LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

LATVIEŠU VALODAS MORFOLOĢISKĀ MARĶĒŠANA, IZMANTOJOT DZIĻĀS MAŠĪNMĀCĪŠANĀS METODES

BAKALAURA DARBS

Autors: Artūrs Treimanis
Studenta apliecības Nr.: at19039
Darba vadītājs: Dr. dat. Pēteris Paikens

RĪGA 2023

ANOTĀCIJA

Viens no aktuālākajiem dabiskās valodas apstrādes uzdevumiem ir teksta morfoloģiska marķēšana, kuras ietvaros katram teikuma vārdam tekstā tiek piešķirts marķējums, kas atspoguļo vārda valodnieciskās īpašības. Šī darba mērķis bija realizēt modernāko dabiskās valodas apstrādes risinājumu pielāgošanu latviešu valodai. Realizācijā tika izmantoti BERT modeļi, kas apmācīti ar latviešu valodas teksta korpusiem, vārdu konteksta analīzei un daudznozīmības problēmas adresēšanai, un ilgās īstermiņu atmiņas (LSTM) mākslīgā neironu tīkla kombinācija.

Darba ietvaros tika izstrādāts automātiskas marķēšanas modelis, kas vārdšķiras atpazīšanas uzdevumā sasniedza 98.59% precizitāti un pilnā morfoloģiskā marķējuma prognozēšanā - 93.12% precizitāti.

Atslēgvārdi: mašīnmācīšanās, dabiskās valodas apstrāde, morfoloģiska marķēšana, BERT, LSTM, neironu tīkli.

ABSTRACT

The topic of this study is *Latvian language morphological tagging using deep learning method*s.

One of the more relevant natural language processing tasks is morphological tagging, during which each word in a sentence gets assigned a tag that describes its linguistic features. The goal of this study was to execute state-of-the-art natural language processing solution adaptation for Latvian language. For the implementation, BERT models trained on Latvian text corpora were used for word contextual analysis and polysemy problem addressing, as well as a long short-term memory (LSTM) artificial neural network combination was used.

In the scope of the study a model for automatic tagging was developed. The model achieved accuracy of 98.59% for part of speech tagging and 93.12% accuracy for full morphological tagging.

Keywords: machine learning, natural language processing, morphological tagging, BERT, LSTM, neural networks.

SATURA RĀDĪTĀJS

APZĪMĒJUMU SARAKSTS

Atbirumu (*dropout*) slāņi - modeļa arhitektūras slāņi, kas pēc nejaušības principa atspējo noteiktu daudzumu tīkla neironu, ar mērķi samazināt modeļa pakļaujamību pārpielāgoties.

BERT (*Bidirectional Encoder Representations from Transformer*s) - transformatoru-balstīti neironu tīklu modeļi, kontekstuālas informācijas iekodēšanai teksta vārdu skaitliskās vērtībās.

CRF (*conditional random field*) - modeļi, kuru darbība ietver statistisku modelēšanu, bieži izmantoti prognozēšanas uzdevumos, kuros ir svarīga kontekstuāla informācija vai apskatāmajai prognozei blakusesošo elementu stāvoklis.

CUDA (*Compute Unified Device Architecture*) - paralēlās programmēšanas platforma, kas nodrošina iespēju izmantot NVIDIA video kartes skaitļošanas procesiem, izmantojot to lietojumprogrammas saskarnes.

JSON (*JavaScript Object Notation*) - tekstuāls datu reprezentācijas formāts, kura galveno struktūru sastāda atslēgu-vērtību pāri.

LSTM (*Long Short-Term Memory*) - rekurents neironu tīkla paveids, kura galvenās iezīmes ir selektīvi iemācīties un aizmirst informāciju.

Pārpielāgošana (*Overfitting*) - process, kurā apmācāmais mašīnmācīšanās modelis izgūst pārāk lielu informācijas daudzumu par likumsakarībām apmācības datos, kas tā prognozes veikšanu padara pārāk nevispārinātu.

PyTorch - programmēšanas valodas Python satvars, kas paredzēts neironu tīklu modeļu izstrādei, apmācībai un testēšanai.

Python - augsta līmeņa programmēšanas valoda, plaši izmantota datu zinātnes un pētniecības uzdevumos.

SSH (*Secure Shell*) - tīmekļa protokols droša savienojuma nodrošināšanai starp divām ierīcēm.

Tokenizators - datu priekšapstrādes solis, kas nodrošina teksta pārveidošanu vienkāršā skaitliskā reprezentācijā.

Word2Vec - tehnoloģija, kas izmantota vārdu skaitliskas reprezentācijas iegūšanai vektortelpā, saglabājot to semantiskās attiecības, izmantojot neironu tīklus.

IEVADS

Viens no primārajiem uzdevumiem naturālas valodas apstrādē ir morfoloģiska marķēšana, kuras galvenais mērķis ir atpazīt vārdu valodnieciskās pazīmes, kā piemēram, to vārdšķiru. Morfoloģiski marķējumi ir simbolu virkne, kas satur visu informāciju par runas vārda valodnieciskajām pazīmēm kompaktā formātā.

Morfoloģiskās marķēšanas pielietojums atrodams vairākos procesos, ieskaitot sentimenta analīzi un mašīntulkošanu, bet tā realizācija ir sarežģīta valodās, kurās ir iespējams vārdus locīt dažādos veidos un vārdiem piemīt daudznozīmības iezīmes, kuros to nozīme un lietojums var atšķirties, balstoties uz konkrēto kontekstu.

Šādā, augstas lokāmības un daudznozīmības, valodu kopumā iekļaujama arī latviešu valoda. Latviešu valodā, piemēram, darbības vārdiem piemīt trīs konjugāciju kategorijas un lietvārdi kategorizējami sešās deklinācijās.

Šī **darba mērķis** ir izstrādāt un izvērtēt automātiskas marķēšanas modeli latviešu valodai. Tiek sagaidīts, ka modelis spēj piešķirt marķējumu katram vārdam teikumā, balstoties uz tā kontekstu un valodnieciskajām pazīmēm. Tiek izvirzīta hipotēze, ka ir iespējams izstrādāt modeli, izmantojot mašīnmācīšanās paņēmienus, kurš sniedz augstus precizitātes rādītājus teksta marķēšanai latviešu valodā.

Darba uzdevumos ietilpst:

- veikt literatūras apskatu un analīzi par eksistējošiem risinājumiem darbā izvirzītajai problēmai;
- izvēlēties un aprakstīt izvēlēto mašīnmācīšanās pieeju praktiskai uzdevuma izpildei;
- izvēlēties atbilstošu datu kopu veicamajam uzdevumam un veikt datu prims-apstrādi;
- izmantojot mašīnmācīšanās metodi, izstrādāt modeli, kas pēc trenēšanas, izmantojot marķētu datu kopu, spēj piešķirt marķējumus iepriekš neanalizētam tekstam latviešu valodā;
- salīdzināt izstrādātā modeļa sniegumu ar citiem eksistējošiem risinājumiem;
- veikt kļūdu analīzi un izvirzīt secinājumus.

Darbā izmantotās metodes:

- zinātniskās literatūras apskats par pētījumiem latviešu un citās valodās izvēlētās tēmas ietvaros;
- praktiska darba uzdevuma izpilde, izmantojot mašīnmācīšanās algoritmus.

Darba struktūra.

Darba saturu veido ievads, 4 nodaļas, rezultātu nodaļa un secinājumi. Ievadā pamatota darba aktualitāte, izvirzīti darba mērķi un veicamie uzdevumi, kā arī definētas darbā izmantotās pētniecības metodes. Pirmajā nodaļā veikts problēmas apkopojums, kurā raksturota problēma, izvirzītas problēmas risinājumu idejas, kā arī apskatīti sasniegumi latviešu valodas apstrādē. Otrajā nodaļā apskatītas pasaulē modernākās metodes problēmas risināšanai un apskatītas alternatīvas metodes, kas neizmanto mašīnmācīšanos. Trešajā nodaļā tiek definēts tehniski ieviešamais risinājums, kā arī definētas eksperimentāli pārbaudāmās hipotēzes. Ceturtajā nodaļā aprakstīts eksperimentālais ietvars - izmantotā datu kopa, eksperimentālā vide, veiktā datu priekšapstrāde un eksperimentu tehniskie iestatījumi un apmācības process. Piektajā nodaļā strukturizēti apkopoti un apspriesti novērotie eksperimentu rezultāti. Secinājumos veikts darba kopsavilkums, apkopotas atziņas par darba rezultātiem un izvirzīti priekšlikumi turpmākai pētniecībai un risinājuma uzlabošanai.

1. PROBLĒMAS APSKATS

1.1. Problēma

Latviešu valoda ir relatīvi sarežģīta valoda, kurā vārdiem eksistē daudz locījumu variāciju. Lai gan morfoloģiska marķēšana ir globāli aktuāls temats, un dabiskās valodas apstrādes sfēra ir strauji attīstījusies pēdējo gadu laikā, marķēšanu latviešu valodā īpaši aktuālu padara tās sarežģītība, vārdu augstās lokāmības un daudznozīmības dēļ.

Ņemot vērā augsto locāmību, latviešu valodā bieži sastopami gadījumi, kuros neapsverot vārda kontekstu, ir sarežģīti vārdiem piešķirt marķējumu vārdu daudznozīmības dēļ. Piemēram, vārds "sienu" var būt lietvārds akuzatīva locījumā ("mājas sienu") vai darbības vārds pirmās personas īstenības izteiksmē ("sienu kurpi").

Lai gan šai problēmai eksistē mūsdienīgi tehnoloģiski risinājumi, kuros, kā iespēju risināt daudznozīmības problēmu, autori, piemēram, katram daudznozīmes vārdam piedāvā visu atbilstošo marķējumu alternatīvas [1], tiek izvirzīts pieņēmums, ka, izmantojot dziļās mašīnmācīšanās metodes, iespējams nodrošināt viena precīza marķējuma rezultātus.

Tāpat, ņemot vērā tēmas aktualitāti, jāmin, ka dziļās mašīnmācīšanās metodes ir tikušas izmantotas citu valodu teksta marķēšanai, taču šī brīža modernākie morfoloģiskās marķēšanas risinājumi nav adaptēti un realizēti latviešu valodas marķēšanā.

1.2. Risinājums

Acīmredzama pieeja morfoloģiskai marķēšanai ir - veikt teksta marķēšanu, pielietojot manuālu cilvēka darbu, katram vārdam pierakstot tā morfoloģisko marķējumu ar roku, balstoties uz vārda kontekstu. Lai gan šī pieeja ir arī fundamentāli nepieciešama ("gudro" risinājumu ietvaros ir nepieciešamas valodnieku marķētas kopas [2]), tomēr, šis process ir laikietilpīgs, sevišķi liela apjoma datu kopu apstākļos.

Šī procesa automatizēšanai iespējams pielietot mašīnmācīšanās paņēmienus. Izmantojot mašīnmācīšanos, ir iespējams izgūt izejas datus no datorsistēmām, neizstrādājot specifiskas instrukcijas un pārveidojumus, kas ļautu datorsistēmām radīt atbilstošu izvadi, balstoties uz ievades datiem. Izmantojot marķētu datu kopu un pārraudzītās mašīnmācīšanās metodes, iespējams veiksmīgi apmācīt modeli morfoloģiskai marķēšanai iepriekš neredzētām datu kopām. Marķēta datu kopa norāda sagaidāmos izvades datus katram ievades datu teikumam, kas ļauj modelim atrast likumsakarības starp ievades atribūtiem, vārda kontekstu un atbilstošajiem izvades datiem.

Ņemot vērā, ka sagaidāmā izvade ir katra vārda morfoloģisks marķējums, šis ir raksturojams kā klasifikācijas uzdevums. Katrs marķējums sastāv no rakstzīmju kopas, kurā

katra rakstzīme simbolizē kādu konkrētā vārda valodniecisko pazīmi. Balstoties uz to, ka tiek sagaidīts, ka modelis spēj izvadē norādīt vairāk kā divas valodnieciskās pazīmes, lai izveidotu morfoloģisko marķējumu un katra pazīme uztverama kā klase - šis ir konkrētāk klasificējams kā vairākklašu klasifikācijas uzdevums.

Pēc modeļa apmācīšanas tiek sagaidīts, ka modelis patstāvīgi spēj radīt morfoloģiskus marķējumus patvaļīgam ievades tekstam latviešu valodā.

Lai spētu secināt, cik veiksmīgs ir izvēlētais risinājums darba problēmai, ir nepieciešams veikt modeļa rezultātu analīzi. Analīzes ietvaros, modeļa darbība jāizpilda, izmantojot iepriekš sagatavotu testa datu kopu, kurā norādīti pareizie vārdu marķējumi. Pēc izvaddatu ieguves, modeļa izvadītie marķējumi jāsalīdzina ar sagaidītajiem un jāaprēķina modeļa izvades precizitātes rādītāji.

1.3. Sasniegumi latviešu valodas apstrādē

Pēdējo gadu laikā ir novērojama augoša interese dabiskās valodas apstrādes rīku un resursu attīstībā latviešu valodai. Neskatoties uz pētnieciskās informācijas pieaugumu dabiskās valodas apstrādes uzdevumos kā mašīntulkošana, sentimenta analīze u.c., šī informācija tāpat ir ierobežota, salīdzinot ar citās valodās pieejamo informāciju.

Savā 2016. gada pētījumā, arī A. Znotiņš identificēja, ka dabiskās valodas daudznozīmība ir pamatproblēma dabiskās valodas apstrādes uzdevumos [3]. Pētījuma ietvaros, autors apskatījis 5 dažādus modeļu veidus dažādos dabiskās valodas apstrādes uzdevumos, starp kuriem viens no uzdevumiem bija vārdšķiru atpazīšanas uzdevums. Rezultātā tika konstatēts, ka morfoloģiskajā marķēšanā labākie rezultāti tika iegūti, izmantojot LSTM tīklu ar SSG (no angļu valodas - *Structured Skiprams*), apmācot 200 dimensiju jēdzientelpu ar 5 vārdu kontekstu.

Viens no mašīnmācīšanās rīkiem, kas guvis augošu popularitāti tā izmantojamībā, pateicoties augstajiem lietojamības rezultātiem, kas vektorus spēj piešķirt katrai daudznozīmīgā vārda nozīmei, ir BERT modeļi. BERT ir transformatoru modeļi, kas apmācīti, izmantojot lielus teksta korpusus, cenšoties paredzēt zudušus vārdus ieejas tekstā, balstoties uz teksta kontekstu. 2020. gadā tika publicēts BERT modelis, kas apmācīts, izmantojot latviešu valodas korpusus [4]. Šis modelis uzrādīja uzlabotu sniegumu, rezultātus salīdzinot ar iepriekšējo modernāko risinājumu rezultātiem un vairākvalodu BERT modeļiem (mBERT). LVBERT modelis uzrādīja augstākus rezultātus vārdšķiru identificēšanas, nosaukto entītiju atpazīšanas un universālo atkarību parsēšanas uzdevumos.

Arī morfoloģiskās marķēšanas uzdevumā, eksistē veikti centieni automātiskas latviešu valodas marķēšanai. Viens no ievērojamajiem risinājumiem ir Latvijas Universitātes Matemātikas un informātikas institūta izstrādātais morfoloģiskās analīzes rīks, kas izstrādāts

programmēšanas valodā “Java” [1]. Rīka galvenais darbības princips balstīts uz to, ka latviešu valoda ir fleksīva, kas nozīmē, ka lielākā daļa vārdu sastāv no nemainīgas saknes un galotnes, kas norāda dažādas valodnieciskas pazīmes. Daudznozīmības aspekts risinājumā adresēts, apsverot vārda sintaktisko kontekstu. Morfoloģiskajā analīzē izmantota galotņu datu bāze, kura izmantota, lai ģenerētu iespējamās vārdu formas.

Arī Tildei eksistē savs morfoloģiskās analīzes rīks, kura tehniskā realizācija gan nav publiski pieejama, taču tas ir detalizēti aprakstīts [5]. Aprakstā iekļauts ieskats morfoloģiskā rīka organizācijas mehānismos. Rīks izmanto trīs (analīzes, sintēzes, visu-formu) pārveidotājus. Analīzes pārveidotājs pieņem tekstvienību kā ieeju un izejā nodrošina attiecīgās lemmas un aprakstu marķējumus. Sintēzes pārveidotājs pieņem lemmu un apraksta marķējumu un ģenerē atbilstošo tekstvienības locījumu. Visu-formu pārveidotājs pieņem lemmu un tā vārdšķiru un ģenerē visas iespējamās vārdformas ar to aprakstošajiem marķējumiem izejā. Šī rīka ātrdarbība tika salīdzināta ar tā priekšteča rīku. Iepriekšējā risinājuma pārbaudes uzdevuma izpildes laiks bija 7 minūtes un 25 sekundes, kamēr jaunā rīka ātrdarbība bija vien 27 sekundes.

Tāpat, 2015. gadā tika izstrādāts morfosintaktisks marķētājs, izmantojot vairāku klašu perceptronu. Modeļa darbība galvenokārt balstās uz dažādām vārdu pazīmēm, kā piemēram, kāds ir pašreizējais apskatāmais vārds, iepriekšējais vārds, nākamais vārds, pašreizējā vārda pēdējie burti utml. [6]. Risinājums sasniedza 98.29% precizitāti vārdšķiru noteikšanas uzdevumā un 94.33% morfoloģiskā marķējuma noteikšanas uzdevumā. Līdzīgi, arī 2020. gadā tika realizēts perceptrona risinājums morfoloģiskajai marķēšanai [7], taču šī risinājuma ietvaros tika izmantoti konteksta logi, kas galvenokārt risinājumā apsvēra katra marķējamā mērķa vārda apkārtējo *i* vārdu marķējumus (kur *i* ir iepriekš definēts loga izmērs). Izmantojot optimālo konfigurāciju, klasifikatora precizitāte bija aptuveni 89.03%.

Paikens 2016. realizēja neironu tīklu risinājumu morfoloģiskai marķēšanai [8]. Realizācijā tika izmantots viens divvirzienu LSTM slānis (bieži saukts arī par BiLSTM) un izejas slānis, kas nodrošināja klasifikāciju. Tekstvienību skaitliskai reprezentācijai tika izmantoti iegulto vērtību dati, kas izgūti no liela, nemarķēta korpusa. Risinājums sasniedza 97.8% vārdšķiras atpazīšanas precizitāti un 93.8% pilnā marķējuma precizitāti.

2. PIEEJAMĀS METODES

Lai realizētu automatizētu morfoloģisko marķēšanu, iespējams izmantot vairākas metodes. Šī literatūras apskata ietvaros, tiks veikts dažādu metožu apskats, to implementācijas priekšrocības un trūkumi, kā arī sasniegtie rezultāti eksperimentu izpildē. Tiks apskatīti gan modernākie risinājumi morfoloģiskās marķēšanas uzdevumā, gan citas iespējamās pieejas uzdevuma izpildei, kas neiekļauj mašīnmācīšanos.

Apskatot eksistējošus risinājumus un tos aprakstošās publikācijas, tiks nodrošināta informēta morfoloģiskās marķēšanas metodes izvēle šī darba ietvaros izvirzītā uzdevuma realizēšanai.

2.1. Modernākie risinājumi dabiskās valodas apstrādē

Lai noskaidrotu kādi ir modernākie risinājumi pasaulē dažādos dabiskās valodas apstrādes uzdevumos, ir noderīgi iepazīties ar *NLP-progress*[1] saturu. Tas ir tīmekļa resurss, kurā regulāri tiek atjaunināti apskati par sasniegumiem un labākajiem rezultātiem dabiskās valodas apstrādē dažādās valodās. Veicot navigāciju uz attīstību angļu valodā[2], iespējams apskatīt labākos rezultātus vārdšķiru marķēšanai angļu valodā. Lai gan tas ir tikai viens no apakšuzdevumiem konkrētā darba ietvaros, jo morfoloģiskai marķēšanai nepieciešamas vairākas lingvistiskās pazīmes kā tikai vārdšķira, tas var būt noderīgs informācijas avots, jo uzdevuma princips ir līdzīgs.

Izmantojot "Universal Dependencies" satvaru šķērs-valodu anotācijām, kurā iekļautas vairāk kā 60 valodu tekstuālās datu kopas, realizēti vairāki dziļās mašīnmācīšanās risinājumi. Labākais no tiem sasniedz 96.77% precizitāti. Modeļu testēšana realizēta, izmantojot 21 daudzresursu valodu. Tabulā 2.1 apkopoti 5 labākie risinājumi, balstoties uz *NLP-progress* pieejamo informāciju vārdšķiru atpazīšanas uzdevumā.

[1] https://github.com/sebastianruder/NLP-progress

[2] https://nlpprogress.com/english/part-of-speech_tagging.html

*2.1. tabula "***NLP-progress" modernāko risinājumu apkopojums**

Risinājums	**Vidējā precizitāte (* - testēta uz 17 valodām)**
Multilingual BERT and BPEmb [9]	96.77%
Adversarial Bi-LSTM [10]	96.65%
MultiBPEmb [9]	96.62%
Bi-LSTM [11]	96.40%
Joint Bi-LSTM [12]	95.55% *

Visu pētījumu ietvaros [9, 10, 11, 12] kādā mērā tikuši izmantoti un pārbaudīti divvirzienu LSTM.

Pētījumā, kura ietvaros tika izstrādāti divi risinājumi, kas sniedza augstāko precizitāti un trešo augstāko precizitāti vārdšķiras atpazīšanā [9], tika apskatīti vairāki tekstvienību apstrādes veidi. Tika konstatēts, ka vidēji labākais risinājums ir vairākvalodu BERT modeļa un BPEmb (baitu pāru iekodēšanas) apstrādes kombinācija, izmantojot LSTM pirms klasifikācijas. Starp 5 augstākās precizitātes risinājumiem ir divi risinājumi, kas realizēti apskatītā pētījuma ietvaros. Trešās augstākās precizitātes risinājumā tika izmantota viena - vairākvalodu BPEmb vārdnīca. Pētījuma autori arī min, ka tika novērots, ka pielāgojot BERT modeļus konkrētajam uzdevumam, atjaunojot to svarus visos tā slāņos, ir resursus patērējoša pieeja, taču sniedz labākus rezultātus kā pazīmju izgūšana (izmantojot tikai noteiktu skaitu BERT slāņu, tos nepielāgojot).

Darba izstrādes laikā otrās augstākās precizitātes sniedzošajam risinājumam [10] modeļa realizācijā tika izmantota pretendīvā apmācība. Tīkla arhitektūra tika sastādīta no simbola-līmeņa divvirziena LSTM slāņiem, kuriem tika padoti simbola-līmeņa reprezentācijas katram vārdam. Tīkla beiga stāvokļi tika apvienoti ar vārdu iegultajām vērtībām un tālāk padoti vēl vienam divvirzienu LSTM līmenim (vārda-līmeņa divvirzienu LSTM), lai apstrādātu visu teikumu. Tāpat, vārda-līmeņa divvirzienu LSTM izvades virkne tika padota CRF slānim, lai aprēķinātu nosacīto varbūtību mērķa marķējumu secībai. Risinājumā pretendīvā apmācība tika izmantota ar mērķi nodrošināt efektīvāku regularizāciju. Pirms modeļa apmācības, apmācības datos tika radīti gadījumi, kas ir pietuvināti oriģinālajai ievadei un ar tādu pašu sagaidāmo klasifikāciju, taču ar augstu varbūtību tikt nepareizi klasificētiem. Šāda veida pretendīvie gadījumi tika radīti ar nolūku, ieviešot mākslīgi radītus traucējumus ievaddatos.

Pēc līdzīga principa, arī jPTDP (*joint POS tagging and dependency parsing*) [12] modeļa realizācijā tika implementēta simbolu-balstīta vektoru reprezentācija katram vārdam, izmantojot divvirzienu LSTM. Simbolu-balstītās vektoru reprezentācijas tika konkatenātas ar vārda iegultās vērtības vektoru, rezultējoties ieejas vektorā, kas tika padots vēl vienā divvirzienu LSTM.

2.2. Citas metodes morfoloģiskās marķēšanas uzdevumā

Morfoloģiskās marķēšanas uzdevums ir ticis adresēts arī pirms straujā mašīnmācīšanās lietojuma pieauguma. Šīs apakšnodaļas ietvaros apskatītas citas metodes, ar to lietojumu piemēriem augsti lokāmās valodās.

2.2.1. Likumu balstīta morfoloģiska marķēšana

Uz likumiem balstīta marķēšana ietver noteiktu instrukciju vai likumu izmantošanu, lai secinātu ieejas vārda valodnieciskās pazīmes, kā arī, lai vārdiem piešķirtu atbilstošu marķējumu.

Uz likumiem balstīta marķiera ieviešanas uzbeku valodai ietvaros [13], M. Šaripovs (u.c.) realizēja *Python* valodas likumu balstītu risinājumu. Programmatūras realizācijā sākumā katra tekstvienība tiek uzmeklēta lemmu vārdnīcā. Ja šī lemma ir atrasta, vārds tiek attiecīgi klasificēts. Savukārt, gadījumā, ja vārdam piešķirama vairāk kā viena klase, tiek analizēti tekstvienības piedēkļi. Ja tekstvienībai nepiemīt piedēkļu vai klasifikācija ir neveiksmīga, tiek izmantota likumu kopa, kas klasifikāciju realizē, balstoties uz citiām konkrēto tekstvienību aptverošajām tekstvienībām. Lai gan risinājumā tiek ņemtas vērā arī apkārtējās tekstvienības, svarīgi atzīmēt, ka šī nav konteksta balstīta analīze, jo netiek ņemta vērā teikuma semantika, bet apkārtējo vārdu valodnieciskās pazīmes. Risinājums tika testēts, izmantojot valodu korpusus, kas iegūti no grāmatām, iedalītām pēc to tematiskajiem novirzieniem. Testa datu kopu veido 23'482 vārdi. Marķētājs sasniedza 89.78% vispārējo vidējo precizitāti, sfērās kā anatomija, sasniedzot 96.97% precizitāti.

Ņemot vērā, ka likumu-balstīti marķētāji darbojas pēc cieti-iekodētas likumu loģikas, to darbības princips ir skaidri pārredzams un intuitīvs, kas padara tos par potenciālu metodi marķēšanas uzdevumu izpildē, taču tiem ir arī ievērojami trūkumi. Augstas lokāmības valodu gadījumos, likumu kopums sasniedz augstu sarežģītības pakāpi un to programmātiska izstrāde var prasīt ievērojamu darba ieguldījumu. Tāpat, tiem ir augsta atkarība no realizācijai pieejamajām lemmu vārdnīcām. Arī darba autori piemin, ka anatomijas sfēras augstie rezultāti sasniegti, pateicoties bagātīgajai vārdnīcai, kas pieejama sfēras ietvaros un pārējo rezultātu uzlabošanai, nepieciešams paplašināt to vārdnīcu terminoloģiju.

2.2.2. Varbūtības modeļu pieejas morfoloģiska marķēšana

Varbūtības modeļi izmanto statistikas metodes, lai analizētu un ģenerētu datu virknes. Viena no izplatītākajām metodēm ir Apslēpto Markova modeļu (*HMM*) izmantošana. Līdzīgi kā mašīnmācīšanās pieejā - HMM var tikt apmācīti ar marķētu datu kopu. Atšķirībā no mašīnmācīšanās modeļiem, HMM paredz varbūtību vārdiem piederēt kādai noteiktai runas daļai, tā vietā, lai analizētu tekstu veiktu marķējumu piešķiršanu, balstoties uz ieejas teksta likumsakarībām. Balstoties uz HMM apmācībā iegūtajām varbūtībām, ir iespējams piešķirt marķējumus iepriekš neredzētam tekstam.

Morfoloģiskā marķētāja turku valodai [14] ieviešanā, darba autori izmantoja pirmās pakāpes Markova modeli ar viena-soļa vēsturi stāvokļu telpā. Viena-soļa vēstures stāvokļu telpā pielietojums norāda, ka morfoloģiskā marķējuma piešķiršanai katrai tekstvienībai, modelis ņem vērā tikai marķējumu, kas piešķirts iepriekšējai tektsvienībai, neņemot vērā iepriekšējās tekstvienības teikumā. Tāpat, marķētāja realizēšanai, autori nodefinēja *l-leksēmu* modeļu versijas. *L-leksēmas* norāda, cik tekstvienības pēdējie burti tiek ņemti vērā varbūtību aprēķinos. Skaitlis *l* ieņem vērtības no 1 līdz 7.

Modeļa apmācībai tika izmantots turku valodas korpuss, kurš sastāv no 45'000 teikumu un 655'720 tekstvienībām (no kurām 90'655 ir unikālas vērtības). Modeļa testēšanai tika izmantoti 100 teikumi un 1078 tekstvienības (no kurām 598 ir unikālas vērtības).

Risinājuma testēšanas rezultātā, pirmajā eksperimentālajā daļā, kurā tika testēts modelis bez viena-soļa vēstures stāvokļu telpā, tika iegūts 88.9% precizitātes rādītājs, izmantojot 5-leksēmas. Savukārt otrajā eksperimentālajā daļā, kurā tika testēts modelis ar viena-soļa vēsturi stāvokļu telpā, tika sasniegta 90.2% precizitāte, arī izmantojot 5-leksēmas.

3. IZVĒLĒTAIS RISINĀJUMS

Eksperimentālās daļas realizācijai nepieciešams pieņemt konkrētu tehnisko pieeju problēmas risināšanai. Ņemot vērā LSTM augošo popularitāti dabiskās valodas apstrādes jomā un to izraisošo augsto sniegumu, LSTM tīkla ieviešana ir šķietami pamatots lēmums.

Balstoties uz literatūras apskatu, secināms, ka LSTM augstā lietojamība dabiskās valodas apstrādes uzdevumos, sevišķi morfoloģiskajā marķēšanā, saistīta ar to spēju apstrādāt sekvenciālus datus, katrā apstrādes solī saglabājot un utilizējot informāciju no iepriekšējiem soļiem. Ņemot vērā, ka teksts ir vārdu vai simbolu sekvenciālas virknes, LSTM ir ievērojams kandidāts morfoloģiskās marķēšanas nodrošināšanai. Sevišķi piemērota šim uzdevumam varētu būt divvirzienu LSTM realizācija. Divvirzienu LSTM sastāv no priekšēji orientēta LSTM, kas ievadi apstrādā no kreisās uz labo pusi, un atpakaļgaitā orientēta LSTM, kas to apstrādā no labās uz kreiso pusi [15]. Ar šādu pieeju iespējams izgūt svarīgu informāciju ievadē katra vārda gan pagātnes, gan nākotnes kontekstos.

Neironu tīkli nespēj veiksmīgi darboties un tikt apmācīti ar neapstrādātu tekstu tā reālajā reprezentācijā, jo neironu tīklu realizācijas ir balstītas uz matemātisku operāciju realizēšanu, izmantojot tenzorus. Standarta pieeja datu priekšapstrādē ir ieejas teksta pārvēršana skaitliskās vērtībās. Viena no visveiksmīgāk pielietojamajām skaitliskajām reprezentācijām ir vektora forma.

Divas no izplatītākajām pieejām teksta vektorizācijā ir Word2Vec un BERT modeļi. Word2Vec modeļi galvenokārt analizē katra vārda apkārtējo kontekstu. Savukārt BERT modeļi apsver visa pieejamā dokumenta kontekstu. Citiem vārdiem, Word2Vec modelis vairāknozīmju vārdam piešķirs vienu vektoru, lai gan tas apmācības laikā pasniegts vairākos, dažādos kontekstos. Turpretī, BERT modelis vairāknozīmju vārdam piešķirs vairākus vektorus, kur katrs vektors būs pietuvinātāks tā aktuālajam kontekstam.

Ņemot vērā to, ka viena no morfoloģiskās marķēšanas galvenajām problēmām ir lingvistisko pazīmju atpazīšana vārdiem, kuriem iespējamas vairākas nozīmes, šī darba ietvaros ir svarīgi saglabāt informāciju par katra vārda iespējamajiem kontekstiem. Balstoties uz šo iemeslu, teksta skaitliskai reprezentācijai tika nolemts izmantot BERT modeļus.

Balstoties uz apskatīto literatūru, secināms, ka BERT modeļu efektivitāte palielinās, pielāgojot tos konkrētam uzdevuma ietvaram. Šī parādība tiks empīriski pārbaudīta arī latviešu valodas apstrādē šī darba ietvaros. Tiks pārbaudīta arī risinājuma efektivitātes atkarība no tekstvienību pārveidošanas tikai mazos burtus saturošās tekstvienībās. Tiek pieņemts, ka šāda

veida priekšapstrādei ir pienesums BERT modeļu efektivitātē, jo tiek mazināta ārpus-vārdu-krājuma problēmas ietekme.

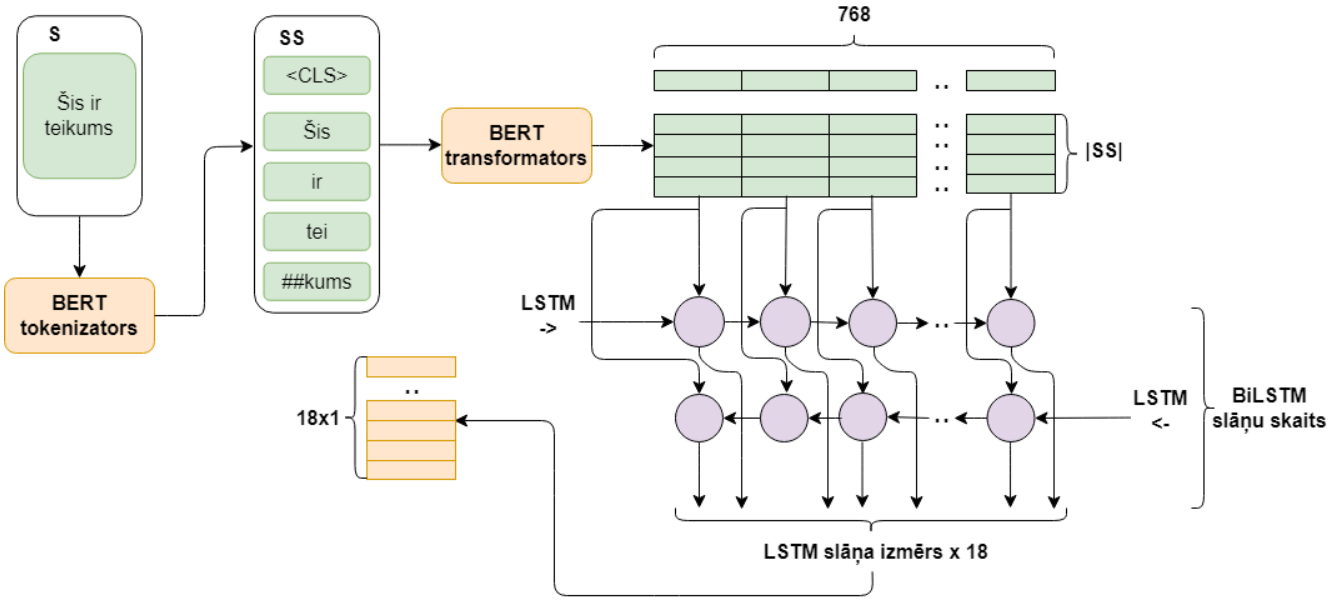
Tāpat, lai izvairītos no modeļa pārpielāgošanas apmācības laikā, tiks izmantoti atbiruma slāņi. Tehniskā tīkla arhitektūras implementācija sīkāk aprakstīta turpmākajās nodaļās.

Lai nodrošinātu tehnisko implementāciju, tika identificētas visas iespējamās latviešu valodas vārdu kopas valodnieciskās pazīmes. Ņemot vērā, ka tika konstatētas 18 unikālas valodnieciskās pazīmes un darba ietvaros katra pazīme uzskatāma kā klase, kurai piešķirams vārds, tīkla izvadē tiks realizēti 18 lineāri klasifikācijas slāņi. Tāpat, katrai klasei iespējamas vairākas klasifikācijas vērtības.

Apsvērto valodniecisko pazīmju sarakstā ietilpst:

- Vārdšķira;
- Pieturzīmes tips;
- Skaitlis;
- Rekcija;
- Locījums;
- Dzimte;
- Skaitlis 2;
- Lietvārda tips;
- Laiks;
- Persona;
- Darbības vārda tips;
- Izteiksme;
- Lokāmība;
- Noteiktība;
- Saīsinājuma tips;
- Apstākļa vārda tips;
- Vietniekvārda tips;
- Reziduāļa tips.

Attēlā 3.1 shematiski parādīta izvēlētā arhitektūra augsta līmeņa diagrammā. Diagrammā attēlots klasifikācijas process - ieejas teikums tiek sadalīts tekstvienībās/apakštekstvienībās, izmantojot BERT tokenizatoru. Pēc tā, tokenizētās vērtības tiek padotas BERT transformatora modelim, kas nodrošina izvadi 768 vienību izmērā. BERT izvades dati tiek nodoti LSTM tīklam. Diagrammā parādīts viens divvirzienu slānis, taču realitātē slāņu skaits var mainīties, atkarībā no arhitektūras konfigurācijas. LSTM slāņu izvade tiek konkatenēta un nodota 18 lineārajiem klasifikācijas slāņiem. Katram slānim eksistē vairākas iespējamās vērtības, kas var tikt piešķirtas klasifikācijā, atkarībā no konkrētās valodnieciskās pazīmes.



3.1. att. **Risinājuma arhitektūra**

Konteksta analīzei tiks izmantoti divi latviešu valodu atbalstoši BERT modeļi - LVBERT[3] modelis un LitLat BERT[4] modelis. Abu modeļu veiktspēja konkrētā uzdevuma ietvaros tiks empīriski salīdzinātas dažādos tehnisko implementāciju iestatījumos.

Abi BERT modeļi sastāv no 12 slāņiem un 768 apslēptajām vienībām. Modeļu galvenā atšķirība ir to vārdnīcas (jeb vārdu krājumi). LitLat BERT vārdnīca sastāv no aptuveni 84 tūkstošiem vienību, kamēr LVBERT vārdnīca sastāv no aptuveni 32 tūkstošiem vienību [16].

Realizācija tiks izmantots BERT un papildus LSTM slāņu risinājums, balstoties uz pieņēmumu, ka vēl viens divvirzienu LSTM slānis nodrošinās papildus svarīgu iezīmju izgūšanu, kas varētu būt zaudēta, izmantojot tikai BERT un klasifikācijas slāņus. Tāpat, plānots darba ietvaros šo pieņēmumu pārbaudīt empīriski, veicot eksperimentus.

[3] https://huggingface.co/AiLab-IMCS-UL/lvbert
[4] https://huggingface.co/EMBEDDIA/litlat-bert

Lai realizētu klasifikāciju, katrai valodnieciskajai pazīmei tika identificētas attiecīgās vērtības, kuras iespējams piešķirt katrai tekstvienībai. Tabulā 3.1 apkopotas visas iespējamās vērtības.

Darba tehnisko risinājumu iespējams apskatīt autora *github* saitē[5].

3.1. tabula **Valodniecisko pazīmju iespējamās vērtības klasifikācijai**

Valodnieciskā pazīme	**Pazīmes iespējamās vērtības**		**Valodnieciskā pazīme**	**Pazīmes iespējamās vērtības**
Vārdšķira	Lietvārds, darbības vārds, pieturzīme, vietniekvārds, īpašības vārds, saiklis, apstākļa vārds, prievārds, reziduālis, partikuila, saīsinājums, skaitļa vārds		Darbības vārda tips	Patstāvīgs darbības vārds, palīgverbs ‘būt’, modāls, palīgverbi ‘tikt’ un ‘tapt’, saitiņas ‘tikt’ un ‘tapt’, fāzes, izpausmes veida
Pieturzīmes tips	Komats, punkts, pēdiņa, domuzīme, iekava, kols, cita		Persona	1, 2, 3, nepiemīt
Skaitlis	Vienskaitlis, daudzskaitlis, nepiemīt		Izteiksme	Īstenības, divdabis, nenoteiksme, vēlējuma, vajadzības, pavēles, atstāstījuma
Rekcija	Akuzatīvs, datīvs, ģenitīvs, nepiemīt		Lokāmība	Lokāms, nelokāms, daļēji lokāms
Locījums	Nominatīvs, ģenitīvs, akuzatīvs, datīvs, lokatīvs, vokatīvs, nepiemīt		Noteiktība	Nenoteiktā, noteiktā, nepiemīt
Dzimte	Vīriešu, sieviešu, nepiemīt		Saīsinājuma tips	Īpašvārds, sugasvārds, diskursa iezīmētāji, apstāklis, īpašības vārds, verbāls
Skaitlis 2	Daudzskaitlinieks, vienskaitlinieks		Apstākļa vārda tips	Veida, laika, mēra, vietas, cēloņa/nolūka
Lietvārda tips	Sugas vārds, īpašvārds		Vietniekvārda tips	Norādāmais, personas, attieksmes, nenoteiktais, noteiktais, piederības, atgriezeniskais, jautājamais
Laiks	Pagātne, tagadne, nākotne, nepiemīt		Reziduāļa tips	Kārtas skaitlis ciparіem, skaitlis ciparіem, vārds svešvalodā, cits, URI

[5] https://github.com/atreimanis/BERTPosTaggerLV/tree/main

4. EKSPERIMENTĀLAIS IETVARS

4.1. Datu kopa

Modeļa apmācībai un eksperimentālās daļas realizācijai, izmantots *Latvian Treebank* latviešu valodas korpuss [17]. Šī datu kopa izmantota arī citos šīs problēmas risinājumos, kā piemēram, dziļās mašīnmācīšanās metožu izmantošanā morfoloģiskai marķēšanai [8], kas nodrošina atsauces punktu jēgpilnai rezultātu salīdzināšanai.

Datu kopa tika nodalīta trijās apakškopās, atšķirīgu datu izmantošanai modeļa apmācībai, validācijai un testēšanai. Apmācības datu kopa sastāvēja no 16’594 teikumiem un 276’204 tekstvienībām, validācijas - 2’131 teikuma un 35’637 tekstvienībām, testēšanas - 2’412 teikumiem un 37’347 tekstvienībām.

Sākotnēji, datu kopa bija pieejama teksta datnes formātā, kurā teikumi nodalīti ar speciāli atvēlētiem teikuma sākuma un beigu indikatoru simboliem, un katra teikuma vārds pierakstīts jaunā rindā, katrai rindai saturot ieejas vārdu, tā marķējumu un vārda pamatformu. Piemēram, teikuma vārds “viņi”, datnē pieejams formātā:

viņi pp3mpnnviņš m-s133-p5s1w2

Priekšapstrādes ietvaros, teksta datne tika pārvērsta JSON formāta datnē, pārveidojot katru teikumu par masīvu un katru tekstvienību par JSON objektu, kas satur ieejas vārdu, marķējumu, pamatformu un marķējumu veidojošās valodnieciskās pazīmes. Formāta pārveidošanai izmantots latviešu valodas morfoloģiskās analīzes rīks. Iepriekš minētais vārda “viņi” piemērs pēc reprezentācijas pārveidojuma pieejams formātā:

{"gold_tag_simple":"pp3mpn_","wordform":"viņi","options":["pp3mpn_"],"gold_tag":"pp3mpnn","gold_lemma":"viņš","gold_attributes":{"Skaitlis":"Daudzskaitlis","Persona":"3","Vārdšķira":"Vietniekvārds","Vietniekvārda tips":"Personas","Locījums":"Nominatīvs","Dzimte":"Vīriešu"}}

Tabulā 4.1 apkopota informācija par apmācības datu kopas vārdšķiras vērtību sadalījuma. Tabulā apkopots vārdu skaits un procentuālā attiecība pret kopējo vienību skaitu, atkarībā no vārdšķiras tipa.

4.1. tabula **Vārdu sadalījums pēc to vārdšķirām**

Vārdšķira	**Vienību skaits**	**Procentuālā attiecība**
Lietvārds	84’860	30.7%
Darbības vārds	49’265	17.8%
Pieturzīme	47’973	17.4%
Vietniekvārds	21’322	7.7%
Saiklis	17’460	6.3%
Apstākļa vārds	15’037	5.4%
Īpašības vārds	13’351	4.8%
Prievārds	12’210	4.4%
Partikula	5’644	2%
Reziduālis	4’647	1.7%
Skaitļa vārds	2’536	0.9%
Saīsinājums	1’752	0.6%
Izsauksmes vārds	147	0.1%

4.2. Eksperimentālā vide

Risinājuma praktiskai implementācijai, izmantota programmēšanas valoda Python. Mašīnmācīšanās modeļa realizācijai izmantota satvara PyTorch 2.0.1 versija. BERT modeļu un tokenizatoru ielādei izmantota “Huggingface” bibliotēka *transformers*.

Pirmkoda implementācijai izmantota tīmekļa-balstīta skaitļošanas vide Jupyter Notebook. Lai nodrošinātu datu ielādi pirmkoda mainīgajos un to priekšapstrādi, tika izmantota bibliotēkas Torchtext 0.6.0 versija.

Ņemot vērā, ka mašīnmācīšanās realizēšanai, ir nepieciešams apstrādāt lielus datu apjomus un veikt lielu skaitu matemātisku operāciju, tiek patērēti lieli skaitļošanas resursi. Apmācības paātrināšanai tika iespējota CUDA lietojumprogrammas saskarne, kas *PyTorch*

satvara implementācijai ļauj izmantot darbstacijas grafisko karti, kas savukārt nodrošina paralēlo apstrādi, kuras rezultātā iespējams veikt vairākas skaitļošanas operācijas vienlaicīgi.

Eksperimentālās daļas ietvaros tika izmantotas divas darbstacijas - autora lokālā darbstacija un "Google Cloud Platform" piedāvātā pakalpojuma virtuālā darbstacija. Kā virtuālās darbstacijas grafiskā karte tika izmantota NVIDIA T4 grafiskā karte. Kā lokālās darbstacijas grafiskā karte tika izmantota GeForce RTX 3070 Ti. Savienojums ar virtuālo darbstaciju tika nodrošināts izmantojot *ssh* savienojumu.

4.3. Datu priekšapstrāde

Datu priekšapstrādē netika mainīts ieejas teikumu tekstuālais saturs, jo tiek pieņemts, ka datu kopas ir iepriekš "attīrītas".

Ieejas datu iekodēšanai un pēc tā iegulto vērtību (skaitlisko vērtību vektortelpā) iegūšanai tika izmantoti BERT modeļu tokenizatori. Tokenizatori, izmantojot tiem pieejamo vārdu krājumu, katru vārdu tekstā aizstāj ar unikālu iekodējuma vērtību. Tokenizatori adresē "ārpus-vārdnīcas" problēmu, kurā kāda no tekstvienībām nav pieejama tokenizatora vārdu krājumā, aizstājot tekstvienības ar apakštekstvienībām. Apakšteksvienības parasti tiek atzīmētas ar speciāli atvēlētiem simboliem. Piemēram, LVBERT atzīmēšanu nodrošina ar diviem restes simboliem, indicējot, ka tās ir daļa no lielāka vārda. Piemēram, teikumu "Brīdi turēja to rokās.", LVBERT tokenizators pārvērstu virknē, kas sastāv no sekojošām tekstvienībām:

"Brīdi", "tur", "##ēja", "to", "rokās", ". ".

Turpretī, LitLat BERT ar apakšsvītras simboliem atzīmē pirmo apakštekstvienību, indicējot, ka apakštekstvienības, kas nesākas ar šo atzīmi ir daļa lielākas tekstvienības. Iepriekš apskatīto piemēru LitLat BERT pārvērš:

"_Br", "īdi", "_tur", "ēja" "_to", "_rokās", ".".

Tāpat, katrs teikums tiek pārveidots vārdnīcās datu struktūrā, kas sastāv no 20 atslēgu-vērtību pāriem. Katra atslēga apzīmē visu 18 valodniecisko pazīmju nosaukumus. Katrai valodnieciskās pazīmes atslēgai pakārtotās vērtības ir saraksti, kuru saturā saglabātas atbilstošajās pozīcijās katra teikuma vārda atbilstošās valodnieciskās pazīmes vērtības. Piemēram, vārdnīca, iepriekš jau apskatītā teikuma "Brīdi turēja to rokās." LVBERT sadalījuma gadījumā, saturēs ierakstu, kurā atslēgas "Vārdšķira" saraksta saturs būs:

['Lietvārds', 'Darbības vārds', 'Vietniekvārds', 'Lietvārds', 'Pieturzīme'].

Deviņpadsmitā atslēga apzīmē ieejas teikuma saturu.

Ņemot vērā, ka tika izmantota arī tekstvienību sadalīšana apakštekstvienībās, tabulā 4.2 apskatāma katra apakštekstvienība un tai piešķirtā valodnieciskās pazīmes "vārdšķira" vērtība, atbilstoši priekšapstrādei.

4.2. tabula **Apakštekstvienību "vārdšķira" valodnieciskās pazīmes vērtības teikuma piemēram**

LitLat BERT izveidotā apakštekstvienība	**"Vārdšķira" saraksta apakštekstvienībai atbilstošā vērtība**
_Br	Lietvārds
īdi	Lietvārds
_tur	Darbības vārds
ēja	Darbības vārds
_to	Vietniekvārds
_rokās	Lietvārds
.	Pieturzīme

Redzams, ka darbības vārdiem "brīdi" un "turēja" abām vārdu apakštekstvienībām piešķirtas tās pašas vārdšķiru vērtības. Lai nodrošinātu testēšanas konsekvenci, tika realizēts arī kartēšanas saraksts katrā datu piemērā. Kartēšanas saraksts norāda to, kurai teikuma sākotnējai tekstvienībai pieder konkrētā apakštekstvienība. Apskatītajam piemēram kartēšanas saturs būtu: [0, 0, 1, 1, 2, 3, 4]. Tātad, skaitīšanu sākot ar 0, apakštekstvienības "_Br" un "īdi" atbilst tekstvienībai ar numuru 0, kas ir vārds "Brīdi", un apakštekstvienības "_tur" un "ēja" atbilst tekstvienībai ar numuru 1, kas ir vārds "turēja".

Pēc datu ielādes, katrai no trijām (apmācības, validācijas un testēšanas) datu kopām tika izveidoti iteratoru objekti, kuri iespējo datu padošanu dziļās mašīnmācīšanās modelim partijās, kas savukārt ir iepriekš konfigurētos lielumos.

4.4. Eksperimentālie iestatījumi un apmācība

Lai nodrošinātu pilnvērtīgu modeļa apmācību un empīriski atrastu labāko iespējamo risinājumu, tika realizēta ablā sākotnēji tika nodrošināta ablācijas pētīšana (no angļu valodas - *ablation study*). Sākotnēji tika identificēti risinājuma hiperparametri, kas varētu ietekmēt apmācības gaitu un modeļa testēšanas rezultātus. Modeļu apmācība, validācija un testēšana

notika uz vienām un tām pašām datu kopām visu konfigurāciju ietvaros, lai nodrošinātu rezultātu objektivitāti.

Modeļu apmācība tika nodrošināta iteratīvi jeb izmantojot epohas. Katrā nākamajā epohā modelim tika padoti tie paši apmācības dati, kas tika izmantoti apmācībai iepriekšējā epohā. Lai mazinātu modeļa tieksmi pārpielāgoties, tika salīdzinātas katras tekošā un labākās epohas zuduma funkcijas vērtība. Zuduma funkcija aprēķināta, izmantojot validācijas datu apakškopu. Gadījumā, ja zuduma funkcijas vērtība tekošajā epohā attiecībā pret labāko epohu ir palielinājusies - šis modeļa stāvoklis netiktu saglabāts, kā rezultātā tas netiktu izmantots testēšanas fāzē.

Lai radītu priekšstatu par apmācībā patērētajiem laika resursiem, izmantojot *Python datetime* bibliotēkas laika funkcijas, tika mērīts katras epohas ilgums.

Visās modeļa realizācijās tika izmantota šķērsentropijas zuduma funkcija. Šķērsentropijas zuduma funkcija ir labi piemērojama morfoloģiskās marķēšanas uzdevumos, jo tā izvērtē, cik lielā mērā neironu tīkla modeļa paredzētā varbūtības izkliede atbilst patiesajai valodniecisko pazīmju izkliedei.

Tabulā 4.3 apkopoti dažādi hiperparametri, kas izmantoti un mainīti katra eksperimenta ietvaros. Kolonnu vērtības atspoguļo risinājuma pirmkodā izmantotās vērtības (datu tipu reprezentācija un vērtības saglabātas). Tabulas kolonnu paskaidrojumi:

- **Nr**. - eksperimentālā iestatījuma kārtas numurs;
- **BERT** - izmantotais BERT modelis un tokenizators;
- **L** - ievades teksta pārveidošana tikai mazo burtu rakstzīmēs;
- **HS** - slēptā slāņa neironu skaits;
- **NL** - LSTM slāņu skaits tīkla arhitektūrā;
- **BS** - datu partijas izmērs;
- **DO** - atbiruma slāņu attiecība.

4.3. tabula **Eksperimentālās konfigurācijas**

Nr.	**BERT**	**L**	**HS**	**NL**	**BS**	**DO**
1.	LVBERT	False	512	1	16	0.25
2.	LVBERT	False	1024	1	16	0.25
3.	LVBERT	False	512	2	16	0.25
4.	LVBERT	False	256	2	16	0.25
5.	LVBERT	False	512	1	16	0.5
6.	LVBERT	False	256	3	16	0.25
7.	LVBERT	True	512	1	16	0.25
8.	LVBERT	True	1024	1	16	0.25
9.	LitLat	True	512	1	16	0.25
10.	LitLat	True	768	1	16	0.25
11.	LitLat	True	256	2	16	0.25
12.	LitLat	True	512	1	32	0.25
13.	LitLat	True	1024	1	16	0.25
14.	LitLat	True	1024	1	32	0.25
15.	LitLat	False	512	1	16	0.25
16.	LVBERT	True	1024	1	32	0.25

Papildus minētajām eksperimentālajām konfigurācijām un to testēšanai, eksperimentālās daļas ietvaros, izmantojot labākos rezultātos uzrādošās konfigurācijas katram BERT modelim, tika veikti šādi eksperimenti:

- modeļa apmācība, priekšapstrādē nepārveidojot tekstvienības apakštekstvienībās;
- modeļa apmācība un testēšana, izmantojot tikai BERT izejas datus un lineāros izejas slāņus (bez LSTM slāņiem);
- modeļa apmācība un testēšana, izmantojot vienvirziena LSTM slāņus;
- modeļa apmācība un testēšana, "iesaldējot" BERT modeļa svarus.

5. REZULTĀTI

5.1. Eksperimentālo konfigurāciju rezultāti

Tabulā 5.1 apkopoti risinājuma testēšanas rezultāti, izmantojot tabulā 4.1 definētās konfigurācijas.

Ņemot vērā, ka šī eksperimenta ietvaros tekstvienības tika sadalītas apakštekstvienībās, tika definēta precizitātes novērtēšanas metode, kas ņem vērā šo priekšapstrādi. Par paredzēto marķējumu tekstvienībai tika izvēlēts marķējums pirmajai apakštekstvienībai. Tas nozīmē, ka tekstvienībai, kas priekšapstrādes rezultātā tika pārveidota trijās apakštekstvienībās - pirmais no trim apakštekstvienību marķējumiem tika uzskatīts kā visu sākotnējo tekstvienību reprezentējošs marķējums.

5.1. tabula **Eksperimentālo rezultātu apkopojums ar sadalīšanu apakštekstvienībās**

Konfigu-rācijas nr.	**Vārdšķiras precizitāte (%)**	**Morfoloģiskā marķējuma precizitāte (%)**	**Vidējais apmācības epohas ilgums**	**Labāko rezultātu epoha**
1.	97.37	90.55	4m 21s	14
2.	97.34	90.44	3m 40s	11
3.	92.28	89.27	**2m 13s**	17
4.	97.07	89.27	3m 23s	19
5.	97.29	90.04	3m 7s	10
6.	32.64	0	3m 9s	19
7.	97.70	90.35	3m 5s	14
8.	97.62	90.38	3m 19s	11
9.	98.52	92.36	3m 25s	18
10.	48.01	5.86	3m 23s	7
11.	35.22	0.47	3m 17s	17
12.	98.53	92.71	3m 22s	19
13.	47.59	4.61	3m 10s	7
14.	**98.59**	**93.12**	5m 12s*	24
15.	53.16	3.84	5m 28s*	1
16.	**97.74**	**91.17**	5m 21s*	18

Tika novērots, ka LitLat BERT modeļa optimālākā arhitektūra uzrādīja par 0.85% augstāku rādītāju vārdšķiras noteikšanas precizitātē un par 1.95% augstāku precizitātes rādītāju morfoloģiskā marķējuma noteikšanas precizitātē, salīdzinot ar optimālāko LVBERT arhitektūru.

Tāpat, rezultātos redzams, ka LitLat BERT modeļa gadījumā, risinājums uzrāda krietni sliktākus rezultātus vārdšķiras atpazīšanā un morfoloģiskā marķējuma atpazīšanā, palielinot papildus LSTM slāņu skaitu.

Tika novērotas arī konfigurācijas, kurās ir ievērojami sliktāki rezultāti, lai gan arhitektūra netika būtiski mainīta. Piemēram, salīdzinot konfigurācijas nr. 9 un 10, novērots ievērojams kritums precizitātes rādītājos, lai gan to vienīgā atšķirība ir LSTM neironu skaits – attiecīgi 512 un 768. Šajos gadījumos tika veikta cilvēka kļūdu kontrole, atkārtojot eksperimentus, izmantojot pirmkodu, kas izmantots citos, augstu precizitāti uzrādošos eksperimentos, un atbilstoši nomainot slāņa konfigurāciju. Tika konstatēta rezultātu atkārtojamība. Šī darba ietvaros netika padziļināti pētītas likumsakarības starp zemo rezultātu gadījumiem un to atbilstošajām konfigurācijām ar mērķi atklāt zemo rezultātu cēloni.

* - modeļi tika apmācīti, izmantojot *Google Cloud Platform* pakalpojumus, tādēļ tie var nebūt salīdzināmi ar citiem rezultātiem.

5.2. Rezultāti bez dalīšanas apakštekstvienībās

Tabulā 5.2 apkopoti labākie rezultāti eksperimentiem, kuros netika izmantota tekstvienību dalīšanas apakštekstvienībās priekšapstrādes metode. Kā labākās eksperimentālās konfigurācijas katram BERT modelim tika konstatētas 16. konfigurācija LVBERT modelim un 14. konfigurācija - LitLat modelim.

5.2. tabula **Eksperimentālo rezultātu apkopojums, neizmantojot dalīšanu apakštekstvienībās**

BERT modelis	**Vārdšķiras precizitāte (%)**	**Morfoloģiskā marķējuma precizitāte (%)**	**Vidējais apmācības epohas ilgums**	**Labāko rezultātu epoha**
LVBERT	92.51	79.56	2m 51s	4
LitLat	72.75	42.28	3m 2s	8

Eksperimenta ietvaros, vārdi netika sadalīti apakštekstvienībās, līdz ar to palielinājās BERT tokenizatoriem nezināmo jeb ārpus-vārdu-krājuma tekstvienību skaits.

Novērojams, ka implementācijas gūst ievērojamus uzlabojumus, izmantojot sadalīšanu apakštekstvienībās. LVBERT modelim novērota vārdšķiras klasificēšanas precizitātes samazināšās par 5.23% un morfoloģiskā marķējuma precizitātes samazināšanās par 11.61%, neizmantojot dalīšanu apakštekstvienībās. Tāpat novērots, ka modeļi ievērojami ātrāk konverģē (samazināšanās kopējā epohu skaitā - par 14 epohām LVBERT modelim un 16 epohām - LitLat BERT modelim).

5.3. Rezultāti bez LSTM slāņiem

Tabulā 5.3 apkopoti rezultāti eksperimentiem, neizmantojot LSTM slāņus. Klasifikācijā tika izmantoti BERT izejas dati, kas nodoti lineārajiem klasifikācijas slāņiem. Eksperimentos tika izmantotas apakšnodaļā 5.1 augstāko rezultātu sniedzošās konfigurācijas katram BERT modelim (konfigurācija nr. 14 LVBERT modelim un konfigurācija nr. 16 - LitLat modelim). Ņemot vērā, ka konfigurāciju hiperparametri **HS** (slēpto slāņu izmēri) un **NL** (slēpto slāņu skaits) attiecās tikai uz BiLSTM uzstādījumiem - šī eksperimenta ietvaros tie tika ignorēti.

5.3. tabula **Eksperimentālo rezultātu apkopojums, neizmantojot LSTM**

BERT modelis	**Vārdšķiras precizitāte (%)**	**Morfoloģiskā marķējuma precizitāte (%)**	**Vidējais apmācības epohas ilgums**	**Labāko rezultātu epoha**
LVBERT	97.57	90.74	5m 8s*	30
LitLat	98.41	92.80	5m 30s*	47

LitLat modeļa gadījumā tika novērota vārdšķiras rādītāja samazināšanās par **0.18%** un marķējuma precizitātes samazināšanās par **0.32%**. LVBERT modeļa gadījumā, vārdšķiru precizitāte samazinājās par **0.17%** un marķējumu precizitāte samazinājās par **0.43%**.

Tāpat novērots pieaugums nepieciešamo epohu skaitā - papildus **12** epohas LVBERT modeļa risinājumam un **23** epohas - LitLat BERT risinājumam.

* - modeļi tika apmācīti, izmantojot *Google Cloud Platform* pakalpojumus, tādēļ tie var nebūt salīdzināmi ar citiem rezultātiem.

5.4. Rezultāti, iesaldējot BERT svarus

Tabulā 5.4 apkopoti rezultāti eksperimentiem, iesaldējot LSTM slāņus. Standarta eksperimentālajā iestatījumā, arī BERT modeļu svari tika atjaunināti apmācības procesā. Šī eksperimenta ietvaros, visi BERT svari, izņemot klasifikācijas slāņu svari, tika iesaldēti, un tiek izmantoti tikai iepriekš apmācītie svari. Gluži kā apakšnodaļā 5.3, eksperimenta ietvaros tika apskatīti rezultāti labākos rezultātus uzrādošajām konfigurācijām.

5.4. tabula **Eksperimentālo rezultātu apkopojums, iesaldējot BERT svarus**

BERT modelis	**Vārdšķiras precizitāte (%)**	**Morfoloģiskā marķējuma precizitāte (%)**	**Vidējais apmācības epohas ilgums**	**Labāko rezultātu epoha**
LVBERT	95.68%	82.08%	3m 7s*	105**
LitLat	94.01	74.56%	3m 14s*	38**

Veicot testēšanu, tika konstatēts, ka LVBERT modeļa risinājums ar iesaldētiem visiem, izņemot klasifikācijas, slāņiem vārdšķiras noteikšanā uzrādīja par 2.06% zemāku sniegumu un visa morfoloģiskā marķējuma noteikšanā - par 9.09% zemāku sniegumu. LitLat modeļa risinājumā rādītāji samazinājās par attiecīgi 4.58% un 18.56%. Tāpat, tika konstatēts, ka vidējais apmācības epohas ilgums abiem risinājumiem bija par aptuveni 2 minūtēm īsāks, taču nepieciešamais epohu skaits, izmantojot to pašu apmācības tempu, bija krietni lielāks.

* - modeļi tika apmācīti, izmantojot *Google Cloud Platform* pakalpojumus, tādēļ tie var nebūt salīdzināmi ar citiem rezultātiem.

** - Eksperimenta laikā pēc pirmajām 20 epohām tika konstatēts, ka modeļi ievērojami lēnāk konverģē, tādēļ laika resursu taupīšanas nolūkos tika palielināts modeļu apmācības temps.

5.5. Rezultātu salīdzinājums ar citiem risinājumiem

Rezultātu salīdzināšanā tiks ņemta vērā tikai labākos rezultātus uzrādošā risinājuma arhitektūra (14. eksperimentālā konfigurācija).

Tabulā 5.5 apkopots salīdzinājums ar citiem risinājumiem morfoloģiskās marķēšanas uzdevumā.

5.5. tabula **Risinājumu rezultātu salīdzinājums**

Risinājums	**Vārdšķiras precizitāte (%)**	**Morfoloģiskā marķējuma precizitāte (%)**
Ņikiforovs [5]	98.29	**94.33**
Voļska [6]	-	89.03
Paikens [8]	97.8	93.80
Paikens [1]	98.46	93.30
BERT + BiLSTM	**98.59**	93.12

Svarīgi atzīmēt, ka lai gan datu resurss visiem risinājumiem ir tas pats, resurss tiek pastāvīgi mainīts un rezultāti var nebūt maksimāli objektīvi. LVTagger risinājums [1] objektīvākas salīdzināšanas labad tika apmācīts un testēts, izmantojot šī darba ietvaros izmantotos datus. Darbā izstrādātais risinājums vārdšķiras precizitātē uzrādījis labāko sniegumu, bet marķējuma precizitātē par **1.21%** zemāku rezultātu, salīdzinot ar augstāko marķējuma rezultātu.

5.6. Rezultātu diskusijas

Pētījuma laikā, izmantojot empīrisko validāciju, tika secināts, ka optimālākās hiperparametru konfigurācijas ir 14. (LitLat konfigurācija) un 16. (LVBERT konfigurācija) katram no BERT izmantotajiem modeļiem, izmantojot sadalīšanu apakštekstvienībās. LitLat modeļa risinājums sniedza augstākos precizitātes rādījumus - **98.59%** vārdšķiras atpazīšana un **93.12%** - morfoloģiskā marķējuma atpazīšana. Optimālākā LVBERT konfigurācija sasniedza **98.41%** vārdšķiras atpazīšanas precizitāti un **92.80%** morfoloģiskā marķējuma precizitāti.

Tika novērots, ka modeļu sniegumi samazinājās pēc LSTM apslēpto slāņu skaita palielināšanas, bet uzlabojās, saglabājot 1 apslēpto slāni, bet palielinot tā neironu skaitu. Ievērojami sliktākais rezultāts tika novērots, izmantojot 3 divvirzienu LSTM slāņus. Lai gan eksperimentālajā iestatījumā netika izmantota tekstvienību pārveidošana tikai mazo burtu vienībās - rezultāts bija ievērojami zemāks nekā līdzīgam iestatījumam, kurā izmantots 1 divvirzienu LSTM.

Veicot konfigurāciju eksperimentālo validāciju, neizmantojot tekstvienību sadalīšanu apakštekstvienībās, tika novērota veiktspējas samazināšanās abiem risinājumiem. LitLat BERT modeļa risinājuma precizitāte samazinājās līdz **72.75%** precizitātei vārdšķiras noteikšanā un **42.28%** precizitātei marķējuma noteikšanā. Lai gan ne tik ievērojami, bet arī LVBERT risinājuma precizitātes nokritās attiecīgi uz **92.51%** un **79.56%**. Tika secināts, ka modeļi gūst ievērojamus ieguvumus veiktspējā, sevišķi pilnā marķējuma paredzēšanā, izmantojot sadalīšanu apakštekstvienībās. Šī parādība saistīta ar to, ka liela daļa vārdformu nav pieejamas BERT vārdnīcās. Līdz ar to, liela daļa ieejas vārdu tiek aizstāti ar "nezināmo" simboliem, kā rezultātā priekšapstrādes procesā zūd liels daudzums nozīmīgas informācijas.

Divvirzienu LSTM slāņa pienesuma arhitektūrā pārbaudei - tika veikts eksperiments, kurā LSTM slāņi netika izmantoti. Tika novērota neliela pasliktināšanās risinājumu precizitātē (attiecīgi kritumi par **0.18%** un **0.32%** LitLat un **0.17%** un **0.43%** - LVBERT). Tāpat, tika secināts, ka modeļiem bija nepieciešamas krietni vairāk epohas, lai konverģētu, izmantojot to pašu apmācības tempu - papildus **23** epohas LitLat, papildus **12** epohas - LVBERT.

Ieguvumu no BERT modeļu pielāgošanas konkrētajam uzdevumam un datiem pārbaudei, tika iesaldēti visi, izņemot modeļu klasifikācijas, slāņi. Izmantojot optimālās konfigurācijas, tika novērots, ka LVBERT modeļa risinājuma precizitāte vārdšķiras prognoze samazinājās par **2.06%** un pilnā morfoloģiskā marķējuma prognoze - par **9.09%**. LitLat modeļa risinājumā šie rādītāji samazinājās par attiecīgi **4.58%** un **18.56%**, indicējot, ka modeļu pielāgošana konkrētajam uzdevumam nodrošina augstākus rezultātus. Tāpat tika konstatēts, ka modeļu konverģēšanai bija nepieciešams ievērojami lielāks epohu skaits.

Labākos rezultātus uzrādošās konfigurācijas testēšanā, tika konstatēts, ka testēšanas laiks aizņem **1 minūti un 15 sekundes**, aprēķinot risinājuma marķēšanas precizitāti uz testa kopu, kas sastāv no 2412 teikumiem. Testēšanas laiks tika mērīts uz darbā izmantotās lokālās darbstacijas.

SECINĀJUMI

Vārdšķiras atpazīšanas uzdevumā, darbā izstrādātā risinājuma precizitātes rezultāti atbilst pasaulē modernāko risinājumu rezultātiem. Tāpat, salīdzinot ar eksistējošo perceptrona risinājumu, kurā sasniegts augstākais marķēšanas rādītājs, šajā darbā izstrādātais risinājums uzrāda labākus rezultātus. Pilnā marķējuma noteikšanā, izstrādātais risinājums uzrādījis zemāku rezultātu. Rezultātu atšķirība var būt saistīta ar katrā gadījumā apsveramo valodniecisko pazīmju kopu. Iespējams, samazinot šo apsveramo valodniecisko pazīmju kopu, varētu tikt sasniegts labāks rādītājs. Svarīgi arī atzīmēt, ka gadu mijas rezultātā, apmacības un testēšanas dati apskatītajos pētījumos un risinājumos var atšķirties, lai gan to tīmekļa resurss ir tas pats.

Ņemot vērā, ka morfoloģiskās marķēšanas risinājumā tika novērotas dažas nepilnības, autors izvirza priekšlikumus turpmākai risinājuma uzlabošanai.

Marķējuma noteikšanā varētu būt lietderīgi izmantot morfoloģiska analizatora piešķirtos iespējamos tagus. Tā vietā, lai marķēšanu veiktu, ģenerējot marķējumus no šajā darbā izstrādātā risinājuma piešķirtajām valodniecisko pazīmju, vērtībām - būtu iespējams apskatīt morfoloģiskā analizatora veidotos iespējamos marķējumus, aprēķināt, kurš iespējamais marķējums vairāk atbilst modeļa prognozētajām marķējuma vērtībām un kā gala rezultātu izvēlēties tieši to. Šis palīdzētu izslēgt iespējamos marginālos gadījumus, kuros modelis piešķir vērtību kādai valodnieciskai pazīmei, kurai nebūtu jābūt apsvērtai gala marķējumā, kā rezultātā viss marķējums tiek uzskatīts par nepareizu.

Darbā izvirzītie mērķi tika sasniegti - tika apskatīti pasaulē modernākie risinājumi morfoloģiskās marķēšanas uzdevumā un veikta to adaptēšana morfoloģiskai marķēšanai latviešu valodā. Tika veiksmīgi izmantotas pētnieciskās metodes, lai izpildītu darba uzdevumus.

Darbā tika izvirzītas sekojošas hipotēzes:

- iespējams sasniegt relatīvi augstus rezultātus morfoloģiskajā marķēšanā, izmantojot modernākos risinājumus latviešu valodā;
- ievades tekstvienību sadalīšana apakštekstvienībās palīdz mazināt ārpus-vārdu-krājuma problēmu un nezaudēt svarīgu informāciju (apakšnodaļa 5.2);
- Papildus LSTM slāņi var palīdzēt saglabāt svarīgu informāciju un uzlabot rezultātus (apakšnodaļa 5.3);
- BERT modeļu pielāgošana morfoloģiskās marķēšanas uzdevumam uzlabo tā sniegumus (apakšnodaļa 5.4).

Darbā izvirzītā hipotēzes tika eksperimentāli pārbaudītas un tika konstatēts, ka visas no tām apstiprinās. Lai gan risinājums nepārsniedz šobrīdējos labākos morfoloģiskās marķēšanas rezultātus, tika sasniegts augsts vārdšķiras atpazīšanas rezultāts un relatīvi augsts morfoloģiskās marķēšanas rezultāts (apakšnodaļa 5.5).

IZMANTOTĀ LITERATŪRA

1. P. Paikens, "Lexicon-based morphological analysis of Latvian language." *Proceedings of the 3rd Baltic Conference on Human Language Technologies*, 2007.
2. Pretkalniņa, Lauma, Laura Rituma, and Baiba Saulīte. "Universal dependency treebank for Latvian: a pilot." *Human Language Technologies–The Baltic Perspective*. IOS Press, 2016. 136-143.
3. Znotiņš, Artūrs. "Jēdzientelpas un to pielietojumi." (2016).
4. Znotins, Arturs, and Guntis Barzdins. "LVBERT: Transformer-Based Model for Latvian Language Understanding." *Baltic HLT*. 2020.
5. Deksne, Daiga. "Finite state morphology tool for Latvian." *Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing*. 2013.
6. Ņikiforovs, Pēteris. "Latviešu valodas morfosintaktiskais marķētājs." (2015).
7. Voļska, Kristīne. "Mašīntulkotu nosaukto entitāšu gramatisko locījumu noteikšana automātiskajā pēcrediģēšanā." (2020).
8. Paikens, Pēteris. "Deep neural learning approaches for Latvian morphological tagging." *Human Language Technologies–The Baltic Perspective*. IOS Press, 2016. 160-166.
9. Heinzerling, Benjamin, and Michael Strube. "Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation." *arXiv preprint arXiv:1906.01569* (2019).
10. Yasunaga, Michihiro, Jungo Kasai, and Dragomir Radev. "Robust multilingual part-of-speech tagging via adversarial training." *arXiv preprint arXiv:1711.04903* (2017).
11. Plank, Barbara, Anders Søgaard, and Yoav Goldberg. "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss." *arXiv preprint arXiv:1604.05529* (2016).
12. Nguyen, Dat Quoc, Mark Dras, and Mark Johnson. "A novel neural network model for joint POS tagging and graph-based dependency parsing." *arXiv preprint arXiv:1705.05952* (2017).
13. Sharipov, Maksud, et al. "UzbekTagger: The rule-based POS tagger for Uzbek language." *arXiv preprint arXiv:2301.12711* (2023).
14. Dincer, Taner, Bahar Karaoglan, and Tarik Kisla. "A suffix based part-of-speech tagger for Turkish." *Fifth International Conference on Information Technology: New Generations (itng 2008)*. IEEE, 2008.

15. Bohnet, Bernd, et al. "Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings." *arXiv preprint arXiv:1805.08237* (2018).
16. Ulčar, Matej, and Marko Robnik-Šikonja. "Training dataset and dictionary sizes matter in bert models: the case of baltic languages." *Analysis of Images, Social Networks and Texts: 10th International Conference, AIST 2021, Tbilisi, Georgia, December 16–18, 2021, Revised Selected Papers*. Cham: Springer International Publishing, 2022.
17. Pretkalniņa, Lauma, et al. "Towards a Latvian Treebank." *Las tecnologías de la información y las comunicaciones: presente y futuro en el análisis de corpus: Actas del III Congreso Internacional de Lingüistica de Corpus*. Editorial Universitat Politècnica de València, 2011.