TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Sten Marcus Nelson

Automaatse sünkroontõlke katsetamine ja optimeerimine inglise-eesti keele näitel

Bakalaureusetöö (9 EAP)

Juhendaja: Mark Fišel, PhD

Tartu 2024

Automaatse sünkroontõlke katsetamine ja optimeerimine inglise-eesti keele näitel

Lühikokkuvõte:

See bakalaureusetöö keskendub inglise-eesti sünkroontõlke mudelite katsetamisele ja hindamisele. Treeniti kuus wait-k mudelit kolmel erineval k-väärtusel ja kahel erineval andmestikul, lisaks treeniti mõlemal andmestikul üks järjend-järjendiks mudel. Töö käigus näidati, et eesti keelde on võimalik teha automaatset sünkroontõlget, kasutades selleks wait-k mudelit. Leiti, et wait-k ei ole halvem kui traditsiooniline järjend-järjendiks mudel, on piisavalt kiire reaalajas kasutamiseks, kuid aeglasem, kui treeningandmeid on vähem.

Võtmesõnad:

tehisõpe, reaaltõlge

CERCS: P176 Tehisintellekt

Testing and optimising synchronous translation from English to Estonian

Abstract: This bachelor's thesis focuses on the testing and evaluation of English-Estonian simultaneous translation models. Six wait-k models were trained on three different k-values and two different datasets, along with one sequence-to-sequence model on each dataset. The study demonstrated that it is possible to perform automatic simultaneous translation into Estonian using the wait-k model. It was found that wait-k is not inferior to the traditional sequence-to-sequence model, is fast enough for real-time use, but slower when there is less training data.

Keywords:

automatic learning, machine translation

CERCS: P176 Artificial intelligence

Sisukord

Sissejuhatus

Keeletöötluse maastikul on sünkroontõlge suur väljakutse, eriti väiksema andmehulgaga keelte puhul, nagu seda on eesti keel. Keelte omapärad mõjutavad sünkroontõlke puhul rohkem mudelite tulemust. Rolli mängib lausetes sõnade järjekord ja võib ette tulla olukordi, kus sünkroontõlke mudelid on sunnitud tõlkima, aga pole veel piisavalt infot sisse loetud, et korrektselt tõlkida. Olemasolevad sünkroontõlke mudelid baseeruvad tihti *wait-k* süsteemil, kus mudel on teatud arv sõnu sisendist maas. Siiani pole eriti põhjalikult uuritud *wait-k* mudeleid kasutades eesti keelde tõlkimist.

Töö eesmärk on katsetada sünkroontõlget eesti keelde ja leida vastused küsimustele:

1. Kui palju halvem on *wait-k* võrreldes traditsioonilise järjend-järjendiks (ingl *sequence to sequence*) mudeliga?
2. Kas *wait-k* on piisavalt kiire reaalolukorras kasutamiseks?
3. Kas *wait-k* töötab teisiti, kui treeningandemid on vähem?

Töö koosneb teoreetilisest, metoodikat kirjeldavast ja praktilisest peatükist. Esimeses peatükis kirjeldatakse erinevaid töö praktilisest osast arusaamiseks vajalikke mõisteid ja samas valdkonnas varem tehtud töid. Teine peatükk annab ülevaate, kuidas on praktiline osa realiseeritud ning milliseid tööriistu on kasutatud. Töö kolmas peatükk on praktiline, kus kahel EuroParl inglise-eesti erineva suurusega andmestikul treenitakse kolme erineva k-väärtusega mudelit. Lisaks kuuele treenitud *wait-k* mudelile treeniti ka mõlemal andmestikul üks järjend-järjendiks mudel. Seega luuakse kokku kaheksa mudelit. Seejärel hinnatakse iga mudeli treenimiskiirust ja sooritust. Kõige lõpuks proovitakse kuut *wait-k* mudelit optimeerida. Tagamaks suuremat arvutuskiirust, kasutatakse töös Tartu Ülikooli kõrgjõudlusega andmetöötluskeskkonda HPC *(High-Performance Computing).*

1. Taust

1.1. Masintõlge

Masintõlge on protsess, milles tarkvara abil muudetakse lähtetekst mõneks teiseks keeleks. Idee ei ole uus ja erinevaid projekte, kus sellist kontseptsiooni rakendatakse, leiab juba 1970-ndatest aastatest (Németh, 2019). Läbi aastate on masintõlge arenenud ja teksti tõlkimise probleemile on tekkinud mitmeid lähenemisviise.

Kõige lihtsam lähenemine tõlkimisele on võtta ette sõnaraamat ja tõlkida iga sõna ühest keelest teise. Masintõlke kontekstis tähendab see, et peab leiduma kahe keele vahel sõnastik, mida kasutatakse, et leida igale sõnale vaste. Selline lähenemine töötab hästi, kui keeled on omavahel grammatiliselt sarnased (Lone *et al.*, 2023). Kahjuks ei ole keeled omavahel alati sarnased ja alati ei pruugi kahe keele vahel sõnastikku leiduda. Probleem tekib ka siis, kui sõna ei leidu sõnastikus või sõna ei saa üks ühele tõlkida.

Reeglipõhine masintõlge oli populaarne aastatel 1970-1990 (Németh, 2019). Reeglipõhine lähenemisviis nõuab põhjalikke teadmisi lähte- ja sihtkeelest ning jaguneb veel omakorda otseseks ja kaudseks(vahekeelt kasutatavaks tõlkeks) (Su *et al.*, s.a.). Vahekeelena on kasutatud näiteks esperantot. Vahekeelsel lähenemisviisil pole vaja reegleid iga keele vahel, vaid piisab, kui defineerida reeglid iga keele ja näiteks esperanto keele vahel. Kaudne lähenemisviis toimub kolmes faasis: alguses muudetakse lähtetekst vahepealseks lähtekeele esituseks, seejärel vahepealne lähtekeele esitus vahepealseks sihtkeele esituseks ja lõpuks vahepealne sihtkeel lõplikuks sihtkeeleks (Su *et al.*, s.a.). Vahepealsed esitused sisaldavad näiteks sõnade lemmatiseeritud vormi, sõnade liiki, käändeid jmt

Statistikapõhine masintõlge oli populaarne aastatel 1990-2010 ja see jaguneb kolmeks faasiks (Koehn *et al.*, 2003).

1. Keele modelleerimine. Selles etapis proovib mudel ennustada, mis on õige sõna, arvestades konteksti (näiteks eelnevad sõnad lauses).
2. Tõlke modelleerimine. Selles etapis ennustab mudel, mis on kõige parem tõlge antud sõnale.
3. Dekodeerimine. Kui eelnevad kaks etappi on leidnud igale sõnale lauses tema kõige

tõenäolisema tõlke, siis viimases etapis ennustab mudel tõlgitud sõnade järjekorda. Närvivõrgud on alates 2014. aastast saanud kõige populaarsemaks masintõlke viisiks (Britz *et al.*, 2017). Närvivõrgu eeliseks on tema suurem paindlikkus ja ta ei ole jagatud mitmesse etappi, vaid on tehtud kõik ühe närvivõrguga (Bahdanau *et al.*, 2016). Närvivõrkude ohuks on ebavõrdsed treeningandmed, sest siis ei suuda mudel õppida haruldastest olukordadest paralleelselt sagedaste olukordadega (Bahdanau *et al.*, 2016). Kui alguses kasutasid närvivõrgud tõlkimiseks LSTM rakke, siis tänapäeval on üle mindud juba Transformeritele.

1.2. Transformerid

Transformer on närvivõrgu komponent, mis muudab sisendjärjendi väljundjärjendiks. See saavutatakse tähelepanu mehhanismile tuginedes.(Vaswani *et al.*, 2017) Transformeri kasutus on saanud masintõlkes laialt levinuks, sest on kõige parema sooritustulemusega (Sankararaman *et al.*, 2022). Transformer on närvivõrgu mudel, mis koosneb mitmest kodeerija-dekodeerija kihist. See võtab sisendiks järjendi märgendeid (ingl *tokens)* ja väljastab transformeeritud ehk tõlgitud märgendite järjestuse. Märgendid on enamasti sõnad. Märgend võib ka näiteks olla ingliskeelne väljend *let's go*, mille vasteks oleks eestikeelne märgend *lähme* või hispaaniakeelne märgend *vamonos*. Transformerid muudab efektiivseks nende enesetähelepanu (ingl *self-attention*) omadus.

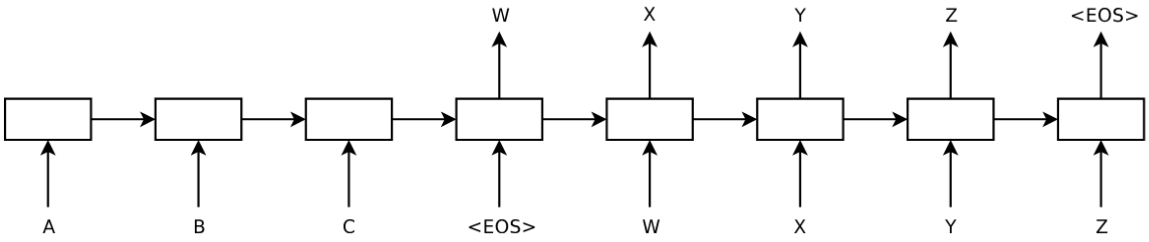
Enesetähelepanu määrab ära, kui tähtis on mingi märgend lauses. Selle saavutamiseks õpivad iga järjendi märgendid ära, millistele märgenditele peaksid nad keskenduma sama järjendi sees (Vaswani *et al.*, 2017). Masintõlke ülesannetes on järjendiks tihti laused. Seega õpib mudel esiteks seda, mis märgendid on järjendis olulised ja paneb paika nende omavahelised suhted.

Transformerite arhitektuur võimaldab mudelil paraleelselt enesetähelepanu funksiooni rakendada. Kuna enesetähelepanu funksioon on ruutkeerukusega, siis tranformerite võime paralleeselt enesetähelepanu funksioonidega töötada vähendab märgatavalt treenimisaega, võrreldes sellega, kui arvutused oleks tehtud lineaarselt (Vaswani *et al.*, 2017).

1.3. Järjend-järjendiks transformeritega

Otsese masintõlke puhul tõlgitakse teksti, leides igale sõnale vaste tõlgitavast keelest. Selline lähenemine töötab, kui keelte grammatikareeglid on sarnased. Kui keeled on erinevad ja üks-ühele tõlkimine ei anna soovitud tulemust, siis on vaja kasutada teisi lahendusi. järjend-järjendiks (ingl

sequence to sequence) on üks lahendus. Selle asemel et tõlkida iga sõna eraldi, taandatakse tõlkimine järjendite tasemele, enamasti lausetele. Kuna lause on keeles eraldiseisev üksus, siis võib mõelda, et kui tõlkida teksti iga lause õigesti, peaks ka tervikteksti tõlge olema õige. Järjendid koosnevad märgenditest. Märgendid võivad lisaks sõnadele sisaldada ka lause lõputähist, sõnapaare jmt. (Sutskever *et al.*, 2014) Iga märgenditest koosnev lause tõlgitakse tervikuna, lauseid eraldab lauselõputähistus <EOS> (ingl *end of sentence*) (vt joonis 1).



Joonis 1. Näidismudel, kus lause, mis koosneb märgenditest ABC tõlgitakse lauseks, mis kosneb märgenditest XYZ (Sutskever *et al.*, 2014)

Mudeli tööpõhimõte on järgnev:

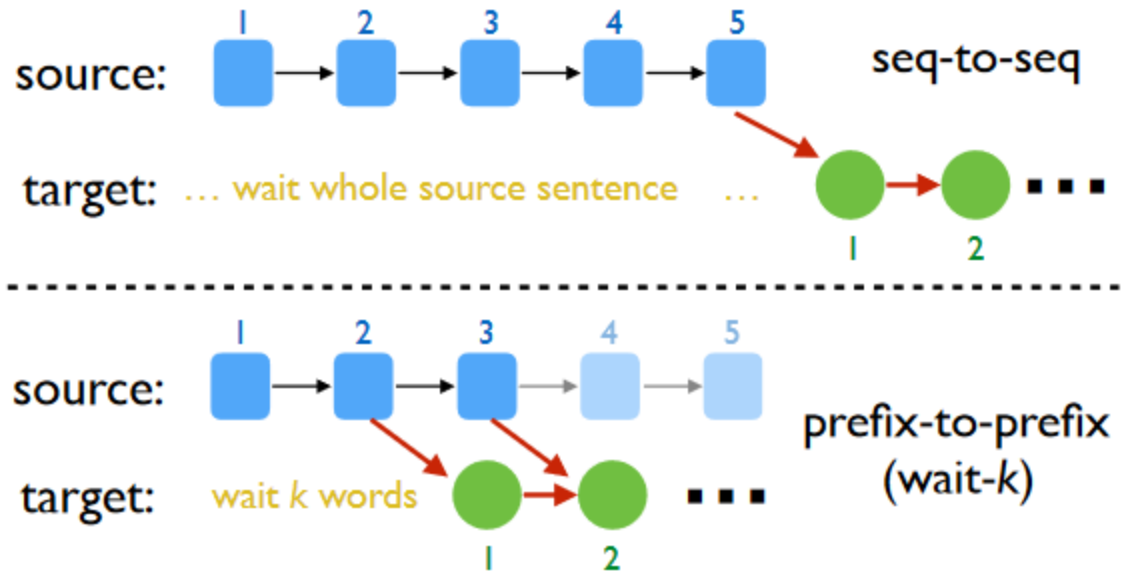
1) sisendiks võetakse märgendite järjend;
2) märgendite järjend kodeeritakse algse keele vektorite järjendiks;
3) vektorjärjend dekodeeritakse;
4) tulemuseks on tõlgitud märgendite järjend.

Probleem tõlkimise taandamisel lausete tasemele tekib, kui kontekst läheb lausete vahel kaduma. Näiteks, kui tõlgitakse laused “Mari on laisk. Ta ei läinud kooli”, siis järjend-järjendiks mudeliga tõlgitakse laused inglise keelde kui “Mari is lazy. He did not go to school.” Seega läks antud näites tõlkimisel kontekstina kaduma inimese sugu. Probleem tekib ka siis, kui kasutada järjend-järjendiks mudelit reaalajas tõlkimiseks. Kuna mudel vajab sisendiks tervet järjendit, mis enamasti on lause, siis peab mudel ootama, millal lause lõpeb, et seda tõlkida. Reaalajas võib see olla aga liiga aeglane. Selle probleemi lahenduseks oleks *wait-k* mudel. (Chang *et al.*, 2022)

1.4. *Wait-k*

Sünkroontõlge on närvivõrgupõhise masintõlke (NMT) laiendus. Selle eesmärk on teostada voogedastuse tõlget, väljastades tõlke enne, kui allika sisend on lõppenud. See on rohkem rakendatav reaalmaailma olukordades nagu rahvusvahelised konverentsid, kus inimesed saavad suhelda viivituseta.

Reaalajas tõlkimise probleemile oleks lihtne lahendus *wait-k* dekodeerija. *Wait-k* idee loeb kõigepealt sisse k-märgendit, peale mida ta ennustab vaheldumisi uue märgendi ja loeb sisse uue märgendi (M. Ma *et al.*, 2019). Selline lähenemise töötab sarnaselt inimtõlkijate tööle, kus tõlkija on alati mingi arv sõnu rääkijast maas. Selline mudel peab arvesse võtma ka olukordi, kus sisseloetud märgendi põhjal ei saa samal ajal tõlkida märgendit. Erinevalt järjend-järjendiks mudeliga algab tõlkimine pihta enne lause lõppu (vt joonis 2).



Joonis 2. Järjend-järjendiks aj *wait-2* mudeli võrdlus (M. Ma *et al.*, 2019)

Wait-k mudeli dekodeerimine algab k-märgendi lugemisega ning seejärel vaheldub ühe märgendi lugemise ja kirjutamise vahel, kuni kogu allikas on loetud või genereerimine on lõppenud. Kui tavaliselt on sisseloetud märgendite arv dekodeerimise faasis kindlaksmääratud, siis *wait-k* mudelil on funktsioon, millega üritatakse leida minimaalset märgendite arvu (Elbayad *et al.*, 2020).

1.5. Sünkroontõlke hindamine

Masintõlke mudelite hindamine on oluline samm mudelite arendamisel. Kõige täpsemalt suudab mudelit hinnata mõlemat keelt valdav inimtõlkija, aga see oleks liiga ajakulukas. Vaja on viisi, kuidas anda automaatselt hinnang mudeli täpsusele. Sünkroontõlke puhul on oluline ka lisaks tõlkekvaliteedile hinnata tõlkekiirust.

1.5.1. BLEU

Üks tuntumaid automaatseid masintõlke hindamise meetodeid on BLEU, mis tugineb n-grammi kattuvusele lähteteksti ja tõlgitud teksti vahel, arvestades sealjuures lähtetekstis sõnade esinemisarvuga (Papineni *et al.*, 2002). BLEU eelised on kiirus ja väike ressursivajadus. BLEU skoor on vahemikus 0-100, kus kõrgem number tüüpiliselt tähendab täpsemat tõlget. BLEU-l on ka palju probleeme, aga üldjuhul on see kiire ja efektiivne meetod mudeli tõlkekvaliteedi hindamiseks.

1.5.2. AL ja AL-CA

Keskmine mahajäämus (ingl *average lagging)*, edaspidi AL, on sünkroontõlke mudelite puhul hinnang, kui palju on tõlkija maas rääkijast. Masintõlke puhul on rääkija algtekst ja tõlkijaks mudel. (M. Ma *et al.*, 2019) AL-i ühik on sõnade arv, mille võrra mudel on maas lähtetekstist loetud sõnade poolest, ehk mida suurem on AL, seda rohkem sõnu on mudel maas. Ideaalis oleks *wait-k* mudelite puhul AL alati võrdne k-väärtusega. Siiski pole reaalsuses keelte omavahelise erinevuse ja arvutuste ajakulu tõttu AL alati võrdne k-väärtusega, aga on tugevalt sellega korrelatsioonis.

Arvutusteadlik keskmine mahajäämus (ingl *Computation Aware Average Lagging)*, edaspidi AL-CA mõõdab ajaliselt, kui palju on tõlkemudel maas lähteteksti sisselugemisest millisekundites (X. Ma, Pino, *et al.*, 2020). AL-CA loodi algselt automaatsete kõne-tekst mudelite hindamiseks, aga seda saab ka kasutada tekst-tekst mudelite hindamiseks. Mida suurem on AL-CA, seda rohkem millisekundeid keskmiselt peab ootama, et sõna sisselugemisest mudel selle sõna ära tõlgiks. Seda kasutatakse, et võrrelda sünkroontõlke mudelite tõlkekiirust.

1.6. Seotud tööd

Mõte sünkroontõlget teha masintõlkega ei ole uus. Aastal 2008 kirjeldasid Fügen, Waibel ja Kolss enda töös inimtõlke probleeme ja pakkusid välja lahendusi, kuidas masintõlget rakendada sünkroontõlke probleemile. Lahendus, mis välja pakuti, sisaldas nelja mudelit. Esimene oli sõna-sõna ja fraas-fraas tõlkemudel, kus treeningandmetel õppis mudel ära sõna ja fraasi kaupa, mis sõna või fraas tõlgitavas keeles neile vastab. (Fügen *et al.*, 2007) Tegemist oli statistikapõhise mudeliga, kus valiti kõige suurema tõenäosusega tõlke vaste. Teine oli 4-gramm mudel. Kolmas oli sõnade järjekorra muutmise mudel ning neljas oli lihtne sõnade ja fraaside loendusmudel. Tööpõhimõte seisnes sõnade ja fraaside tõlkimises ning ümberpaigutamises nii, et igale sõnale ja fraasile tõlgitavas keeles vastaks tulemuses mõni tõlgitud fraas ja sõna.

Aastal 2017 kirjeldasid Vaswani, Shazeer, Parman, Uszkoreit ja Jones uut tüüpi arhitektuuri: transformer, kus tähelepanu kihi lisamine parandas hüppeliselt tõlkemudelite võimekust (Vaswani *et al.*, 2017). Transformereid hakati peale seda artiklit kasutama massiliselt erinevates masinõppe mudelites k.a. sünkroontõlke ülesannetes.

Esimest korda kirjeldati *wait-k* mudelit aastal 2019 artiklis "STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework" (Ma *et al.*, 2019). Aastal 2020 tutvustasid Elbayad, Besacier ja Verbeek efektiivsemat *wait-k* mudelit transformerite baasil ja näitasid, kuidas *wait-k* mudelid erinevates keskkondades käitusid (Elbayad *et al.*, 2020).

Aastal 2022 kirjeldasid Chang, Chuang ja Lee oma töös "Anticipation-Free Training for Simultaneous Machine Translation" sünkroontõlke mudelite probleeme ja pakkusid välja enda lahenduse hallutsinatsiooni probleemile, kus mudel ennustab sõnu, mis lähtetekstist puudusid (Chang *et al.*, 2022). Chang, Chuang ja Lee näitasid, kuidas optimeerida *wait-k* mudelit inglise-saksa ja ka inglise-hiina keelte näitel. Selleks jagati tõlkimise kaheks sammuks: tõlkimiseks ja ümberpaigutamiseks. Idee oli selles, et sõnade järjekord pannakse paika peale tõlkimist, et mudel ei tõlgiks sõnu, mida algtekstis pole. Seega välditakse probleemi, mis on omane *wait-k* mudelitele (Chang *et al.*, 2022).

2. Metoodika

Andmete ettevalmistamine, mudelite treenimine ja hindamine toimub Tartu Ülikooli kõrgjõudlus-keskkonnas HPC (*high performance computing)*. Mudelite treenimine toimub ühel GPU-l (*graphics processing unit)*. GPU võimaldab paralleelselt teha mitmeid arvutusi, võimaldades mudeleid kiiremini treenida.

Andmete märgendamiseks kasutatakse SentencePiece märgendajat. SentencePiece on avatud lähtekoodiga ja võimaldab treenida mudeleid sõnade märgendamiseks erinevate keelte vahel (Kudo & Richardson, 2018). Andmete puhastamiseks kasutatakse Moses märgendajat (*Moses - Moses/Overview*, s.a.). Mosese märgendajaga tuleb kaasa kaust *scripts*. Kasutades seal asuvaid programmijuppe, saab enda teksti eeltöödelda, näiteks eemaldades kõik ebavajalikud märgid tekstist või väiketähestades teksti.

Töö on tehtud kasutades vabavarana saadaval tööriistakomplekti Fairseq (Ott *et al.*, 2019). Fairseq põhineb PyTorchil, mis on Püütoni programmeerimiskeelele loodud raamistik erinevate masinõppega seotud ülesannete jaoks (Paszke *et al.*, 2019). Fairseq võimaldab andmeid eeltöödelda ja treenida (Ott *et al.*, 2019).

Treenimise protsessi jälgimiseks kasutati projektis Wandb-d. Wandb salvestab treenimise käigus erinevaid andmeid ja visualiseerib neid reaalajas. Wandb-d kasutades on kerge olla kursis, kuidas mudeli treenimine edeneb ja veenduda, et mudeli treenimine on sujuv.

Mudelite tulemuste hindamiseks on kasutatud SimulEvali. Sünkroontõlke mudelite hindamine on keerulisem kui järjend-järjendiks mudelite hindamine, sest lisaks tõlkekvaliteedile tuleb hinnata ka mudelite ooteaega. (X. Ma, Dousti, *et al.*, 2020) Sünkroontõlke mudel hakkab tõlkima enne, kui ta loeb läbi kogu lähteteksti ja selliste mudelite hindamiseks on kasutusele võetud serveri-kliendi süsteem. Selles süsteemis simuleeritakse olukorda, kus server saadab lähteteksti reaalajas mudelile, mis reaalajas saadab tõlget vastu. SimulEval hindab terve aja vältel mudeli sooritust ja ka ooteaega. SimulEval kasutamiseks tuleb implementeerida agent süsteemiloogika haldamiseks. Agent määrab ära, mis tegevus peaks mudeliga toimuma kindlal ajahetkel, ehk kas mudel loeb sisse uue märgendi või ennustab uut märgendit. (*Agent — SimulEval 1.1.0 documentation*, s.a.)

Töös andmete allalaadimiseks, ettevalmistamiseks, mudelite treenimiseks ja hindamiseks kasutatakse inglise-eesti keele jaoks modifitseeritud *scripte*, mida originaalis kasutati töö "Anticipation-Free Training for Simultaneous Machine Translation" implementatsiooniks (Chang *et al.*, 2022). Kui artiklis treeniti *wait-k* mudelid võrdluseks autorite poolt välja pakutud mudelile,

siis käesolevas töös kasutati ainult baas *wait-k* mudelite treenimiseks ja hindamiseks loodud *scripte* ja SimuEvali agenti.

Artikli implementatsioonis baas *wait-k* mudelid optimeeriti, kasutades distileerimise meetodit, ehk õpetaja-õpilane süsteemi, kus õpetajaks oli järjend-järjendiks mudel. Sama optimeerimist katsetati ka käesolevas töös, et optimeerida kuute *wait-k* mudelit.

3. Sünkroontõlke katsetamine inglise-eesti keele näitel

3.1. Andmete ettevalmistus

Andmestik on EuroParl inglise-eesti paralleelkorpus, mis pärineb Euroopa Parlamendi veebilehelt (Tiedemann, s.a.). Töö kasutab EuroParl V8 inglise-eesti Moses-formaati. Korpuse esimesed 2000 lauset võeti testandmeteks, järgmised 1000 võeti valideerimise andmeteks. Ülejäänud 648 236 lauset võeti treenimiseks suure ressursiga mudelite jaoks ja 300 000 lauset võeti treenimisandmeteks vähese ressursiga mudelite jaoks.

Järgmisena kasutatakse Mosese märgendajat, et normaliseerida kirjavahemärgid, eemaldada mitteprinditavad tähemärgid ja väiketähestada kogu tekst. See on vajalik, et mudelil oleks võimalikult lihtne tõlkimist õppida. Pärast andmete puhastamist treenitakse SentencePiece märgendaja mudel treeningandmetel ja märgendatakse treening-, valideerimis- ja testandmed. Mudeli sõnavara suuruseks määrati 32 000. Kõige lõpuks kasutatakse Fairseqi funktsiooni *pre_process*, mis muudab andmestiku mudeli treenimise jaoks sobivale kujule.

3.2. Mudelite treenimine

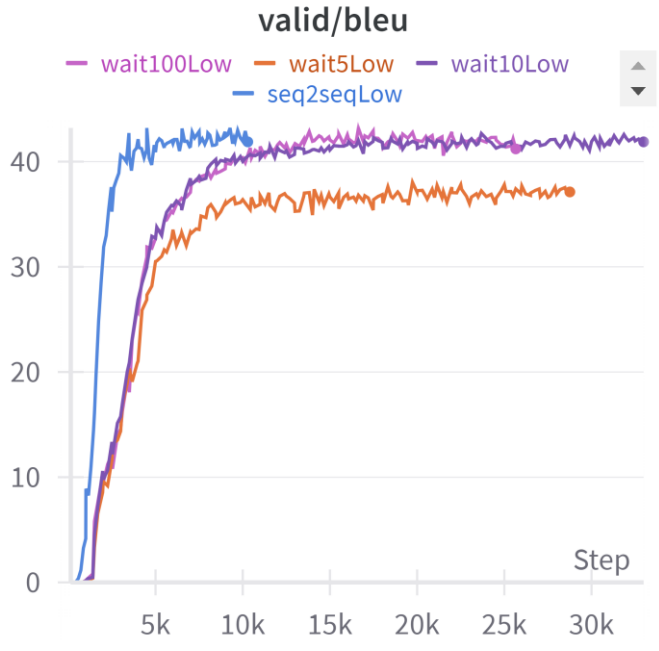
Kokku treeniti kolme erineva k-väärtusega mudelit kahe erineva suurusega andmestikul, seega on kokku loodud kuus mudelit. On võimalik, et erinevad k-väärtused eelistavad erinevaid parameetreid treenimisel. Selles töös erinesid parameetrid ainult oma k-väärtuse poolest, et neid saaks omavahel võrrelda.

Mudelite treenimisel kasutati mudeli jaoks, lisaks lähteandmete asukoha, salvestusasukohale ja keeltele, järgmisi parameetreid:

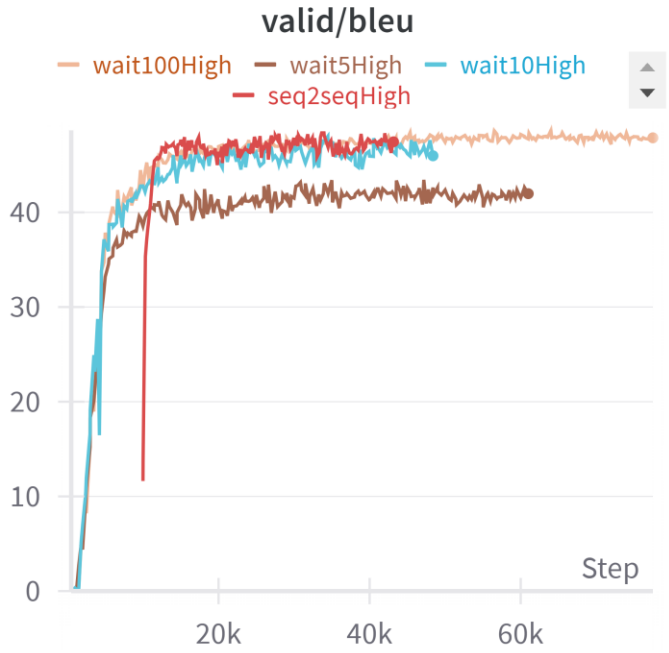
1) **max-tokens** 8000 – maksimaalne märgendite arv ühes treening *batchis*;
2) **update-freq** 4 ja **fp16** – treenimise kiirust parandavad parameetrid;
3) **arch** waitk_transformer – mudeli arhitektuur, milleks on *waitk_transformer*;
4) **criterion** label_smoothed_cross_entropy ja **label-smoothing** 0.1 – kasutatakse mudeli treenimisel selleks, et mudel liiga enesekindel poleks enda ennustustel, lubades 0.1 kõikumisruumi ennustuse tõenäosuses, sellega vältides mudeli ülesobitumist;
5) **clip-norm** 10.0 aitab mudeli treenimise protsessi stabiliseerida, et mudel ei teeks liiga suuri järeldusi, vaid õpiks stabiilselt;

6) **weight-decay** 0.0001 – mudeli treenimise käigus karistab mudelit liiga suurte kaalude puhul, see sunnib mudelil õppimise käigus leidma lahendusi, kus kaalud ei kasvaks liiga suureks;
7) **optimizer** adam **lr** 1e-4 ja **lr-scheduler** inverse_sqrt – adam on populaarne optimiseerimise algoritm mudeli treenimisel ja inverse_sqrt aitab mudelil treenimise lõpu poole olla stabiilsem, vähendades õppimise kiirust õppimise sammude suurenedes;
8) **warmup-updates** 4000 – kui pikk on mudelil soojendusfaas. Soojendusfaasis mudel tõstab oma õppimiskiirust treenimise algul, optimiseerides mudeli treenimse algusfaasi;
9) **max-update** 300000 – piirab mudeli treenimisaega;
10) **patience** 50 – kui mudel ei saa paremaks, siis treenimine lõpetatakse.

Veel on parameetreid parima mudeli salvestuse kohta. Lõpuks salvestatakse valideerimisandmestikul parima BLEU skoori saavutanud mudel. Mudeli treenimise protsessi saab jälgida *wandb* projektist. Vähese ressursiga mudelite puhul saavutasid mudelid treenimisel kesmiselt ~39 BLEU skooriga tulemuse (vt joonis 3). Suure ressursiga mudelid saavutasid märkimisväärselt parema tulemuse treenimisel, kus iga mudel oli keskmiselt 6 BLEU skoori võrra parem, kui sama k-väärtusega mudel vähese ressursiga mudelite puhul (vt joonis 4).



Joonis 3. Vähese ressursiga mudelite BLEU skoor treenimisel valideerimisandmestikul



Joonis 4. Suure ressursiga mudelite BLEU skoor treenimisel valideerimisandmestikul

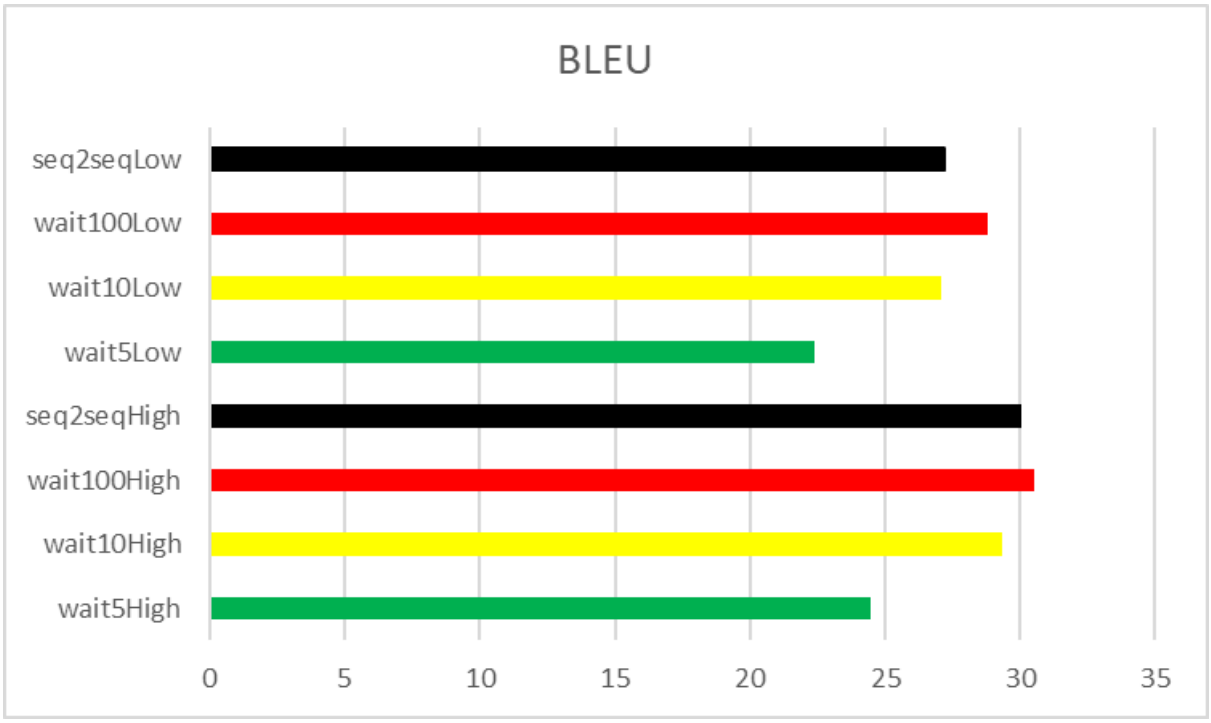
Graafikud näitavad mudelite BLEU skoori valideerimise andmestikul treenimise käigus. Lisaks *wait-k* mudelitele treeniti kontrolliks samadel andmetel ka üks järjend-järjendiks mudel kõige tavalisema transformeri arhitektuuriga. Graafikud on enamasti sujuvad, alguses õppides kiiremini ja hiljem aeglasemalt. Treeningu lõpuks salvestati iga mudeli parima BLEU skoori saavutanud kaalude konfiguratsioon. Nii vähese kui ka suure andmemahu puhul treenis järjend-järjendiks mudel kiiremini kui *wait-k* mudel. *Wait-100* mudel, mis simuleeris järjend-järjendiks mudelit, kasutades *wait-k* arhitektuuri, saavutas tavalise järjend-järjendiks mudeliga võrdväärse taseme treeningfaasis. Treenimiskiirusest saab järjeldada, et traditsioonilised järjend-järjendiks mudelid treenivad tunduvalt kiiremini kui *wait-k* mudelid.

3.3. Mudelite hindamine

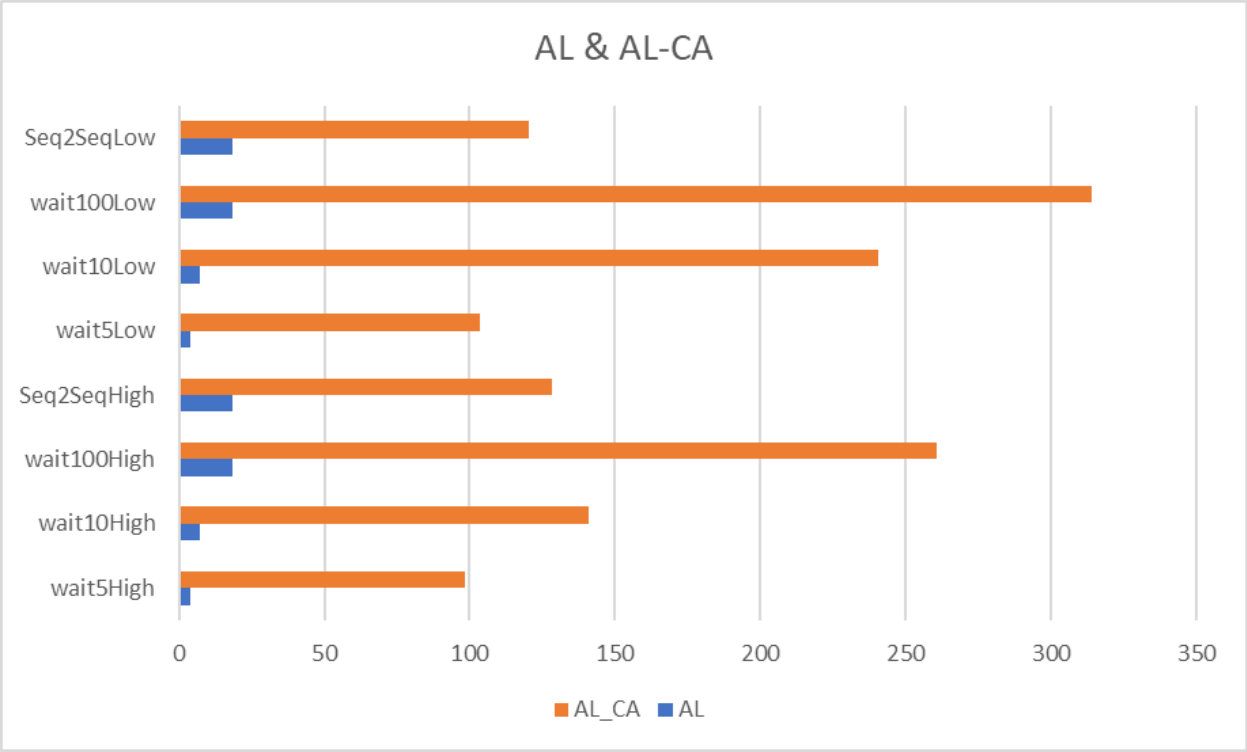
Mudelite hindamine toimus SimulEvalis, kus järjend-järjendiks mudeli puhul anti parameetriks *test-waitk* <k-väärtus> asemel *fullsentence*. Ehk SimulEval andis ka järjend-järjendiks mudelile ette teksti tervete lausete kaupa. Agendiks kasutati Changi loodud agenti enda sünkroontõlke mudelite hindamiseks (Chang *et al.*, 2022). SimulEvali ülejäänud parameetrid olid seotud varasemalt treenitud SentencePiece märgendajaga, serveri *port*'i määramisega ja sacreBLEU

märgendajaga hindamiseks. SacreBLEU märgendajaks kasutati kõige tavalisemat 13a märgendajat.

Iga *wait-k* mudelit hinnati kokku kolm korda ja AL-CA väärtuseks võeti kolme korra keskmine. Ülejäänud SimulEvali poolt hinnatavad väärtused olid konstantsed kõigi kolme katse tulemusel. Test andmetel saavutatav BLEU skoor oli madalam, kui treeningandmetel (vt joonis 5). Test andmestikul parimad tulemused olid *wait-100* mudelil, aga neil oli ka kõige pikem ooteaeg (vt joonis 6). Tulemustest saab järeldada, et *wait-k* mudelid ei ole kehvema tulemusega, kui järjend-järjendiks mudelid, ning suure k-väärtuse puhul on *wait-k* parem, kui järjend-järjendiks mudelite puhul.



Joonis 5. Mudelite BLEU skoor testandmestikul



Joonis 6. Mudelite AL ja AL-CA skoor

Üldine trend on, et mida suurem on k-väärtus, seda parem on mudeli tulemus, aga ühtlasi on ka suurem ajakulu. Tulemustest on ka näha, kuidas suurema ressursiga treenitud mudelite sooritus oli parem kui vähese ressursiga mudelite puhul. Järjend-järjendiks mudelite puhul on kiirem ajaline kulu seletatav sellega, et teksti serveeriti mudelile ette lausete, mitte sõnade kaupa (vt joonis 6). Tulemustest saab järeldada, et *wait-k* mudelite ennustuskiirus on piisav reaalolukorra jaoks kasutamiseks. Veel saab tulemustest järeldada, et väikese ressursiga treenitud mudelite puhul on keskmine ooteaeg pikem.

Mudelite sooritus on hea. Kõik mudelid jäid alla 40 BLEU väärtuse, kuid kui vaadata, kuidas mudelid tõlgivad, siis ei leitud näidet, kus mudel tõlkis lauset valesti.

Tabel 1. Ingliskeelse lause hea tõlke näide punasega märgitud mudeli hinnangul valed sõnad

Ingliskeelne sisend	**debates on cases of breaches of human rights, democracy and the rule of law**
seq2seqHigh	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>

seq2seqLow	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>
wait5High	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine (esitatud resolutsiooni ettepanekud) </s>
wait5Low	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine (esitatud resolutsiooni ettepanekute) (arutelu) </s>
wait10High	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>
wait10Low	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>
wait100High	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>
wait100Low	inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine </s>
Inimtõlge	**inimõiguste, demokraatia ja õigusriigi põhimõtete rikkumise juhtumite arutamine (arutelu)**

Lausega „debates on cases of breaches of human rights, democracy and the rule of law“ said kõik mudelid hästi hakkama (vt tabel 1). Kõige rohkem erinesid inimtõlkest mõlemad *wait-5* mudelid ja said selletõttu kehvema BLEU skoori. Samas, isegi lause puhul, kus mudel enda arvates tõlkis kehvasti (~17BLEU), võib tõlkega rahule jääda, sest lause tõlge läheb kokku algse lausega ja lause mõte ei lähe tõlkes kaotsi (vt tabel 2).

Tabel 2. Inglisekeelse lause kehva tõlke näide (mudeli arvates) punasega märgitud mudeli hinnangul valed sõnad

Ingliskeelne sisend	**strengthening european legislation in the field of information and consultation of workers (tabling of motions for a resolution): see minutes**
seq2seqHigh	euroopa õigusaktide tugevdamine töötajate teavitamisel ja konsulteerimisel (resolutsiooni ettepanekute esitada) (vt protokoll) </s>
seq2seqLow	euroopa õigusaktide tugevdamine teabe ja konsulteerimise valdkonnas (esitamine) (vt protokoll) </s>
wait5High	euroopa õigusaktide tugevdamine teabe ja konsultatsioonide valdkonnas (esitamine) (arutelu) </s>
wait5Low	euroopa õigusaktide tugevdamine teabe- ja konsultatsioonide valdkonnas (esitamine), milles käsitletakse resolutsiooni ettepanekut (vt protokoll) </s>
wait10High	euroopa õigusaktide tugevdamine töötajate teavitamise ja nõustamise valdkonnas (esitatud resolutsiooni ettepanekud) (vt protokoll) </s>
wait10Low	euroopa õigusaktide tugevdamine töötajate teabe- ja konsulteerimise valdkonnas (esitamine resolutsiooni ettepanekute esitamiseks) (vt protokoll) </s>
wait100High	euroopa õigusaktide tugevdamine töötajate teavitamise ja konsulteerimise valdkonnas (muudatusettepanekute esitamine resolutsiooni kohta) (vt protokoll) </s>
wait100Low	euroopa õigusaktide tugevdamine töötajate teavitamise ja konsulteerimise valdkonnas (esitamine resolutsiooni ettepanekud) (vt protokoll) </s>
Tegelik	**euroopa õigusaktide täiustamine seoses töötajate teavitamisega ja nendega konsulteerimisega (esitatud resolutsiooni ettepanekud) (vt protokoll)**

Lause „strengthening european legislation in the field of information and consultation of workers (tabling of motions for a resolution): see minutes“ puhul on lihtne näha, miks SimulEval hindas mudelite sooritust kehvemini. Tihti on tõlkimisel mitu õiget viisi ja treenitud mudelid tõlkisid kõik üsna sarnaselt: näiteks *strengthening* tõlgiti kui *tugevdamine*, samas kui inimtõlkija oli seda tõlkinud kui *täiustamine*. Lisaks on näha tõlkemudelite puhul grammatikavigu – *esitamine ja esitada kasutati esitatud asemel*. Tähelepanuväärne on ka lauselõpumärgend peale igat lauset. On võimalik, et hindamisel arvestati ka märgendiga <s/> ja sellepärast tuli keskmine BLEU skoor madalam, kui see reaalsuses oli.

3.4. Optimeeritud mudelid

Mudelite optimeerimise põhimõtteks on tugevama mudeli õpetajaks võtmine. Tugevam mudel aitab õppida nõrgemal mudelil, optimeerides nii nõrgemat mudelit. Töös on õpetaja mudeliks samadel andemetel treeenitud järjend-järjendiks mudel. Mudeli optimeerimine sellist meetodit kasutades parandas *wait-k* sooritust vähesel määral (vt tabel 3).

Tabel 3. Võrdlus opimeerimata ja optimeeritud *wait-k* mudelite vahel (BLEU).

Ressurss	k-väärtus	Baas wait-k	Optimeeritud wait-k
High	5	24.476	24.688
High	10	29.32	29.391
High	100	30.515	31.733
Low	5	22.363	23.027
Low	10	27.085	27.251
Low	100	28.79	28.168

Kokkuvõte

Bakalaureusetöö eesmärk oli katsetada ja hinnata inglise-eesti sünkroontõlke mudeleid. Töös treeniti kuut *wait-k* mudelit kolmel erineval k-väärtusel ja kahel erineval andmestikul. Veel treeniti kummalgi andmestikul üks järjend-järjendiks mudel. Kõige lõpus katsetati optimeerimist, võttes õpetaja mudeliks järjend-järjendiks mudeli.

Töö käigus demonstreeriti, et eesti keelde on võimalik automaatset sünkroontõlget teha, kasutades selleks *wait-k* mudelit. Töös näidati, et *wait–k* ei ole halvem võrreldes traditsioonilise järjend-järjendiks mudeliga. Tulemusena leiti, et *wait-k* on piisavalt kiire reaalolukorras kasutamiseks ja et *wait-k* on aeglasem, kui treeningandmeid on vähem.

Töö võimalik edasiarendus oleks treenida suuremal andmestikul *wait-k* mudel, mis töötaks kõne-tekst (ingl *speech-text*) meetodil. Veel saaks katsetada töös mainitud optimeerimisi *wait-k* probleemide kõrvaldamiseks.

Viidatud kirjandus

Agent—SimulEval 1.1.0 documentation. (s.a.). Salvestatud 28. aprill 2024, https://simuleval.readthedocs.io/en/v1.1.0/user_guide/agent.html

Bahdanau, D., Cho, K., & Bengio, Y. (2016). *Neural Machine Translation by Jointly Learning to Align and Translate* (arXiv:1409.0473). arXiv. https://doi.org/10.48550/arXiv.1409.0473

Britz, D., Goldie, A., Luong, M.-T., & Le, Q. (2017). *Massive Exploration of Neural Machine Translation Architectures* (arXiv:1703.03906). arXiv. http://arxiv.org/abs/1703.03906

Chang, C.-C., Chuang, S.-P., & Lee, H. (2022). *Anticipation-Free Training for Simultaneous Machine Translation* (arXiv:2201.12868). arXiv. https://doi.org/10.48550/arXiv.2201.12868

Elbayad, M., Besacier, L., & Verbeek, J. (2020). *Efficient Wait-k Models for Simultaneous Machine Translation* (arXiv:2005.08595). arXiv. http://arxiv.org/abs/2005.08595

Fügen, C., Waibel, A., & Kolss, M. (2007). Simultaneous translation of lectures and speeches. *Machine Translation*, *21*(4), 209–252. https://doi.org/10.1007/s10590-008-9047-0

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical Phrase-Based Translation. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 127–133. https://aclanthology.org/N03-1017

Kudo, T., & Richardson, J. (2018). *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing* (arXiv:1808.06226). arXiv. https://doi.org/10.48550/arXiv.1808.06226

Lone, N. A., Giri, K. J., & Bashir, R. (2023). Machine translation status of Indian scheduled languages: A survey. *Multimedia Tools and Applications*, *82*(29), 45145–45173. https://doi.org/10.1007/s11042-023-15287-z

Ma, M., Huang, L., Xiong, H., Zheng, R., Liu, K., Zheng, B., Zhang, C., He, Z., Liu, H., Li, X., Wu, H., & Wang, H. (2019). STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework. A. Korhonen, D. Traum, & L. Màrquez (Toim), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (lk 3025–3036). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1289

Ma, X., Dousti, M. J., Wang, C., Gu, J., & Pino, J. (2020). *SimulEval: An Evaluation Toolkit for Simultaneous Translation* (arXiv:2007.16193). arXiv.

https://doi.org/10.48550/arXiv.2007.16193

Ma, X., Pino, J., & Koehn, P. (2020). SimulMT to SimulST: Adapting Simultaneous Text Translation to End-to-End Simultaneous Speech Translation. K.-F. Wong, K. Knight, & H. Wu (Toim), *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing* (lk 582–587). Association for Computational Linguistics. https://aclanthology.org/2020.aacl-main.58

Moses—Moses/Overview. (s.a.). Salvestatud 2. mai 2024, https://www2.statmt.org/moses/?n=Moses.Overview

Németh, G. D. (2019, oktoober 30). *Machine Translation: A Short Overview*. Medium. https://towardsdatascience.com/machine-translation-a-short-overview-91343ff39c9f

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. W. Ammar, A. Louis, & N. Mostafazadeh (Toim), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (lk 48–53). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-4009

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. P. Isabelle, E. Charniak, & D. Lin (Toim), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (lk 311–318). Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

Sankararaman, K. A., Wang, S., & Fang, H. (2022). *BayesFormer: Transformer with Uncertainty Estimation* (arXiv:2206.00826). arXiv. http://arxiv.org/abs/2206.00826

Su, S. Y. W., Fang, S. C., & Lam, H. (s.a.). *AN OBJECT-ORIENTED RULE-BASED*

APPROACH TO DATA MODEL AND SCHEMA TRANSLATION.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems, 27*. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html

Tiedemann, J. (s.a.). *Parallel Data, Tools and Interfaces in OPUS*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems, 30*. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Sten Marcus Nelson,

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

Automaatse sünkroontõlke katsetamine ja optimeerimine inglise-eesti keele näitel mille juhendaja on Mark Fišel, 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 4.0, 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.

Sten Marcus Nelson

15.05.2024