarkvaras kasutamiseks pakub autor välja, et kõnesüntesaatorit tuleks täiendada. Jadamärgendajat annaks täiendada, kui lisada funktsionaalsus, mis lisaks lemmadele võimaldaks sisendiks anda ka lisa märgendeid, mis tähistaks näiteks, kas lause on minevikus või käskivas kõneviisis. Sünteesija täpsuse suurendamiseks pakub autor

tehtud töö põhjal välja, et ennustada tuleks kümme kõige tõenäolisemat sõna ning treenida välja keelemudel, mis suudab nende sõnade seast välja valida õige sõna.

Kasutatud kirjandus

[1] „Augmentative and Alternative Communication: Key Issues“, *ASHA*. [Online]. Available at: https://www.asha.org/PRPSpecificTopic.aspx?folderid=8589942773&section=Key_Issues. [Vaadatud: 09-mai-2019].

[2] Y. Goldberg, „A Primer on Neural Network Models for Natural Language Processing“, okt 2015.

[3] *Official Stanford NLP Python Library for Many Human Languages: stanfordnlp/stanfordnlp*. Stanford NLP, 2019.

[4] „Estnltk“. [Online]. Available at: https://estnltk.github.io/. [Vaadatud: 09-mai-2019].

[5] „Universal Dependencies“. [Online]. Available at: https://universaldependencies.org/. [Vaadatud: 09-mai-2019].

[6] „CoNLL 2017 Shared Task“. [Online]. Available at: http://universaldependencies.org/conll17/data.html. [Vaadatud: 09-mai-2019].

[7] „ISAAC – What is AAC?“ [Online]. Available at: https://www.isaac-online.org/english/what-is-aac/. [Vaadatud: 09-mai-2019].

[8] „Augmentative and Alternative Communication: Overview“, *American Speech-Language-Hearing Association*. [Online]. Available at: https://www.asha.org/Practice-Portal/Professional-Issues/Augmentative-and-Alternative-Communication/. [Vaadatud: 09-mai-2019].

[9] „[PSV] Eesti keele põhisõnavara sõnastik“. [Online]. Available at: http://eki.ee/dict/psv/index.cgi. [Vaadatud: 09-mai-2019].

[10] A. Graves, A. Mohamed, ja G. Hinton, „Speech recognition with deep recurrent neural networks“, *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, lk 6645–6649.

[11] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, ja W. Xu, „CNN-RNN: A Unified Framework for Multi-Label Image Classification“, esitatud Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, lk 2285–2294.

[12] K. Cho *et al.*, „Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation“, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, lk 1724–1734.

[13] „Text Generation with Recurrent Neural Networks (RNNs)“, *Hello Paperspace*, 07-apr-2017. [Online]. Available at: https://blog.paperspace.com/recurrent-neural-networks-part-1-2/. [Vaadatud: 09-mai-2019].

[14] „Neural Networks, Types, and Functional Programming -- colah’s blog“. [Online]. Available at: http://colah.github.io/posts/2015-09-NN-Types-FP/. [Vaadatud: 09-mai-2019].

[15] Y. Goldberg, „Neural Network Methods for Natural Language Processing“, *Synthesis Lectures on Human Language Technologies*, kd 10, nr 1, lk 1–309, apr 2017.

[16] I. Sutskever, O. Vinyals, ja Q. V. Le, „Sequence to Sequence Learning with Neural Networks“, *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, ja K. Q. Weinberger, Toim Curran Associates, Inc., 2014, lk 3104–3112.

[17] S. Orasmaa, T. Petmanson, A. Tkachenko, S. Laur, ja H.-J. Kaalep, „ESTNLTK - NLP Toolkit for Estonian“, lk 7.

[18] Heiki-Jaan Kaalep ja Tarmo Vaino, „Complete Morphological Analysis in the Linguist’s Toolbox“. .

[19] S. Dudy ja S. Bedrick, „Compositional Language Modeling for Icon-Based Augmentative and Alternative Communication“, *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, Melbourne, 2018, lk 25–32.

[20] C. Conforti, M. Huck, ja A. Fraser, „Neural Morphological Tagging of Lemma Sequences for Machine Translation“, *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, Boston, MA, 2018, lk 39–53.
[21] D. Hand ja P. Christen, „A note on using the F-measure for evaluating record linkage algorithms“, *Stat Comput*, kd 28, nr 3, lk 539–547, mai 2018.
[22] P. Qi, T. Dozat, Y. Zhang, ja C. D. Manning, „Universal Dependency Parsing from Scratch“, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Brussels, Belgium, 2018, lk 160–170.
[23] „All treebanks“. [Online]. Available at: https://universaldependencies.org/conll18/results-lemmas.html. [Vaadatud: 09-mai-2019].
[24] „CoNLL-U Format“. [Online]. Available at: https://universaldependencies.org/format.html. [Vaadatud: 09-mai-2019].
[25] „CoNLL 2018 Shared Task“. [Online]. Available at: https://universaldependencies.org/conll18/. [Vaadatud: 09-mai-2019].
[26] „EKI Pildilehed“. [Online]. Available at: http://www.eki.ee/dict/psv/pildilehed.pdf. [Vaadatud: 09-mai-2019].