TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Agnes Luhtaru

Grammatiliste vigade parandamine mitmekeelse neuromasintõlkega

Bakalaureusetöö (9 EAP)

Juhendaja: Mark Fišel

Tartu 2020

Grammatical Error Correction via Multilinual Neural Machine Translation

Abstract:
We introduce an approach to grammatical error correction that does not require annotated training data. We train a multilingual neural machine translation model that uses only language-parallel translations. There are more openly available translations available than grammatical error correction corpora, especially for low-resource languages like Estonian. We find out that this system has high recall but low precision. So it corrects plenty of mistakes but adds many mistakes to correct text. Adding artificial mistakes increases the recall and has really positive impact on spelling error correction. Our model reliably corrects grammatical errors, like subject-verb agreement and noun number, but struggles with lexical errors and unnecessary paraphrasing.

Keywords:
natural language processing, neural machine translation, grammatical error correction

CERCS: P176 Artificial intelligence

Grammatiliste vigade parandamine mitmekeelse neuromasintõlkega

Lühikokkuvõte:
Selles töös tutvustame grammatiliste vigade parandamiseks sügavõppe meetodit, mis ei vaja treenimiseks veaparandusnäiteid. Treenime mitmekeelse neuromasintõlke mudeli, mis õppimiseks kasutab vaid tõlkeid. Vabalt kättesaadavaid tõlkenäiteid on erinevate, ka väiksemate keelte kohta rohkem kui veaparandusnäiteid. Seetõttu on meetod kasulik ressursivaesematele keeltele, mille hulka kuulub ka eesti keel. Töös leiame, et süsteemil on kõrge saagis, kuid madal täpsus. Seega parandab mudel palju vigu, kuid muudab ka korrektset teksti. Veatüüpidest on süsteem edukam ühilduvusvigadega, kuid jääb hätta sõnavaliku korrastamisega. Lisaks näitame, et sünteetiliste vigade lisamine suurendab parandatavate vigade hulka ja tõuseb eeskätt õigekirjavigade paranduste hulk.

Võtmesõnad:
loomuliku keele töötlus, neuromasintõlge, grammatiliste vigade parandus

CERCS: P176 Tehisintellekt

Sisukord

1 Sissejuhatus

1.1 Teema tutvustus

Grammatiliste vigade automaatne parandamine on oluline ülesanne, mille edukas lahendamine teeb teksti toimetamise kiiremaks ja aitab keeleõppijatel võõrkeelset sisu luua ja mõista.

Selles töös on vigade all mõeldud nii leksika, grammatika kui ka ortograafia vigu. Veaparadusmeetodi eesmärk on anda väljund, kus vigased sõnad või fraasid on asendatud keeleliselt õigete ja tähenduselt võimalikult lähedastega, kuid korrektne osa tekstist on muutmata. Oluline on lause struktuuri, stiili ja tähenduse säilitamine.

Kuigi aja jooksul on ülesandele lähenetud erinevate meetoditega, nt reeglistikega [1] ja statistilise masintõlkega [2], põhinevad enamik konkureerivaid veaparanduse lahendusi tehisnärvivõrkudel ja sügavõppe kasutamine on saanud peamiseks lähenemisviisiks. 2019. aastal viidi läbi rahvusvaheline lahtine võistlus BEA-2019 [3], kus kaks kolmandikku osavõtjatest kasutasid enesetähelepanul põhinevat neuromasintõlget [4] ja ülejäänud lähenesid ülesandele konvolutsiooniliste närvivõrkudega või kombineerisid kahte eelmainitut.

Nagu paljudes sügavõpet kasutavates meetodites, on ka veaparanduses treenimiseks vaja küllaltki suurt hulka treenimisandmeid, mistõttu nõuavad parimaid tulemusi andvad meetodid palju ressursse ehk sõltuvad palju kvaliteetsest märgendatud veatõlkekorpusest, kus oleksid olemas vigased laused koos parandustega. Veaparanduse süsteemides on andmete vähesuse katsumusele lähenetud mitmetest külgedest, näiteks erinevatel viisidel sünteetiliste vigade genereerimisega [5, 6] ja juhendamata meetoditega [7].

1.2 Töö eesmärk

Selles töös tutvustame alternatiivset meetodit: vigade parandamist, mis põhineb mitmekeelse masintõlke ühekeelsel rakendamisel, on keelest sõltumatu ja ei nõua ülesandele spetsiifiliste andmete kogumist.

Meetod põhineb mitmekeelse masintõlke võimel õppida tõlkima süsteemis olemas olevates suundades, mille näiteid pole õpetamiseks kasutatud. Nii saavutab mudel tasuta (*zero-shot*) tõlke. Muidu masintõlkes olulist omadust saab ära kasutada ka veaparanduses. Nimelt on võimalik saavutada ühekeelset tõlget ehk n-ö tõlkida lähtekeelde ja sellisel tõlkel on võime parandada vigu.

Süsteemi loomiseks ei ole vaja veaparandusnäiteid. Mudeli treenimiseks kasutame tõlkenäiteid. Masintõlke mudeli õpetamiseks sobivaid näiteid on enamasti kordades rohkem kui veaparanduse näiteid, sest taaskasutada saab tekste, mida igapäevaselt tõlgitakse, näiteks subtiitrid ja juhendid. Sellest tulenevalt on kirjeldatud meetod rakendatav ka keeltele, mille jaoks ei ole kogutud juhendatud õppe meetoditeks piisavalt suuri õppijakeelekorpusi, näiteks eesti keelele.

Selles töös otsime vastuseid järgmistele küsimustele.

- Kui hea on mudeli veaparanduskvaliteet ja kas see tõuseb sünteetiliste vigade lisamisel?
- Milliste vigade parandamisega saab kirjeldatud meetod hästi hakkama ja millistega jääb hätta?
- Kas veaparanduse kvaliteet tõuseb mudeli treenimisel tõlkimise võime paranemisega sarnaselt?

Töö taustaks on käesoleva bakalaureusetöö autori kaasautorluses valminud artikkel [8], mille keskmes on üks mitmekeelne masintõlkemudel, mis suudab tõlkida, vigu parandada ja stiili üle kanda. Selle töö raames treenitava baasmudeli peamised erinevused artiklis mainitutega on stiilimärgendi kaotamine ja ühe keele vahetamine. Lisaks neile on ka andmestik mõnevõrra erinev.

Järgmises peatükis vaatame lühidalt nii edukaimaid grammatiliste vigade parandamise meetodeid kui ka ühte konkureerivat lahendust, mis ei kasuta märgendatud andmeid. Seejärel räägime 3. peatükis täpsemalt neuromasintõlkest, ühekeelsest tõlkest, vigade lisamisest ja veaparanduse hindamisest. 4. peatükis kirjeldame katseid ja 5. analüüsime nende tulemusi.

2 Taustainfo

2.1 Grammatiliste vigade parandus tehisnärvivõrkudega

Aastal 2018 esitlesid Edinburghi Ülikooli ja Microsofti teadlased lähenemist [9], mis vaatles grammatiliste vigade parandamist kui tõlkimist vigasest keelest korrektsesse. Selleks võtsid nad masintõlkes edukaima süsteemi arhitektuuri [4] ja õpetasid mudelit, kasutades erinevatest keeltest paralleellausete asemel õppijakeelekorpustest veaparandusnäiteid. Sellega pakkusid nad [9] välja ühe esimestest tehisnärvivõrkudel põhinevatest meetoditest, mis ületas tulemustelt eelnevad meetodid.

BEA-2019 võistlusel [3] kolmest kategooriast kahes võitnud süsteemide autorid [6] kasutasid sama ideed [9] veidi muudetud kujul. Andmete vähesuse probleemi lahendamiseks on neil [6] kasutusele võetud mitmed meetodid, nendest olulisem eeltreenimine sünteetiliste vigadega korpuse peal.

2.2 Annoteerimata andmetega veaparandus

Üks viis, kuidas veaparanduse korpuseta parandusi teha, on kasutada keelemudelit [7]. Autorite lähenemine seisneb idees, et madala tõenäosusega jadad sisaldavad kindlamini vigu kui suurema tõenäosusega jadad. Selles töös [7] arvutatakse esmalt sisendlause tõenäosus, seejärel vahetatakse sõnu segadushulgas olevatega ja hinnatakse seejärel uuesti. Nii proovitakse parandada kolme tüüpi vigu: õigekirjavead, morfoloogilised vead ning artiklid ja eessõnad. Nende kategooriate ühiseks jooneks on kindla segadushulga olemasolu, nt on inglise keeles piiratud arv artikleid ja eessõnu ning omadussõnade vorme.

Sellisel meetodil [7] vigaste kohtade otsimine nendes kategooriates on kõrge täpsusega, kuid madala saagisega. Samuti nõuab see kas keeleteadmisi või korpust, kus eraldada informatsiooni segadushulga jaoks.

3 Metoodika

Selle töö fookuses on *zero-shot* ühekeelse masintõlke veaparandusvõime. Katsetame, kui hästi see vigu parandab ja kuidas mõjutab sünteetiliste vigade lisamine ülesande edukusele. Selles peatükis kirjeldame lähemalt, kuidas toimib töös kasutatav neuromasintõlge ja kuidas saavutame sissejuhatuses kirjeldatud ühekeelse tõlke. Lisaks selgitame sünteetiliste vigade lisamist ja veaparanduse hindamist.

3.1 Neuromasintõlge

Neuromasintõlkes koostatakse väljundlauset sõna kaupa. Nimelt ennustatakse sisendlause ja juba valitud sõnade põhjal tõenäosusjaotus ehk iga sõnastiku elemendi kohta, kui suure tõenäosusega on see järgmiseks elemendiks. Seejärel valitakse üks või mitu parimat hüpoteesi ja jätkatakse järgmise sõna ennustamist, kuni süsteem on ennustanud lauset lõpetava märgendi.

Sageli kasutatav lähenemisviis neuromasintõlke süsteemidele on enkooder-dekooder arhitektuur. Meetodeid kodeerimiseks ja dekodeerimiseks on mitmeid, neist põhilisemad on rekurrentsete närvivõrkudega [10], konvolutsiooniliste närvivõrkudega [11] ja transformer arhitektuuriga [4]. Selles töös on kasutatud transformeril põhinevat enkooder-dekooder süsteemi.

Transformer [4] kasutab tähelepanumehhanismi. Esimestes rekurrentsete võrkudega lahendustes kodeeriti terve lause üheks fikseeritud pikkusega vektoriks, mida seejärel kasutati dekodeerimisel[12]. Selline lahendus sai aga hakkama vaid lühikeste lausetega, kuid ei suutnud mahutada pikema lause informatsiooni ühte vektorisse. Sellele probleemile prooviti leida erinevaid lahendusi, millest edukaks sai tähelepanumehhanismi kasutamine [10]. Esialgne tähelepanumehhanism [10] kujutab endast kahe suunaga rekurrentset võrku. Mõlema suuna peidetud neuronite teave ühendatakse ja iga dekodeerimise sammu juures genereeritakse kordajad, mis määravad, millistele sõnadele tuleb rohkem tähelepanu pöörata ja millistele vähem.

Transformeris [4] on rekurrentsed võrgud asendatud enesetähelepanuga. Igas kodeerija kihis vaadatakse sõna esituse leidmiseks nii sõna ennast kui ka teisi sõnu lauses. Teiste sõnade tähtsus arvutatakse iga kord ja iga sõna puhul uuesti. Seda operatsiooni tehakse paralleelselt mitmes harus, mis tänu juhuslikult seatud kaaludele suudavad õppida erinevaid nüansse. Dekodeerija loogika on sarnane, kuid väljund genereeritakse sõna korraga ja sisendiks saadakse nii kodeerijalt saadud esitused kui juba loodud väljundi algus.

Keeles esineb alati tundmatuid sõnu, kuid masintõlke süsteem töötab fikseeritud pikkusega sõnastikuga. Seega pole võimalik luua sellist nimekirja, kuhu oleks lisatud kõikvõimalikud sõnad, mida mudel sisendiks saada võiks. Üks võimalus, mille puhul süsteem ka tundmatute sõnadega hakkama saab, on sõnade jagamine pseudomorfeemideks [13]. Nimetatud meetodi [13] puhul jaotatakse sõnad pseudomorfeemideks n-ö tükkide koosesinemise sageduse järgi, kui on fikseeritud parameeter, mis täpsustab, kui

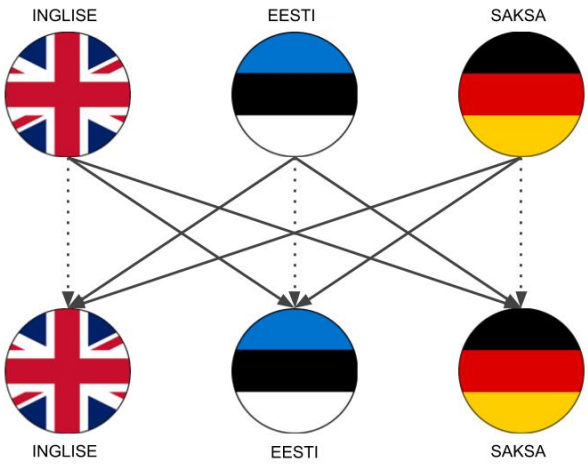
palju pseudomorfeeme kokku olema peaks. Hea parameetri väärtuse korral jäävad enamik tihedamini esinevaid sõnu kokku, mõned lõpud on eraldatud ja väga harva esinevad kooslused on tähthaaval. Näiteks mõte „töös kasutatame transformeri arhitektuuri”jaotatakse järgnevalt „`_töös _kasutame _transform eri _a r hi te k tuuri`”.

3.2 Ühekeelne *zero-shot* tõlge

Taustakirjanduses viidatud artikli [9] põhjal on toimiv meetod läheneda veaparandusele kui vigase teksti tõlkimisele korrektsesse teksti. See idee on selles töös kasutusel teises võtmes.

Loodava süsteemi arhitektuur põhineb *zero-shot* tõlke ideel [14], milles on mitmekeelne masintõlge saavutatud ühe mudeliga. Selles lahenduses on lisatud sünteetiline märgendsõne, millega antakse mudelile teada, mis on soovitud väljundkeel, ja fikseeritud sõnastik on keelteülene.

Selline arhitektuur [14] võimaldab *zero-shot* õpet ehk süsteem on võimeline õppima ja tõlkima keelepaaride vahel, mille näiteid ei ole treenimisandmetes. Näiteks on autorid [14] välja toonud, et kui mudel on õppinud tõlkeid portugali keelest inglise keelde ja inglise keelest hispaania keelde, suudab see ka edukalt tõlkida portugali keelest hispaania keelde, kuigi süsteem ei sisalda portugali-hispaania treenimisandmeid.



Joonis 1. Tõlkesuunad süsteemis, *zero-shot* tõlge märgitud katkendliku joonega.

Ülal kirjeldatud arhitektuuri puhul on võimalik saada ühekeelset tõlget ehk küsida väljundit samas keeles, kui on antud sisendtekst. See on aga vaid siis võimalik, kui keeled, mille puhul meid selline tõlge huvitab, on esindatud nii sisend- kui ka väljundpoolel. Näiteks on võimalik saada eesti keelest eesti keelde tõlget, kui süsteemis on inglise-eesti ja eesti-inglise tõlkesuunad.

Samasse keelde tõlkimine võib esmapilgul tunduda mõttetuna, kuid grammatiliste vigade parandamisel võib see kasuks tulla. Kuna masintõlke süsteemi dekodeerija on keelemudel, mis genereerib väljundit lisatingimuste, masintõlke puhul (teises keeles) sisendteksti põhjal, avaldub omadus parandada tekstis olevaid vigu, sest järgmise elemendi ennustamise juures on süsteem õppinud korrektse teksti põhjal. Vigane tekst pole keeleomane, seega nende sõnade tõenäosus olla lauset jätkav sõna on madalam.

Eeleksperimendis tegime kindlaks, et süsteem õpib ignoreerima infot, mis pole treenimise ajal oluline. Seega on ühekeelse tõlke saavutamiseks vaja süsteemi, mis sisaldaks vähemalt kolme keelt ehk kuut tõlkesuunda. Vähemate keelte puhul on kindel, mis keelne väljund vastab sisendkeelele. Näiteks tõlkesuundadega eesti-inglise ja inglise-eesti süsteem teab alati, et eestikeelse sisendi puhul on väljund ingliskeelne ja vastupidi. See aga kaotab mudeli jaoks keelemärgendi olulisuse, mis on meile oluline ühekeelse väljundi saavutamiseks.

Samal põhimõttel ei toimi ka õppimine näidete põhjal, kus sisend ja väljund on samad. Sel juhul õpib süsteem sisendit kopeerima ja ei õpi korrektset keelt genereerima, kuna see on keerulisem ülesanne, mille õppimine pole nõutud. Piisab ka lihtsalt kopeerima õppimisest.

3.3 Sünteetilised vead

Masintõlkemudelid õpivad nii sisendina saama kui väljundina andma keeleliselt korrektset teksti. Ladusa ja õige lause genereerimine tugeva keelemudeli alusel on osa heast masintõlkest ja võimaldab sel meetodil veaparandust. Samas erinevalt masintõlkest on veaparanduse ülesande täitmiseks saadav sisend vigane tekst, mida tõlkima õppides mudel tähenduse modelleerimise juures ei näe. Seetõttu mõnda tüüpi vigu (nt õigekirjavigu) sisaldavad sõnad/fraasid jäävad mudelile tundmatuks või ajavad segadusse.

Sünteetiliste vigade kasutamine on erinevatel viisidel veaparanduses edukalt kasutust leidnud [6, 15, 5]. Niisiis tahtsime näha, kui edukalt saab seda kombineerida *zero-shot* tõlkega veaparandusega. Selle töö raames on vigade genereerimine motiveeritud ideest „lõhkuda" sisendit, seega pole eesmärk luua võimalikult tõepäraseid keelelisi vigu ega leida optimaalset vigade lisamise tihedust.

Me teeme muudatusi kahel tasemel: muudetakse nii üksikuid tähti kui ka terveid sõnu. Muutmise operatsioone on neli. Nendeks on lausest suvalise elemendi kustutamine, lausesse suvalisse kohta tähe või sõna lisamine, lauses olemasoleva tähe või sõna vahetamine teise vastu ja elemendi liigutamine ühe või rohkema koha võrra edasi või tagasi.

Tihedused, millega vigu sisendisse genereeritakse, panime paika oma loomuliku keeletunnetuse alusel väljundit vaadates.

3.4 Hindamine

Veaparanduste käsitsi hindamine on ajamahukas ülesanne, seega selles töös kasutame automaatseid meetodeid. Hindame kolme peamise veaparanduse automaatse meetodiga: MaxMatch (M2) [16], GLEU [17] ja ERRANT [18]. M2 ja ERRANT on küllaltki sarnased: mõlemad arvutavad täpsust, saagist ja F0.5 skoori. Meetrika GLEU on aga veaparanduseks kohandatud BLEU skoor [19], mida kasutatakse masintõlke hindamiseks [17]. ERRANTi skoore on alles hiljuti hakatud artiklites mainima, mistõttu varasemate töödega võrdlemiseks neid palju kasutada ei saa.

Veaparanduses kasutatakse masinõppes levinud F1 skoori asmel F0.5 skoori, mis loeb täpsust olulisemaks kui saagist. See otsustati 2014. aastal toimunud CoNLLi avaliku võistluse [20] raames. Põhjuseks toodi, et veaparandajaid kasutades teeb väär parandus rohkem kurja kui vea vahele jätmine [20].

Nii ERRANT [18] kui ka M2 [16] võrdlevad hinnatava veaparandussüsteemi tehtud muudatusi inimeste märgendatud parandustega. Seega on nende skooride leidmiseks vaja kindlas formaadis parandusi, mitte lihtsalt sisendlauseid ja parandatud lauseid. Samas GLEU [17] arvutab n-grammide sarnasust, mistõttu saab skoori leida ka hüpoteese ja inimese parandatud tekste võrreldes.

Kuigi skoori arvutamise metoodika on ERRANTi ja M2 arvutamisel sarnane, annab ERRANT enamasti madalamaid tulemusi [18]. Selle põhjuseks toovad autorid välja, et M2 arvutamisel valitakse mõningatel juhtudel võimalusel kunstlikult selle märgendaja parandus, millega saab kõrgema skoori [18].

Paralleelselt kõigi kolme meetodi analüüsimine on motiveeritud nende meetodi eelistest. Nimelt kuna ERRANT annab süsteemi kohta kõige detailsemat infot (tõestelt positiivsed, valepositiivsed, valenegatiivsed, täpsus, saagis ja F0.5), sest kasutab automaatset märgendamist ning võimaldab vaadata sama kirjeldust veatüübiti [18], kasutame seda ingliskeelse veaparanduse kvaliteedi hindamiseks ja erinevate veatüüpide parandamise võrdlemiseks. M2 skoore on kauem artiklites mainitud, kui ERRANTi omi, seega annab M2 parema võrdlusmomendi. GLEU sarnane tõlkimise hindamise meetrikale ega vaja märgendatud andmeid. Kasutame seda tõlkekvaliteediga võrdlemiseks ja eesti keele hindamiseks.

4 Eksperimentide kirjeldus

Selles peatükis kirjeldame esmalt, milliste andmetega me mudeleid treenisime. Seejärel peatume sellel, millised eeltöötlussammud laused enne treenimist läbisid, ja täpsustame treenimise üksikasju. Peatüki lõpus kirjeldame, millistel testhulkadel automaatse hindamise meetodeid rakendame.

4.1 Keeled ja korpused

Mitmekeelse masintõlkemudeli treenimiseks on siin töös kasutusel erinevad kahekeelsed masintõlke paraleelkorpused, mis sisaldavad sama lauset mõlemas keeles. Nagu metoodika alapeatükis selgitasime, siis vigade parandamise seisukohast on oluline, et keeli oleks vähemalt kolm ja kõik keeled oleks nii sisendi kui ka väljundi poolel esindatud. Selleks segame kuue tõlkesuuna laused.

Mudel on õppinud kolme keelepaari andmetel mõlemas tõlkesuunas. Kolmeks keeleks on inglise, eesti ja saksa keel. Kokku on seega kuus tõlkesuunda, milleks on eesti-inglise, inglise-eesti, inglise-saksa, saksa-inglise, saksa-eesti ja eesti-saksa. Need keeled valisime, sest inglise keelt on kõige lihtsam hinnata automaatsete skooridega ja võrrelda teiste meetoditega, eesti keel esindab närvivõrkudel treenimiseks liiga väikese õppijakeelekorpusega keelt ja saksa keelel leidub ühiseid jooni nii eesti kui inglise keelega.

Selleks, et saada kokku treenimiseks piisavalt lauseid ja hoida stiili neutraalsemana, on kasutatavad andmed võetud kolmest erinevast vabalt kättesaadavast korpusest. Nendeks on

- Europarl [21],
- JRC-Acquis [21],
- OpenSubtitles [22].

Esimesed kaks koosnevad ametlikest tekstidest, viimane sisaldab rohkem mitteformaalseid lauseid.

Selle töö raames otsustasime treenitavasse mudelisse igast keelepaarist ja korpusest võtta võrdse koguse andmeid. Lausete arv, mis on igast ülalnimetatud korpusest iga keelepaari jaoks treenimiseks võetud, on 573 500. Lausete hulka piiritleb korpuste suurus, just 573 500 lauset jäi vähimasse hulka (korpusest keelepaari kohta) peale andmete puhastamist ja valideerimishulga eemaldamist. Kõigi keelepaaride laused on dubleeritud, et tõlkida mõlemasse suunda, nt inglise-eesti andmeid on kaks korda. Korra on inglise keel sisend ja eesti keel väljund ning korra vastupidi. Seega on kokku treenimiseks 18 korda (kuus suunda ja kolm korpust) 573 500 ehk 10 323 000 lausepaari. Lausepaaride valik korpusest on suvaline.

4.2 Eeltöötlus

Järgnevalt on esitatud andmete eeltöötluse sammud.

1. Puhastamine (enne suvaliste lausete välja valimist) - tõlkenäide eemaldatakse, kui vähemalt üks pool on tühi sõne, on pikem kui 100 sümbolit või ei sisalda ühtegi tähte.

2. Sõnestamine — kirjavahemärgid eraldatakse sõnadest. Selles töös kasutatud sõnestaja on tööriistast nimega Moses [23].

3. Täheregistri normaliseerimine ehk *true-casing* — muudetakse lause alguse suurtäht väikeseks, kui see pole pärisnimi või muu sõna, mis algaks suure tähega ka lause keskel asetsedes.

4. Pseudomorfeemideks jagamine — sõnad jaotatakse väiksemateks alamjaotusteks tekstis koosesinemise sageduse alusel. Selles töös kasutasime pesudomorfeemide loomiseks SentencePiece'i [24] ja unikaalsete pseudomorfeemide arv on 32 000.

Eestikeelne lause „Tere hommikust!", millele vastab tõlge „Good morning", näeb pärast eeltöötlust välja järgmine.

```
sisend:   _te|to-en re|to-en _hommiku|to-en st|to-en !|to-en
väljund:  _good _morning !
```

Pärast neid eeltöötlussamme genereeritakse väljundkeelt märgendavad sõned ja segatakse kõigi tõlkesuundade ja keelepaaride andmed.

4.2.1 Sünteetiliste vigade lisamine

Müra genereeritakse, kui eeltöötluse sammudes on ära tehtud *true-casing*. Vigu genereeritakse lause kaupa, ühte lausesse genereeritavate vigade arvu leidmiseks võtame Poissioni jaotusest juhusliku valimi ehk sageduse 0.5 korral genereeritakse umbes pooltesse lausetesse vead.

Tabel 1. Kordajad vigade genereerimisel on valitud keeletaju alusel, kuid ei ole leitud optimaalseid väärtusi.

	täht	sõna
lisamine	0.3	0.5
kustutamine	0.9	0.5
asendamine	0.5	0.3
liigutamine	0.5	0.75

Kasutatud on Poissioni jaotust, sest meil on teineteisest sõltumatud sündmused, millele tahame määrata toimumise sageduse. See võimaldab kõige lihtsamalt ennustada mitte lihtsalt seda, kas viga lausesse lisada, vaid ka mitu viga lause kohta luua tuleks. Vea genereerimise sagedus esitub kujul vea tüübi kordaja korrutatud lause pikkuse kordajaga. Vea tüübi kordajad on märgitud tabelis 1. Pikkuse kordaja on lause pikkus skaleeritud 0.5 ja 1 vahele ehk pikima lause kordaja on 1 ja lühima oma 0.5.

4.3 Mudelite treenimine

Treenimiseks kasutasime raamistikku nimega Sockeye [25], milles on implementeeritud transformeril põhinev mudel. SockEye salvestab parameetrid ja hindab mudelit valideerimishulgal kontrollpunktide möödumisel. Selles töös on kontrollpunkti suurus 4000 plokki, ploki suurus on 2048 sõna.

Kasutasime vaikimisi seadistatud parameetreid ehk kuuekihiline transformer ja optimiseerimiseks Adam optimiseerija. Esialgne õpisamm oli 0.0002, mida vähendati iga kord, kui kaheksa kontrollpunkti jooksul valideerimishulgal tulemus (täpsemalt *perplexity*) paranenud polnud. Treenimine lõppes, kui tulemused ei olnud kontrollpunkti jooksul paranenud. Baasmudeli puhul lõppes treenimine 25. epohhi ajal, lisatud müraga mudelil 27. Hüpoteesid on genereeritud kasutatud rindeotsingu parameetrit 5.

4.4 Hindamine

Metoodika peatükis mainisime juba kolme hindamiseks kasutatavat meetodit. Siin selgitame täpsemalt, millistel testhulkadel neid kasutame.

Inglise keelt hindame ERRANTi ja M2 meetrikate alusel kolmel testhulgal:

- CoNLL-2014 [20] (1312 lauset),
- FCE [26] (2695 lauset),
- JFLEG [17] (747 lauset).

GLEU-skoori arvutame vaid viimase jaoks, millele on GLEU algselt loodud ja millel on neli versiooni inimeste parandatud lausetest. See skoor näitab n-grammide kattuvust süsteemi hüpoteesi ja inimeste parandatud lausete vahel [17].

Veatüüpide analüüsimiseks kasutame FCE [26] korpust, sest see on märgendatud ERRANTi jaoks loodud veakategooriatega.

Eesti keelt hindame vaid GLEU alusel, kuna eesti keele jaoks ei ole sobivas formaadis korpust ja praeguse korpuse teisendamine jääb edasiseks tööks. Eesti keele hindamiseks saame kasutada alamhulka Tartu Ülikooli õppijakeelekorpusest [27], kuhu on võetud 800 lauset.

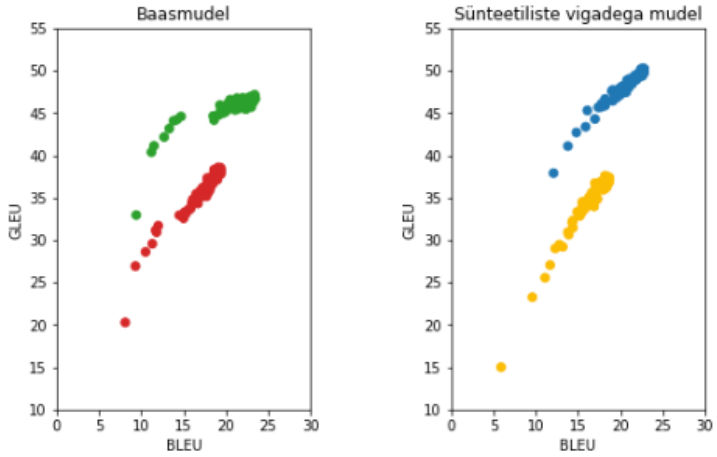
5 Tulemused ja arutelu

Selles peatükis toome välja skoorid, veatüüpide võrdluse ja analüüsi.

5.1 Võrdlus masintõlkekvaliteediga

Võrdlesime skooride kasvamisgraafikuid masintõlke hindamiseks laialdaselt kasutatava BLEU meetrika [19] ja veaparanduse hindamiseks kasutatava GLEU-skoori alusel. GLEU ja BLEU on sarnased ja GLEU skoori saab lihtsasti arvutada ka eesti keelele.

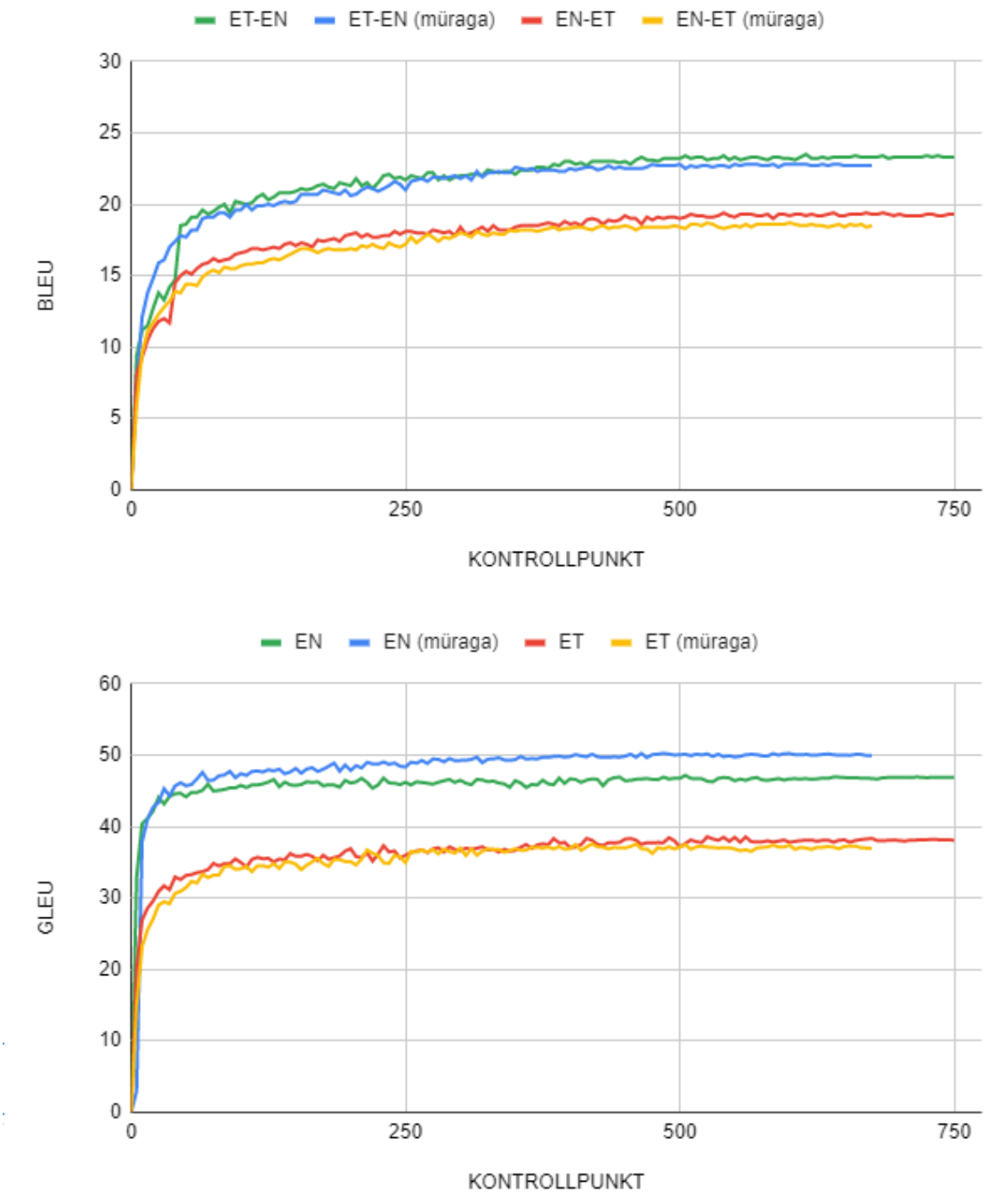
Nagu on näha jooniselt 3, kasvab GLEU skoor masintõlke kvaliteedi suurenemisega. GLEU ja BLEU vahel on tugev seos. Inglise keele veaparanduse ja eesti-inglise tõlke kvaliteedi vahel on Pearsoni korrelatsioonikordaja 0.79 ning eesti keele ja inglise-eesti tõlke puhul 0.95. Lisatud sünteetiliste vigadega mudelite puhul on need numbrid 0.90 ja 0.97. GLEU ja BLEU seost kujutasime joonisel 2.



Joonis 2. GLEU ja BLEU skooride seos, rohelise ja sinisega on tähistatud eesti-inglise tõlkekvaliteet ja inglise keele GLEU ning punase ja kollasega inglise-eesti tõlke ja eesti keele GLEU.

Tabel 2. Masintõlke skoorid, süneetiliste vigade lisamine vähendab tõlkekvaliteeti.

testhulk	baasmudel	sünteetiliste vigadega
wmt18 en-et	19.2	18.6
wmt18 et-en	23.3	22.8
wmt18 en-de	32.4	31.0
wmt18 de-en	32.9	31.9



Joonis 3. BLEU ja GLEU skooride paranemine treenimisel.

Kuigi GLEU ja BLEU on sarnased meetrikad, ei jää nende tulemused samasse vahemikku. Üle 40 BLEU skoori väärtus tähendab juba tipptasemel tõlget, kuid GLEU puhul on juba näiteks muutmata JFLEGi testhulga GLEU-skoori väärtus üle 40.

5.2 Veaparanduse kvaliteet

Parima kontrollpunkti valimiseks arvutasime veaparanduse skoorid iga viienda kontrollpunkti kohta ja valisime kolme testhulga peale kõige kõrgema ERRANT-i F0.5 väärtusega kontrollpunkti. Baasmudeli puhul on see 720. ja lisatud müraga mudeli puhul 570. kontrollpunkt. Kõigi testhulkade parima ja iga testhulga parima kontrollpunkti ERRANT skooride vahe jääb alla 0.5.

Veaparanduse tulemused ERRANTi ja MaxMatch meetrikatele parima kontrollpunkti alusel on baasmudeli kohta esitatud tabelis 3 ja lisatud sünteetiliste vigadega mudeli kohta tabelis 4. GLEU-skooris nii inglise keele kohta JFLEGi testhulgal [17] ja Tartu Ülikooli õppijakeelekorpusel [27] on esitatud tabelis 5.

Tabel 3. Baasmudeli skoorid.

	ERRANT			MaxMatch		
	täpsus	saagis	F0.5	täpsus	saagis	F0.5
JFLEG	25.94	25.31	25.81	49.12	41.65	47.42
CoNLL-2014	25.31	37.73	27.09	36.44	43.20	37.62
FCE	21	25.21	21.73	29.51	27.46	29.08

Tabel 4. Sünteetiliste vigadega mudeli skoorid.

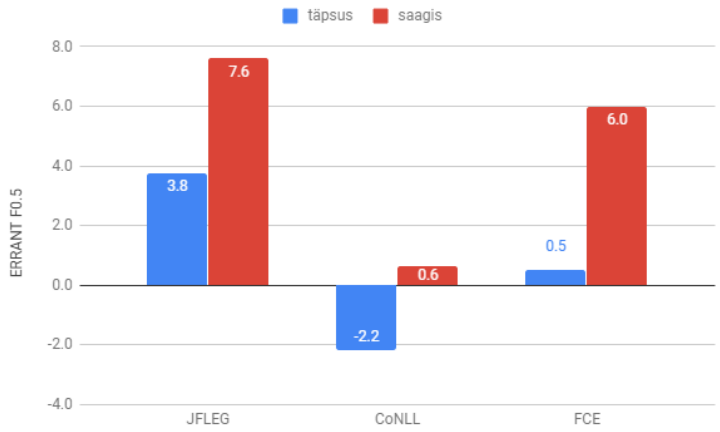
	ERRANT			MaxMatch		
	täpsus	saagis	F0.5	täpsus	saagis	F0.5
JFLEG	29.71	32.91	30.3	53.08	52.15	52.89
CoNLL-2014	23.14	38.37	25.13	34.95	44.66	36.54
FCE	21.53	31.19	22.95	30.53	33.68	31.11

Tabel 5. GLEU-skoorid

	muutmata	baasmudel	sünteetiliste vigadega
JFLEG	40.51	46.91	50.07
UT	27.14	38.14	36.99

Nii baasmudeli kui ka sünteetiliste vigadega mudeli täpsus on kõige kõrgem JFLEGi testhulgal ja kõige madalam FCE testhulgal. Saagis on kõige kõrgem CONLL-2014

testhulgal ja madalaim FCE testhulgal. Kuna JFLEGi testhulgas on vigade arv lausete kohta suurem (3.4 viga lause kohta) kui FCE testhulgas (1.7 viga lause kohta) ja FCE testhulga puhul on nii saagis kui ka täpsus madalamad kui teistel testhulkadel, võib oletada, et mudel saavutab paremaid skoore vigasema teksti peal. Nii korrektsete kui ka valede muudatuste arv on FCE hulgal nii lausete kui ka vigade arvu kohta väiksem. Sellest tuleneb madalam saagis ja kuigi valepositiivseid on vähem, siis mitte nii palju, et kombineerituna väiksema paranduste arvuga oleks täpsus teiste hulkadega samal pulgal.



Joonis 4. Sünteetiliste vigadega mudeli ja baasmudeli ERRANT F0.5 skooride vahe.

Vaadates täpsemalt, mis juhtus sünteetiliste vigade lisamisel, näeme, et kõigi testhulkade puhul tõusis saagis. CoNLLi testhulgal on saagiste vahe väike, tõeselt positiivsete paranduste arv suurenes seal vaid ligi 30 võrra (baasmudelil 1033, lisatud vigadega mudelil 1062), kuid valepositiivsete arv tõusis ligi kuuendiku võrra, mistõttu on täpsus langenud. Ka teiste testhulkade puhul tõusis nii tõeselt positiivsete kui ka valepositiivsete arv, kuid suurenes korrektsete paranduste arvu osakaal kõigi muudatuste hulgas (tõusis täpsus).

Võrreldes valminud artikli [8] raames treenitud stiili eristava inglise, eesti ja läti keelte vahel tõlkiva mudeliga, on selle mudeli CoNLLi M2 ja JFLEGi GLEU skoorid kõrgemad ning eesti keele GLEU sama (teisi skoore pole välja toodud). Sellest võib eeldada, et mudelile ei mõjunud halvasti treenimisandmete vähendamine või aitas veaparanduse võimet parandada stiilimärgendi kaotamine või läti keele vahetamine inglise keelega sarnasema keele vastu.

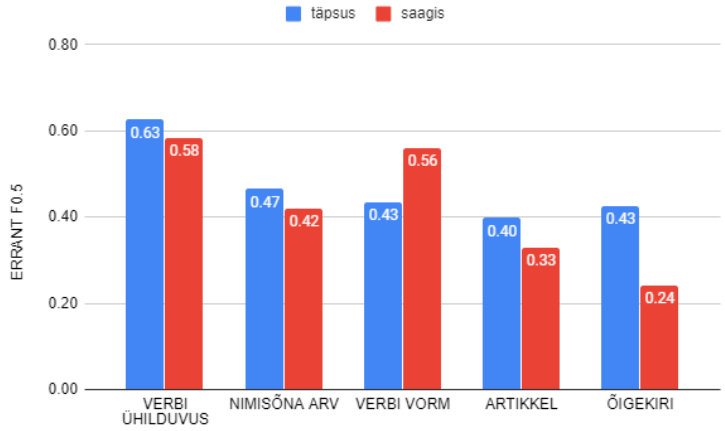
Võrreldes taustakirjanduses mainitud keelemudelil põhinevale veaparandusele, mis ei kasuta märgendatud andmeid[7], on meie mudeli saagis CoNLLi ja FCE testhulkadel oluliselt kõrgem. Müra lisamisel saavutab meie mudel ka JFLEGi testhulgal parema saagise. Samas on mainitud lahendusel [7] märkimisväärselt kõrgem täpsus, mistõttu on neil ka üldskoorid paremad. Võrreldes juhendatud süsteemidega, jäävad meie mudeli skoorid

alla. See on ka mõistetav, kuna need on kasutanud õppimiseks veaparandusnäiteid.

5.3 Võrdlus veatüübiti

ERRANT kasutab automaatset märgendamist, seega on võimalik saada ka detailset tagasisidet veatüüpide kaupa: tõeselt positiivsete, valepositiivsete ja valenegatiivsete koguarv ning täpsus, saagis ja F-skoor. Selles peatükis võrdleme vaid suuremaid kategooriaid, kus on vigu testhulgas üle 50. Väiksemate veahulkadega kategooriate puhul on saagis ja täpsus liiga eksitavad. Esmalt vaatame, milliste vigadega saab baasmudel hästi hakkama ja millistega jääb hätta. Seejärel võrdleme baasmudeli ja lisatud sünteetiliste vigadega mudeli skoore veatüübiti. Detailsemaid andmeid on võimalik näha lisas I.

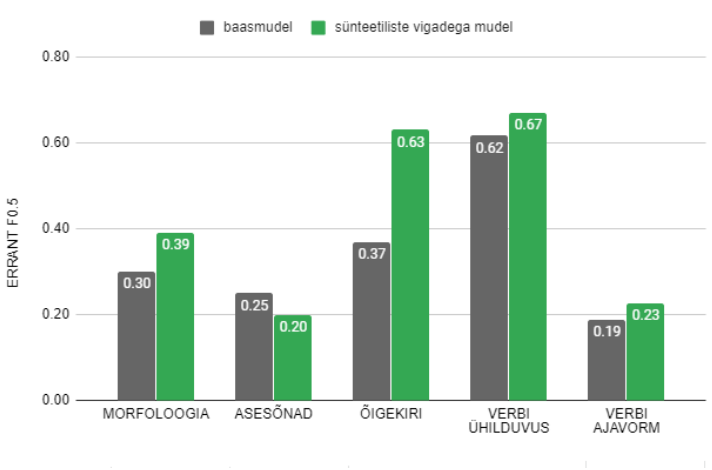
Kõige paremini saab mudel hakkama grammatiliste vigadega. Jooniselt 5 on näha veakategooriad, millega süsteem edukalt hakkama saab. Nendeks on tegusõna ühilduvus nimisõnaga (nt „he walk”), nimisõna mitmuse tunnus („cat”või „cats”), tegusõna vorm (nt „to walk”või „walking”) ja artiklid („a”, „an”või „the”). Samuti on mudel küllaltki edukas õigekirjavigadega, kuid nende puhul on täpsus võrreldav, kuid saagis madalam kui teiste edukaimate veakategooriate puhul.



Joonis 5. Viis kategooriat, millega mudel kõige paremini hakkama saab

Kõige halvemini tuleb mudel toime üldisemate sõnavaliku vigadega, nagu vale nimisõna, omadussõna või tegusõna valimine ja ümbersõnastused. Nende tüüpide analüüsist selgub ka suur valepositiivsete arv, eriti kategoorias „muu”, mis sisaldab ümbersõnastamisi ja muid vigu, mis teistesse kategooriatesse ei sobi.

Joonisel 6 on viis veakategooriat, kus baasmudeli ja sünteetiliste vigadega mudeli ERRANTi skooride erinevus on kõige suurem. Kõige rohkem on vigade lisamine tõstnud õigekirja parandamise skoori.



Joonis 6. Veatüüpide kaupa ERRANTi F-skooride võrdlus – kõige rohkem muutunud kategooriad

Tabelist 6 on näha, et kuigi ka valepositiivsete arv on veidi langenud, siis õigekirjavigade parandamise skoori tõus on peamiselt tingitud ligi 150st lisandnud õigest õigekirjavea parandusest.

Tabel 6. Õigekirja vigade parandamine: tõeselt positiivsete (TP), valepositiivsete (VP) ja valenegatiivsete (VN) koguarv ning täpsus, saagis ja F-skoor

	TP	VP	VN	täpsus	saagis	F0.5
baasmudel	109	147	343	42.58	24.12	36.92
sv-mudel	253	135	199	65.21	55.97	63.12

Dekoderija on õppinud tähenduse ja konteksti väljendavate esituste põhjal looma väljundit, millel on loomulik keele struktuur, mistõttu võib selgelt grammatiliselt valede kohtade (nt ühildumine ja mitmus) parandamine olla kergem, sest sõnad on tähenduselt lähedased. Samas võivad vale sõnavalik või tundmatud (nt valesti kirjutatud) sõnad rohkem segada lause tähendusliku representatsiooni loomisel ning ajada mudeli rohkem segadusse. Lisaks on juba mõne näite ja artikli [8] raames tehtud kvalitatiivse analüüsi põhjal näha, et süsteem asendab sõnu sünonüümidega, mis on ka mõistetav, sest mudelil ei ole ühtegi reeglit või näidet, mis ütleks, et tuleb eelistada sisendlause sõnavalikut ja struktuuri.

6 Kokkuvõte

Selles töös tutvustasime vigade parandamiseks uut mitmekeelse masintõlke ühekeelsel rakendamisel põhinevat meetodit, mis ei vaja treenimiseks veaparandusnäiteid, mistõttu sobib rakendamiseks ka keeltele, mille jaoks pole juhendatud veaparandumudeli õpetamiseks piisavalt suurt korpust. Süsteemi treenimiseks on vaja ainult tõlkelauseid ja mudel on võimeline vigu parandama mitmes keeles.

Sissejuhatuses esitasime küsimused, millele vastuseid otsisime. Nendest esimene oli „Kui hea on mudeli veaparanduskvaliteet ja kas see tõuseb sünteetiliste vigadelisamisel?". Küsimuse esimesele poolele pole vastus ühene. Nimelt on süsteemil korralik saagis, kuid kitsaskohaks on täpsus. Mudel teeb palju muudatusi korrektsesse teksti ja on sellisel kujul ebausaldusväärne.

Kuigi sünteetiliste vigade lisamine teksti suurendas korrektsesse teksti lisatavate muudatuste hulka, siis tõuseb oluliselt ka õigete paranduste arv. Inglise keele puhul kahest testhulgast kolmes suurenes täpsus ehk pärast vigade lisamist tõusis korrektsete paranduste osakaal kõigi paranduste hulgas. Sellele tuginedes võib öelda, et sünteetiliste vigade lisamine võib aidata veaparanduse kvaliteeti tõsta.

Teisena otsisime vastust küsimusele „Milliste vigade parandamisega saab kirjeldatud meetod hästi hakkama ja millistega jääb hätta?". Kõige edukam on mudel grammatiliste vigade parandamisega, näiteks kui tekstis on ühildumise probleeme, siis parandab süsteem neid edukalt ja ei loo teksti juurde selliseid vigu. Õigekirjavigade parandamise täpsus on baasmuselil kõrge, kuid saagis madalam, sünteetiliste vigade lisamisel tõusis korrektsete paranduste osakaal tublisti.

Mudel on aga rohkem kimbatuses sõnavalikuga, nendes kategooriates on nii madal täpsus kui ka saagis. Üdisemates leksikaliste valikudega seotud kategooriates paistab silma suur valepositiivsete arv, mudel vahetab korrektseid sõnu sisendis sünonüümide või tähenduselt lähedaste sõnade vastu. See on seletatav, sest süsteem on õppinud genereerima tähenduselt sarnast teksti, kuid mitte hoidma sama sõnastust. Ühekeelseid näiteid pole treenimisandmete hulgas.

Viimase küsimusena uurisime „Kas veaparanduse kvaliteet tõuseb mudeli treenimisel tõlkimise võime paranemisega sarnaselt?". Kuigi inglise keele veaparanduskvaliteet tõusis esialgu kiiremini, on siiski tõlke- ja veaparanduskvaliteedi vahel tugev seos. Kui tõuseb tõlke kvaliteet, siis parandab mudel ka paremini vigu.

Lõpetuseks võib öelda, et süsteemil on potentsiaali, sest veaparanduse saagis on korralik ja edasise tööga on võimalik leida meetodeid mudeli „loomingulisuse"piiramiseks. Erinevalt enamikest veaparanduse meetoditest, mis eeldavad kas keelespetsiifilisi teadmisi või õppijakeelekorpuse kogumist, on meie välja pakutud meetod vähese vaevaga rakendatav paljudele keeltele.

Viidatud kirjandus

[1] Clément, L., Gerdes, K. ja Marlet, R. A Grammar Correction Algorithm – Deep Parsing and Minimal Corrections for a Grammar Checker (juuli 2009).

[2] Yuan, Z. ja Felice, M. Constrained Grammatical Error Correction using Statistical Machine Translation. Teoses: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Sofia, Bulgaria: Association for Computational Linguistics, august 2013, lk. 52–61. URL: https://www.aclweb.org/anthology/W13-3607. (08.05.20).

[3] Bryant, C. *et al.* The BEA-2019 Shared Task on Grammatical Error Correction. Teoses: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, august 2019, lk. 52–75. URL: https://www.aclweb.org/anthology/W19-4406. (08.05.20).

[4] Vaswani, A. *et al.* Attention is All you Need (2017). Toim. Guyon, I. *et al.*, lk. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf. (08.05.20).

[5] Felice, M. ja Yuan, Z. Generating artificial errors for grammatical error correction. Teoses: *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, aprill 2014, lk. 116–126. URL: https://www.aclweb.org/anthology/E14-3013. (08.05.20).

[6] Grundkiewicz, R., Junczys-Dowmunt, M. ja Heafield, K. Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data. Teoses: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, august 2019, lk. 252–263. URL: https://www.aclweb.org/anthology/W19-4427. (08.05.20).

[7] Bryant, C. ja Briscoe, T. Language Model Based Grammatical Error Correction without Annotated Training Data. Teoses: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, juuni 2018, lk. 247–253. URL: https://www.aclweb.org/anthology/W18-0529. (08.05.20).

[8] Korotkova, E. *et al.* Grammatical Error Correction and Style Transfer via Zero-shot Monolingual Translation (2019). URL: https://arxiv.org/abs/1903.11283. (08.05.20).

[9] Junczys-Dowmunt, M. *et al.* Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task. Teoses: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, juuni 2018, lk. 595–606. URL: https://www.aclweb.org/anthology/N18-1055. (08.05.20).

[10] Bahdanau, D., Cho, K. ja Bengio, Y. Neural machine translation by jointly learning to align and translate. Teoses: *Proceedings of the 3rd International Conference on Learning Representations, ICLR*. San Diego, CA, USA, 2015. (08.05.20).

[11] Gehring, J. *et al.* Convolutional Sequence to Sequence Learning. Teoses: *Proceedings of the 34th International Conference on Machine Learning*. Toim. Precup, D. ja Teh, Y. W. Köide 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, juuni 2017, lk. 1243–1252.

[12] Kalchbrenner, N. ja Blunsom, P. Recurrent Continuous Translation Models. Teoses: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, oktoober 2013, lk. 1700–1709. URL: https://www.aclweb.org/anthology/D13-1176. (08.05.20).

[13] Sennrich, R., Haddow, B. ja Birch, A. Neural Machine Translation of Rare Words with Subword Units. Teoses: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, august 2016, lk. 1715–1725. URL: https://www.aclweb.org/anthology/P16-1162. (08.05.20).

[14] Johnson, M. *et al.* Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics* 5 (2017), lk. 339–351. URL: https://www.aclweb.org/anthology/Q17-1024. (08.05.20).

[15] Xie, Z. *et al.* Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. Teoses: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, juuni 2018, lk. 619–628. URL: https://www.aclweb.org/anthology/N18-1057. (08.05.20).

[16] Dahlmeier, D. ja Ng, H. T. Better Evaluation for Grammatical Error Correction. Teoses: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, juuni 2012, lk. 568–572. URL: https://www.aclweb.org/anthology/N12-1067. (08.05.20).

[17] Napoles, C., Sakaguchi, K. ja Tetreault, J. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. Teoses: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, aprill 2017, lk. 229–234. URL: https://www.aclweb.org/anthology/E17-2037. (08.05.20).

[18] Bryant, C., Felice, M. ja Briscoe, T. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. Teoses: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, juuli 2017, lk. 793–805. URL: https://www.aclweb.org/anthology/P17-1074. (08.05.20).

[19] Papineni, K. *et al.* BLEU: A Method for Automatic Evaluation of Machine Translation. Teoses: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania, 2002, lk. 311–318. (08.05.20).

[20] Ng, H. T. *et al.* The CoNLL-2014 Shared Task on Grammatical Error Correction. Teoses: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, juuni 2014, lk. 1–14. URL: https://www.aclweb.org/anthology/W14-1701. (08.05.20).

[21] Tiedemann, J. Parallel Data, Tools and Interfaces in OPUS. Inglise keel. Teoses: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Toim. Chair), N. C. (*et al.* Istanbul, Turkey: European Language Resources Association (ELRA), mai 2012. (08.05.20).

[22] Lison, P. ja Tiedemann, J. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. Teoses: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), mai 2016, lk. 923–929. URL: https://www.aclweb.org/anthology/L16-1147. (08.05.20).

[23] Koehn, P. *et al.* Moses: Open Source Toolkit for Statistical Machine Translation. Teoses: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, juuni 2007, lk. 177–180. URL: https://www.aclweb.org/anthology/P07-2045. (08.05.20).

[24] Kudo, T. ja Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Teoses: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations. Brussels, Belgium: Association for Computational Linguistics, november 2018, lk. 66–71. URL: https://www.aclweb.org/anthology/D18-2012. (08.05.20).

[25] Hieber, F. *et al.* Sockeye: A Toolkit for Neural Machine Translation. *CoRR* abs/1712.05690 (2017). arXiv: 1712.05690. URL: http://arxiv.org/abs/1712.05690. (08.05.20).

[26] Yannakoudakis, H., Briscoe, T. ja Medlock, B. A New Dataset and Method for Automatically Grading ESOL Texts. Teoses: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, juuni 2011, lk. 180–189. URL: https://www.aclweb.org/anthology/P11-1019. (08.05.20).

[27] Rummo Ingrid ja Praakli, K. TÜ eestikeele (võõrkeelena) osakonna õppijakeele tekstikorpus (2017), lk. 12–13. (08.05.20).

Lisad

I. Veakategooriate skoorid

Tabel 7. Baasmudeli ERRANT: tõeselt positiivsete (TP), valepositiivsete (VP) ja valenegatiivsete (VN) koguarv ning täpsus, saagis ja F-skoor

Category	TP	FP	FN	P	R	F0.5
ADJ	7	80	77	0.0805	0.0833	0.081
ADJ:FORM	1	1	9	0.5	0.1	0.2778
ADV	5	138	82	0.035	0.0575	0.0379
CONJ	0	2	30	0.0	0.0	0.0
CONTR	0	344	3	0.0	0.0	0.0
DET	206	312	419	0.3977	0.3296	0.3819
MORPH	39	100	52	0.2806	0.4286	0.3014
NOUN	10	299	192	0.0324	0.0495	0.0348
NOUN:INFL	31	7	4	0.8158	0.8857	0.8289
NOUN:NUM	73	84	101	0.465	0.4195	0.4551
NOUN:POSS	4	21	27	0.16	0.129	0.1527
ORTH	111	411	103	0.2126	0.5187	0.2411
OTHER	26	850	554	0.0297	0.0448	0.0318
PART	1	21	7	0.0455	0.125	0.0521
PREP	126	299	351	0.2965	0.2642	0.2894
PRON	24	59	124	0.2892	0.1622	0.25
PUNCT	130	539	341	0.1943	0.276	0.2065
SPELL	109	147	343	0.4258	0.2412	0.3692
VERB	31	226	212	0.1206	0.1276	0.122
VERB:FORM	93	121	73	0.4346	0.5602	0.455
VERB:INFL	6	2	5	0.75	0.5455	0.6977
VERB:SVA	52	31	37	0.6265	0.5843	0.6176
VERB:TENSE	38	155	194	0.1969	0.1638	0.1892
WO	24	65	62	0.2697	0.2791	0.2715

Tabel 8. Lisatud sünteetiliste vigadega mudeli ERRANT: tõeselt positiivsete (TP), valepositiivsete (VP) ja valenegatiivsete (VN) koguarv ning täpsus, saagis ja F-skoor

Category	TP	FP	FN	P	R	F0.5
ADJ	8	83	76	0.0879	0.0952	0.0893
ADJ:FORM	2	3	8	0.4	0.2	0.3333
ADV	9	187	78	0.0459	0.1034	0.0517
CONJ	0	22	30	0.0	0.0	0.0
CONTR	1	297	2	0.0034	0.3333	0.0042
DET	229	351	396	0.3948	0.3664	0.3888
MORPH	50	87	41	0.365	0.5495	0.3912
NOUN	14	466	188	0.0292	0.0693	0.033
NOUN:INFL	30	5	5	0.8571	0.8571	0.8571
NOUN:NUM	75	83	99	0.4747	0.431	0.4653
NOUN:POSS	6	29	25	0.1714	0.1935	0.1754
ORTH	119	384	95	0.2366	0.5561	0.2673
OTHER	57	1201	523	0.0453	0.0983	0.0508
PART	1	22	7	0.0435	0.125	0.05
PREP	137	396	340	0.257	0.2872	0.2626
PRON	31	126	117	0.1975	0.2095	0.1997
PUNCT	141	663	330	0.1754	0.2994	0.1912
SPELL	253	135	199	0.6521	0.5597	0.6312
VERB	39	289	204	0.1189	0.1605	0.1254
VERB:FORM	91	111	75	0.4505	0.5482	0.4671
VERB:INFL	7	1	4	0.875	0.6364	0.814
VERB:SVA	49	20	40	0.7101	0.5506	0.6712
VERB:TENSE	43	136	189	0.2402	0.1853	0.2268
WO	27	74	59	0.2673	0.314	0.2755

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Agnes Luhtaru**,
*(*autori nimi)

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

   Grammatiliste vigade parandamine mitmekeelse neuromasintõlkega,
   *(*lõputöö pealkiri)

   mille juhendaja on Mark Fišel,
   *(*juhendaja nimi)

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

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

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

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

Agnes Luhtaru
08.06.2020