LATVIJAS UNIVERSITĀTE

Promocijas darba kopsavilkums

Summary of Doctoral Thesis

Lauma Pretkalniņa

FORMĀLS LATVIEŠU VALODAS GRAMATIKAS MODELIS UN TĀ REALIZĀCIJA MAŠĪNLASĀMĀ SINTAKSES KORPUSĀ

FORMAL MODEL OF LATVIAN GRAMMAR AND ITS IMPLEMENTATION IN A MACHINE-READABLE TREEBANK

Rīga, 2023

LATVIJAS
UNIVERSITĀTE

DATORIKAS FAKULTĀTE

Lauma Pretkalniņa

FORMĀLS LATVIEŠU VALODAS GRAMATIKAS MODELIS UN TĀ REALIZĀCIJA MAŠĪNLASĀMĀ SINTAKSES KORPUSĀ

PROMOCIJAS DARBA KOPSAVILKUMS

Doktora grāda iegūšanai datorzinātņu nozarē
Apakšnozare: datoru un sistēmu programmatūra

Rīga, 2023

Promocijas darbs izstrādāts Latvijas Universitātes
Matemātikas un informātikas institūtā
Mākslīgā intelekta laboratorijā
laika posmā no 2011. gada līdz 2023. gadam

Eiropas Sociālā fonda projekts „Atbalsts doktora studijām Latvijas Universitātē ”
Nr. 2009/0138/1DP/1.1.2.1.2./09/IPIA/VIAA/004.

Darbs sastāv no ievada, 2 nodaļām, secinājumiem, literatūras saraksta.

Darba forma: publikāciju kopa datorzinātņu nozarē, datoru un sistēmu programmatūras apakšnozarē

Darba zinātniskais vadītājs: asoc. prof. Dr. Normunds Grūzītis, Latvijas Universitāte

Darba recenzenti:

1) prof. Dr. Ģirts Karnītis, Latvijas Universitāte;
2) asoc. prof. Dr. Andrjus Utka (*Andrius Utka*), Vītauta Dižā Universitāte;
3) Dr. dat. Mārcis Pinnis, SIA “Tilde”.

Promocijas darba aizstāvēšana notiks 2023. gada 14. aprīlī plkst. 16:00 Latvijas Universitātes Datorzinātnes un informātikas nozares promocijas padomes atklātā sēdē Latvijas Universitātes Matemātikas un Informātikas institūtā Rīgā, Raiņa bulvārī 29, 413. telpā.

Ar promocijas darbu un tā kopsavilkumu var iepazīties Latvijas Universitātes Bibliotēkā Rīgā, Kalpaka bulvārī 4.

LU Datorzinātņu nozares promocijas
padomes priekšsēdētājs: Jānis Bičevskis
promocijas padomes sekretāre: Ruta Ikauniece

ISBN 978-9934-18-975-3
ISBN 978-9934-18-974-6 (PDF)

Pateicības

Vēlos pateikties visiem, kas mani ir atbalstījuši, iedvesmojuši un konsultējuši šī darba gaitā, bez jums šis noteikti nebūtu novests līdz galam. Vēlos pateikties ģimenei un draugiem par neizsīkstošo pacietību un atbalstu. Vēlos pateikties kolēģiem par iedvesmu un zināšanām. Vēlos pateikties Normundam, kurš manam darbam ticēja arī tad, kad es neticēju. Vēlos pateikties Baibai un Madarai par darba lasāmību. Vēlos pateikties Ingum par svešvalodām un matemātiku. Vēlos pateikties Jānim par mīlestību un garšīgajām pusdienām.

Šī darba izstrāde tika sākta ar Eiropas Sociālā fonda atbalstu projektā "Atbalsts doktora studijām Latvijas Universitātē" (2009/0138/1DP/1.1.2.1.2/09/IPIA/VIAA/004). Turpmākie fundamentālie pētījumi tika finansēti Valsts pētījumu programmu "Nacionālā identitāte (valoda, Latvijas vēsture, kultūra un cilvēkdrošība) Nr. 3" un "Humanitāro zinātņu digitālie resursi" (VPP-IZM-DH-2020/1-0001) ietvaros. Savukārt rūpnieciskie pētījumi (sadarbībā ar ziņu aģentūru LETA) tika veikti Eiropas Reģionālā attīstības fonda projektos: Informācijas un komunikāciju tehnoloģiju kompetences centra pētījumā Nr. 2.7 "Teksta automātiskās datorlingvistiskas analīzes pētījums jauna informācijas arhīva produkta izstrādē" (KC/2.1.2.1.1/10/02/001) un praktiskas ievirzes pētījumu projektā "Daudzslāņu valodas resursu kopa teksta semantiskai analīzei un sintēzei latviešu valodā" (1.1.1.1/16/A/219). Pētījumu virziens tiek turpināts Valsts pētījumu programmas "Letonika latviskas un eiropeiskas sabiedrības attīstībai" projektā "Mūsdienu latviešu valodas izpēte un valodas tehnoloģiju attīstība" (VPP-LETONIKA-2021/1-0006). Darba izstrāde pabeigta ar Eiropas Sociālā fonda atbalstu projektā "LU doktorantūras kapacitātes stiprināšana jaunā doktorantūras modeļa ietvarā" (8.2.2.0/20/I/006).

Anotācija

Promocijas darbs veltīts hibrīda latviešu valodas gramatikas modeļa izstrādei un transformēšanai uz Universālo atkarību (Universal Dependencies, UD) modeli. Promocijas darbā ir aizsākts jauns latviešu valodas izpētes virziens – sintaktiski marķētos tekstos balstīti pētījumi. Darba rezultātā ir izstrādāts un aprobēts fundamentāls, latviešu valodai iepriekš nebijis valodas resurss – mašīnlasāms sintaktiski marķēts korpuss 17 tūkstošu teikumu apmērā. Teikumi ir marķēti atbilstoši diviem dažādiem sintaktiskās marķēšanas modeļiem – darbā radītajam frāžu struktūru un atkarību gramatikas hibrīdam un starptautiski aprobētajam UD modelim. Izveidotais valodas resurss publiski pieejams gan lejuplādei, gan tiešsaistes meklēšanai abos iepriekš minētajos marķējuma veidos.

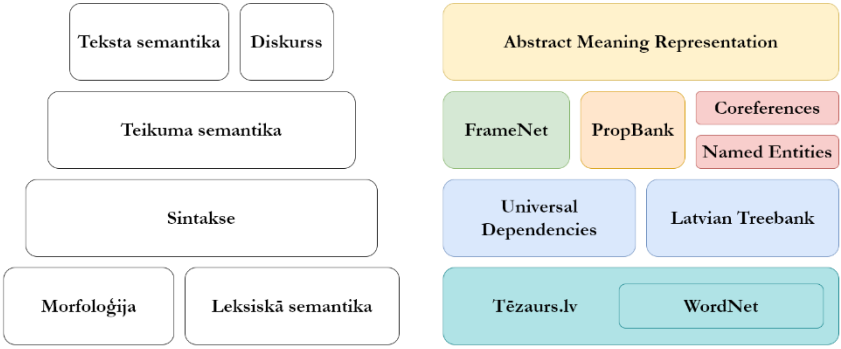
Pētījuma laikā radīta rīku kopa un latviešu valodas sintaktiski marķētā korpusa veidošanai vajadzīgā infrastruktūra. Tajā skaitā tika definēti plašam valodas pārklājumam nepieciešamie LU MII eksperimentālā hibrīdā gramatikas modeļa paplašinājumi. Tāpat tika analizētas iespējas atbilstoši hibrīdmodelim marķētus datus pārveidot uz atkarību modeli, un tika radīts atvasināts UD korpuss.

Izveidotais sintaktiski marķētais korpuss ir kalpojis par pamatu, lai varētu radīt augstas precizitātes (91%) parsētājus latviešu valodai. Savukārt dalība UD iniciatīvā ir veicinājusi latviešu valodas un arī citu fleksīvu valodu resursu starptautisko atpazīstamību un fleksīvām valodām piemērotāku rīku izveidi datorlingvistikā – pētniecības jomā, kuras vēsturiskā izcelsme pamatā meklējama darbā ar analītiskajām valodām.

Saturs

1 Ievads

Promocijas darbs izstrādāts datorlingvistikā – starpdisciplinārā nozarē, kas nodarbojas ar dabiskās valodas modelēšanu un apstrādi ar datorikas metodēm. Tās pamatuzdevums ir automātiska strukturētas, mašīnlasāmas un mašīnai interpretējamas informācijas izgūšana no dabiskās valodas[1], kā arī mašīnlasāmas informācijas (datu) atainošana ar dabiskās valodas līdzekļiem, tādējādi centrālie datorlingvistikas aspekti ir valodas analīze (sapratne) un sintēze (tekstrade). Valodas apstrādi gan valodniecībā, gan datorlingvistikā mēdz aplūkot kā vairāklīmeņu uzdevumu (sk. 1.a attēlu), kur gramatisko un semantisko informāciju katrā no līmeņiem var analizēt, izmantojot formālus nozīmes reprezentācijas modeļus (sk. 1.b attēlu, uzskatāmības labad šeit minēti ar promocijas darbu saistītajos projektos izmantotie reprezentācijas modeļi, lai gan tie nav vienīgie).



(a) Teksta analīzes līmeņi (pēc Grūzītis, 2011)

(b) Izvēlētās reprezentācijas (pēc Grūzītis et al., 2018)

1. attēls. Valodas apstrāde un analīze

Daudzas mūsdienu datorlingvistikas problēmas un to risinājumi ar praktisku lietojumu izriet no valodas analīzes semantikas līmeņos. Piemēram, mūsdienās populāro virtuālo asistentu (*Siri* [2], *Alexa* [3] utt.) darbības nodrošināšana tipiski ietver teksta klasificēšanu (angl. *text*

[1] Šis darbs fokusējas uz tekstu un neaplūko runas atpazīšanas un sintēzes problemātiku, taču jāatzīmē, ka runas atpazīšanas rezultāts ir teksts, kas tālāk jāapstrādā un jāanalizē, un runas sintēzes ieejas dati arī ir teksts.

[2] *Apple Siri* mājaslapa: https://www.apple.com/siri/

[3] *Amazon Alexa* mājaslapa: https://alexa.amazon.com/

classification) un faktu izgūšanu no teksta (angl. *information extraction*). Šādu uzdevumu risināšanai var izšķirt atšķirīgas pieejas atkarībā no tā, kādi starpsoļi un resursi tiek izmantoti risinājuma iegūšanai. Viena no pieejām ir veikt analīzi soli pa solim: sākt ar zemākā līmeņa, t.i., morfoloģisko, analīzi, turpināt ar sintaktisko analīzi, kas balstās morfoloģiskajā analīzē, utt., līdz sasniegts vēlamais analīzes jeb teksta nozīmes reprezentācijas līmenis. Otra pieeja ir uzreiz risināt galaproblēmu, neveicot tiešu, pilnu zemāko līmeņu analīzi. Otrā pieeja risina tikai konkrēto uzdevumu, un risinājums var būt grūti vispārināms citiem tā paša analīzes līmeņa uzdevumiem. Piemēram, atslēgvārdu saraksti vai to jēdzientelpas vektori (angl. *word embeddings*) var būt pietiekams risinājums teksta klasificēšanas uzdevumam, taču ar šo resursu nepietiek, lai izgūtu no teksta faktoloģisku informāciju. Pirmajā pieejā aprakstītā resursu izstrāde ir laika un cilvēkresursu ietilpīgāka, taču jau izstrādāto risinājumu izmantojums ir plašāks un tie atkārtoti noder jaunu lietojumu izstrādē. Tādējādi plaša pārklājuma zemāka līmeņa risinājumu izstrāde ir fundamentāls ieguldījums tālākā augstāka līmeņu problēmu risināšanā. Promocijas darbā veiktais pētījums un tā rezultāti orientēti uz teksta analīzi soli pa solim, t.i., pēc pirmās pieejas.

Viszemākie teksta apstrādes līmeņi ir teksta dalīšana tekstvienībās (angl. *tokens*) un morfoloģiskā analīze. Morfoloģija ir valodniecības apakšnozare, kas pēta un apraksta vārdu gramatisko struktūru, savukārt datorlingvistikā ar morfoloģisko analīzi bieži saprot procesu, kas analizē ne tikai vārdus, bet arī pieturzīmes – visas tekstvienības. Arī šajā līmenī netrūkst atsevišķu problēmgadījumu, tomēr latviešu valodai ir pieejams fundamentāls, visaptverošs risinājums, kas analīzes variantu ģenerēšanai izmanto leksikonu (Paikens, 2007), un statistisks, korpusa datos apmācīts tagotājs ar 98% precizitāti vārdšķiras atpazīšanai un 93% – pilnajam morfoloģiskajam marķējumam, kas ietver, piemēram, locījumu, dzimti un skaitli lietvārdiem un īpašības vārdiem, laiku, personu un skaitli darbības vārdiem u.tml. (Paikens et al., 2013; Paikens, 2017).

Nākamajā, sintakses, līmenī tiek apskatīti vārdu – teikuma elementu – savstarpējie formālie sakari, savukārt teikuma un teksta semantikas līmeņi attiecas uz teikumā[4] vai tekstā iekļauto jēgu. Promocijas darbs ir veltīts valodas analīzei sintakses līmenī. Latviešu valodai pirms promocijas darba sākšanas sintaktiskās analīzes rīku krājums un resursu krājums bija ļoti ierobežots – trūka efektīva plaša pārklājuma parsētāja un sintaktiski marķēta korpusa šāda mašīnmācāma parsētāja izveidei. Eksistēja eksperimentāls likumos balstīts daļējais parsētājs, kas izmantoja manuāli sastādītu likumu kopu (Bārzdiņš et al., 2007), taču šī likumu kopa

[4] Šeit un turpmāk – ja nav norādīts citādi, tad ar *teikumu* īsuma labad tiek saprastas gan predikatīvas, gan nepredikatīvas vienības (izteikumi).

pārklāja tikai vienkāršus paplašinātus teikumus, un iegūtais rezultāts bija daudznozīmīgs – katram teikumam tika piedāvāti visi iespējamie analīzes varianti, kas atbilst teikuma elementu morfoloģiskajām pazīmēm. Eksistēja arī pētījumi, kas veltīti specifiskām problēmām, piemēram, pareizrakstības pārbaudei (Deksne, Skadiņš, 2011) vai ierobežotām dabiskajām valodām (Paikens, Grūzītis, 2012).

Promocijas darbā sekmīgi attīstītās sintakses līmeņa tehnoloģijas, it īpaši Universālo atkarību (*Universal Dependencies*, UD) korpuss, veido nepieciešamo pamatu tālākajiem augstāka līmeņa pētījumiem (sk. 1.b attēlu).

1.1 Pētnieciskā problēma

Promocijas darbā izvirzītā pētnieciskā problēma ir pilnīga (plaša pārklājuma) latviešu valodas formālā gramatikas modeļa izstrāde un aprobācija, t.sk., caur modelim atbilstoša, apjomīga, mašīnlasāma sintaktiski marķēta tekstu korpusa izveidi.

1.2 Problēmas aktualitāte

Promocijas darba rezultāti paver plašas jaunu rīku un pētījumu iespējas.

- Izmantojot sintaktiski marķēto korpusu, ir iespējams izveidot dažādus jaunus valodas analīzes rīkus, īpaši uzsverot augstas precizitātes sintaktiskos parsētājus latviešu valodai un iespēju piedalīties parsētāju būvēšanas sacensībās.
- Sintaktiski marķētais korpuss, kas publiski pieejams divos datu formātos atbilstoši diviem sintaktiskā marķējuma modeļiem, valodas pētījumiem ļauj atlasīt latviešu valodas datus pēc sintaktiskiem kritērijiem, kas pirms tam nebija iespējams.
- Latviešu valodai tiek aprobēts plaši izmantots starptautisks standarts – UD. Latviešu valoda ir pirmā no baltu valodām, kam tiek veidots šāds plašs resurss. Tādējādi dalība UD iniciatīvā veicina starptautisko sadarbību un vēsturiski vairāk anglocentriskajā datorlingvistikas vidē ļauj izplatīties atziņām par fleksīvu valodu, kurās ir bagāta morfoloģija, īpatnībām un vajadzībām.
- Jauni latviešu valodas rīki ir svarīgi ne tikai zinātnei, bet arī plašākai sabiedrībai – tie veicina latviešu valodas plašāku iekļaušanos elektroniskajos saziņas līdzekļos un valodas izdzīvošanu digitālajā laikmetā.

1.3 Pētījuma mērķi un uzdevumi

Pētījuma vispārīgais mērķis ir uzsākt jaunu latviešu valodas izpētes virzienu – sintaktiski marķētos tekstos balstītus datorlingvistikas pētījumus. Pētījuma konkrētais mērķis ir izstrādāt un aprobēt fundamentālu, latviešu valodai līdz šim nebijušu valodas resursu – mašīnlasāmu sintaktiski marķētu korpusu. Darba mērķa sasniegšanai izvirzīti tālāk minētie uzdevumi.

- Izveidot metodes un nepieciešamo tehnisko ietvaru sintaktiski marķēta latviešu valodas korpusa radīšanai, t.sk. salīdzināt dažādus sintaktiskās marķēšanas modeļus.
- Radīt eksperimentālu korpusa prototipu.
- Izstrādāt datu transformācijas sintaktiski marķēta korpusa izmantošanai dažādos formātos, t.sk. atbilstoši starptautiski aprobētajai UD pieejai.
- Sagatavot korpusa datus dalībai parsētāju apmācībā, t.sk. starptautiskās sacensībās, un nodrošināt datu publisku pieejamību dažādiem pētījumiem.

1.4 Hipotēzes

Darbā izvirzītas šādas hipotēzes:

1) atkarību un frāžu struktūru hibrīds gramatikas modelis paplašinās latviešu valodas sintaktiski marķētā korpusa izmantojamību, salīdzinot ar atkarību gramatikas modeli;
2) kvalitatīvs vidēja apjoma (ap 10–20 tūkstoši teikumu[5]) sintaktiski marķēts latviešu valodas korpuss kalpos par pamatu vismodernāko (angl. *state-of-the-art*) parsētāju izveidei.

Darba ietvaros izvirzītās hipotēzes guvušas praktisku apstiprinājumu.

1.5 Pētījuma metodes

Promocijas darbā izmantotas šādas pētījumu metodes:

[5] Retrospektīvi skatoties UD daudzvalodu korpusu datos un *CoNLL* 2018 rezultātos, redzams, ka "liela" korpusa apjoms ir vidēji 12,5 tūkstoši teikumu un labākā parsētāja vidējā precizitāte šāda veida korpusiem ir 84,37%. Par "lieliem" korpusiem šo sacensību organizatori uzskatīja korpusus, kas ir pietiekami lieli, lai mašīnmācīšanās vajadzībām varētu nošķirt ne tikai apmācības (angl. *train*) un novērtēšanas (angl. *test*) datu kopas, bet arī parametru kalibrēšanas (angl. *dev*) kopu.

- literatūras apskats – lai apzinātu perspektīvās metodes un citu valodu pieredzi, analizētas dažādas zinātniskas publikācijas un atsevišķos gadījumos arī atvērtā pirmkoda rīku programmkods;
- iteratīva izstrāde un pielāgošana – darbā radītie rīki un algoritmi tika realizēti, novērtēti un iteratīvi precizēti, vadoties pēc to praktiskā izmantojuma;
- kvantitatīva novērtēšana – darbā izveidotie transformācijas algoritmi tika izvērtēti ar jomā pieņemtajām metrikām;
- kontrolēti eksperimenti – algoritmu varianti tika salīdzināti kontrolētas vides eksperimentos ar kvantitatīvās novērtēšanas palīdzību, analizējot to darbības atšķirības un precizitāti;
- kļūdu analīze – kur iespējams, algoritmu rezultātos izlases veidā tika veikta arī manuāla kļūdu analīze, lai gūtu labāku priekšstatu par iespējamajām problēmām un to veidiem.

1.6 Galvenie rezultāti

Promocijas darba galvenie rezultāti ir šādi.

- Radīta rīku kopa un "Latviešu valodas sintaktiski marķētā korpusa" (*Latvian Treebank*, LVTB) veidošanai vajadzīgā infrastruktūra, t.sk. definēti plašam valodas pārklājumam nepieciešamie LU MII eksperimentālā hibrīdā gramatikas modeļa paplašinājumi.
- Analizētas iespējas hibrīdmodelim atbilstošu datu marķējumu pārveidot par atkarību modelim atbilstošu, pētīta dažādu pārveidojumu ietekme uz parsētāju precizitāti un tālākizmantojamību. Izveidota transformācija, kas no hibrīdajam gramatikas modelim atbilstoši marķētiem datiem rada atvasinātu UD pieejā marķētu korpusu (*Latvian UD Treebank*, UDLV-LVTB).
- Būtiskākais netiešais rezultāts: latviešu valodai radīts jauns starpdisciplināras pētniecības virziens un pamats fundamentālu valodas tehnoloģiju izstrādei: (1) korpuss – LVTB un UDLV-LVTB (latviešu valodas daļa UD versijās v1.3–v2.11); (2) parsētāji latviešu valodai.

1.7 Praktiskā nozīme un rezultātu aprobācija

Izmantojot promocijas darba laikā izveidoto sintakses modeli un marķēšanas infrastruktūru, radīts 17 tūkstošus teikumu liels sintaktiski

marķēts korpuss, kas publiski pieejams gan hibrīdajā [6], gan UD reprezentācijā[7].

Sintaktiski marķētais korpuss ir kalpojis par pamatu LU MII un ziņu aģentūras LETA sadarbībai ERAF praktiskas ievirzes projektā "Daudzslāņu valodas resursu kopa teksta semantiskai analīzei un sintēzei latviešu valodā" (2017–2019; sk. 3.2.1. sadaļu), kurā ir izveidots daudzlīmeņu sintaktiski un semantiski marķēts korpuss un uz tā bāzes – rīkkopa NLP-PIPE (Znotiņš, Cīrule, 2018) pilnai latviešu valodas tekstu analīzei (atbilstoši 1.b attēlam). Savukārt rīkkopu NLP-PIPE savu produktu un pakalpojumu attīstīšanai izmanto LETA, Latvijas Nacionālā bibliotēka u.c. Sintaktiski marķētais korpuss kalpo arī par pamatu tālākiem lingvistiskiem un valodas tehnoloģiju pētījumiem Valsts pētījumu programmās "Humanitāro zinātņu digitālie resursi" (2020–2022) un "Letonika latviskas un eiropeiskas sabiedrības attīstībai" (2022–2024).

Atvasināto sintaktiski marķēto UD korpusu latviešu valodas sintaktisko parsētāju izstrādei ir izmantojušas arī ārvalstu pētnieku grupas. Tas ir izmantots gan *Google SyntaxNet* parsētāja apmācībai, gan četrās starptautiskās UD parsētāju apmācības sacensībās, kurās radītie latviešu valodas parsētāji pārsniedz 85% precizitāti, gan plaši izmantotā daudzvalodu parsētāja *UDPipe*[8] izveidē (Straka et al., 2016; Straka 2018). Tādējādi darba rezultāti ir kalpojuši par pamatu augstas efektivitātes parsētāju radīšanai latviešu valodai (sk. 0. sadaļu). Tāpat UD korpuss tiek izmantots pētījumos, kuros izstrādā fleksīvām valodām piemērotākas metrikas un rīkus (pretstatā vēsturiski dominējošajai angļu valodai, kas ir analītiska valoda), piemēram, CLAS metriku (Nivre, Fang, 2017).

1.8 Pētījuma rezultātu publikācijas

Darbs veidots kā publikāciju kopa, apvienojot 11 autores publikācijas, kurās risināti ar sintaktiski marķēta korpusa izveidi un parsētāju izstrādi saistītie jautājumi. Pētījums izstrādāts Latvijas Universitātes (LU) Matemātikas un informātikas institūta (MII) Mākslīgā intelekta laboratorijā laika posmā no 2010. līdz 2023. gadam vairāku projektu un pētījuma programmu ietvaros. Darbā aprakstītie rezultāti ir kolektīvs darbs, kurā promocijas darba autore ir vadījusi pētījumu vai būtiski piedalījusies šo rezultātu sasniegšanā (sk. promocijas darba tabulā "Promocijas darba autora personiskais ieguldījums" 5. lpp.).

[6] CLARIN-LV repozitorijā http://hdl.handle.net/20.500.12574/63
[7] LINDAT/CLARIN-CZ repozitorijā http://hdl.handle.net/11234/1-4758
[8] *UDPipe* mājaslapa: https://ufal.mff.cuni.cz/udpipe/2

Pētījuma rezultāti publiskoti 7 *Elsevier Scopus* un *Thomson Reuters Web of Science* indeksētās publikācijās:

- Saulīte, B., Darģis, R., Grūzītis, N., Auziņa, I., Levāne-Petrova, K., **Pretkalniņa, L.,** Rituma, L., Paikens, P., Znotiņš, A., Strankale, L., Pokratniece, K., Poikāns, I., Bārzdiņš, G., Baklāne, A., Saulespurēns, V., Ziediņš, J. (2022). *Latvian National Corpora Collection – Korpuss.lv.* Proceedings of 13th International Conference on Language Resources and Evaluation (LREC 2022), Marseille, pp. 5123–5129 (*Scopus*).
- Gruzitis, N., **Pretkalnina, L.**, Saulite, B., Rituma, L., Nespore-Berzkalne, G., Znotins, A., Paikens, P. (2018). *Creation of a Balanced State-of-the-Art Multilayer Corpus for NLU*. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, pp. 4506–4513 (*Scopus* un *WOS*).
- **Pretkalniņa, L.**, Rituma, L., Saulīte, B. (2018). *Deriving enhanced Universal Dependencies from a hybrid dependency-constituency treebank.* Proceedings of the 21sh International Conference "Text, Speech, and Dialogue" (TSD), LNCS, Vol. 11107, Springer Link, pp. 95–105 (*Scopus* un *WOS*).
- **Pretkalniņa, L.**, Rituma, L., Saulīte, B. (2016). *Universal Dependency Treebank for Latvian: a Pilot.* Proceedings of the 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2016), Frontiers in Artificial Intelligence and Applications, Vol. 289, IOS Press, pp. 136–143 (*Scopus* un *WOS*).
- **Pretkalniņa, L.**, Rituma, L. (2014). *Constructions in Latvian Treebank: the Impact of Annotation Decisions on the Dependency Parsing Performance.* Proceedings of the 6th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2014), Frontiers in Artificial Intelligence and Applications, Vol. 268, IOS Press, pp. 219–226 (*Scopus* un *WOS*).
- **Pretkalniņa, L.**, Znotiņš, A., Rituma, L., Goško, D. (2014). *Dependency parsing representation effects on the accuracy of semantic applications – an example of an inflective language.* Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, pp. 4074–4081 (*Scopus* un *WOS*).
- **Pretkalniņa, L.**, Rituma, L. (2012). *Syntactic Issues Identified Developing the Latvian Treebank.* Proceedings of the 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2012), Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 185–192 (*Scopus* un *WOS*).

Vēl pētījuma rezultāti publicēti arī 4 citos starptautiski recenzētos izdevumos:

- **Pretkalniņa, L.**, Rituma, L. (2013) *Statistical syntactic parsing for Latvian.* Proceedings of the 19th Nordic Conference of Computational Linguistics, NEALT Proceedings Series, Vol. 16, Oslo, pp. 279–289.
- **Pretkalniņa, L.**, Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011). *A Prague Markup Language Profile for the SemTi-Kamols Grammar Model.* Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011), Riga, pp. 303–306.
- **Pretkalniņa, L.**, Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011). *Towards a Latvian Treebank.* M.Á. Mora, M. Carrió Pastor, (ed.): Actas del 3 Congreso Internacional de Lingüística de Corpus. Tecnologias de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus, Candel, Valence, pp. 119–127.
- **Pretkalniņa, L.**, Levāne-Petrova, K. (2011). *Preparatory Work for Latvian Treebank.* Proceedings of International Conference CORPUS LINGUISTICS – 2011, St. Petersburg, pp. 53–58.

Par darba rezultātiem autore referējusi 10 starptautiskās konferencēs:

- 21st International Conference "Text, Speech, and Dialogue" (TSD), Brno, 2018;
- 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Rīga, 2016;
- 6th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Kaunas, 2014;
- 9th International Conference on Language Resources and Evaluation (LREC), Reykjavik, 2014;
- 19th Nordic Conference of Computational Linguistics (NODALIDA), Oslo, 2013;
- 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Tartu, 2012;
- 17. starptautiskā zinātniskā konference "Vārds un tā pētīšanas aspekti", Liepāja, 2012;
- 18th Nordic Conference of Computational Linguistics (NODALIDA), Riga, 2011;
- International Conference CORPUS LINGUISTICS – 2011, St. Petersburg, 2011;
- 3rd International Conference on Corpus Linguistics, Valende, 2011.

Ar darbu saistīti rezultāti prezentēti arī divās vietēja līmeņa konferencēs:

- 73. LU konference "Latviešu valodas sintaktiski anotētā korpusa attēlošana universālā atkarību formātā", 2015;

- 72. LU konference "Marķējuma transformācijas sintaktiski marķētā latviešu valodas tekstu korpusā", 2014.

Doktorantūras studiju laikā autore piedalījusies arī šādu starptautiski recenzētu, ar darba tēmu netieši saistītu publikāciju izstrādē:

- Paikens, P., Klints, K., Lokmane, I., **Pretkalniņa, L.**, Rituma, L., Stāde, M., Strankale, L. (2023). *Latvian WordNet*. Proceedings of the 12th Global Wordnet Conference (GWC2023), *tiks publicēta.*
- Paikens, P., Rituma, L., **Pretkalniņa, L.** (2022). *Towards Word Sense Disambiguation for Latvian.* Baltic Journal of Modern Computing, 10(3), 402–408 (*Scopus*).
- Paikens, P., Grasmanis, M., Klints, A., Lokmane, I., Pretkalnina, L., Rituma, L., Stade, M., Strankale, L. (2022). *Towards Latvian WordNet.* Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022), Marseille, pp. 2808–2815 (*tiks indeksēta Scopus*).
- Paikens, P., Gruzitis, N., Rituma, L., Nespore, G., Lipskis, V., **Pretkalnina, L.**, Spektors, A. (2019). *Enriching an Explanatory Dictionary with FrameNet and PropBank Corpus Examples.* Proceedings of the 6th Biennial Conference on Electronic Lexicography (eLex), pp. 922–933 (*Scopus*).
- **Pretkalnina, L.**, Paikens, P. (2018). *Extending Tezaurs.lv online dictionary into a morphological lexicon*. Proceedings of the 8th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2018), Frontiers in Artificial Intelligence and Applications, Vol. 307, pp. 120–125 (*Scopus*).
- Saulīte, B., **Pretkalniņa, L.**, Spektors, A. (2017). *Pirmās konjugācijas darbības vārdi Tēzaurā.* Vārds un tā pētīšanas aspekti: rakstu krājums 21 (1/2), Liepāja, 122.–129. lpp.
- Spektors, A., Auzina, I., Dargis, R., Gruzitis, N., Paikens, P., **Pretkalnina, L.**, Rituma, L., Saulite, B. (2016). *Tezaurs.lv: The largest open lexical database for Latvian.* Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, pp. 2568–2571 (*Scopus*).
- Paikens, P., Rituma, L., **Pretkalniņa, L.** (2013). *Morphological analysis with limited resources: Latvian example.* Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), NEALT Proceedings Series, Vol. 16, Oslo, pp. 267–277.

2 Latviešu valodas sintaktiskās analīzes hibrīdmodelis un korpuss

Sintaktiski marķēts korpuss ir būtisks valodas resurss, kas ļauj izveidot datos balstītu sintaktisko parsētāju, kā arī dod iespēju jauna veida valodniecības pētījumiem.

Veidojot sintaktisko analizatoru jeb parsētāju, tiek apkopotas zināšanas par analizējamās valodas sintaktisko struktūru, t.i., ir nepieciešams definēt modeli, kādā sintaktiskā informācija tiks attēlota, un ir vajadzīgi dati par konkrētās valodas raksturu. Kad sintaktiskās analīzes modelis ir formulēts, zināšanas par valodas īpatnībām var fiksēt un apkopot gan cilvēks, speciālists, likumu veidā, gan dators mašīnmācīšanās (angl. *machine learning*) ceļā (piem., izmantojot statistiskas metodes vai neironu tīklus) no atbilstoši sagatavotas datu kopas.

Pirms šī darba sākšanas latviešu valodai nebija sintaktiski marķēta korpusa un līdz ar to arī statistisku parsētāju būvēšanas iespēju. Bija izmēģināta speciālista rakstītos likumos balstīta parsētāja pieeja: VPP "Informācijas tehnoloģiju zinātniskā bāze" projektā "Semantiskā tīmekļa izpēte, attīstīšana un piemērošana Latvijas vajadzībām" (*SemTi-Kamols*) (2005–2009) tika radītas iestrādes Lisjēna Tenjēra (*Lucien Tesnière*) atkarību gramatikā (Tesnière, 1959) (sīkāk sk. 2.1. sadaļā) balstītam hibrīdam latviešu valodas sintakses modelim, uz kā pamata izveidots *Čankeris* (Bārzdiņš et al., 2007; Nešpore et al., 2010). *Čankeris* bija daļējs analizators, kas, izmantojot lingvistu pierakstītus likumus, spēja pilnas pārlases veidā izanalizēt lielāko daļu vienkāršu paplašinātu teikumu, kā arī vienkāršiem teikumiem atbilstošas daļas saliktos teikumos. *Čankeris* katram analizējamajam teikumam kā rezultātu sniedza visus formāli iespējamos analīzes variantus, jo tā izveides brīdī nebija pieejams sintaktiski marķēts korpuss, kas ļautu novērtēt kādus variantus kā iespējamākus. Ar šādu pieeju piedāvāto variantu skaits pieaug eksponenciāli atkarībā no teikuma garuma.

Analizējot *Čankera* rezultātus, atklājās vairākas problēmas, kas raksturīgas daudziem likumos balstītiem (angl. *rule-based*) analizatoriem. Būtiskākā no tām ir mērogojamības grūtības: pieaugot likumu skaitam, būtiski (eksponenciāli) palielinās vidējais teikumam piedāvāto analīzes variantu skaits, tāpēc samazinās ātrdarbība un likumu mijiedarbība kļūst arvien sarežģītāka, grūtāk izsekojama un grūtāk atkļūdojama.

Palielinoties piedāvātajam analīzes variantu skaitam, arvien būtiskāks kļūst jautājums par to, kā no daudzajiem analīzes variantiem izvēlēties vienu, kuru piedāvāt lietotājam vai izmantot tālākos lietojumos. Dabiska vēlme būtu izvēlēties no visiem variantiem pareizo vai, ja tāda nav,

tad vismazāk kļūdaino, taču valodā netrūkst situāciju, kad teksta fragmentam iespējami vairāki gramatiski pareizi analīzes varianti un cilvēka sagaidītā interpretācija nav viennozīmīgi nosakāma tikai no gramatikas zināšanām. Piemēram, vārdu savienojumu "sieviešu ādas zābaki" un "liellopa ādas zābaki" gadījumā tās ir pasaules zināšanas, kuru dēļ mēs secinām, ka, visticamāk, pirmajā gadījumā zābaki ir paredzēti sievietēm, bet otrajā – gatavoti no liellopa ādas, nevis paredzēti liellopiem. Plaša dziļu pasaules zināšanu integrēšana likumos balstītos parsētājos būtu ārkārtīgi sarežģīta, taču labus rezultātus gūt palīdz statistiski dati par to, kādas struktūras ir biežāk raksturīgas kādam valodas materiālam, piemēram, ka "sieviešu zābaki" ir daudz biežāk lietots formulējums nekā "liellopa zābaki". Tādējādi nācās secināt: pat ja tiktu atrisinātas citas mērogojamības problēmas, *Čankera* tālākai attīstīšanai par plaša pārklājuma praktiski izmantojamu sintakses parsētāju jebkurā gadījumā būtu nepieciešams sintaktiski marķēts korpuss, no kura šādu statistisko informāciju iegūt.

Savukārt, ja izvēlas parsētāju būvēt, par pamatu izmantojot no datiem izskaitļotas statistiskas likumsakarības vai neironu tīklus, kas jāapmāca datos, sintaktiski marķēts korpuss kā šo likumsakarību avots kļūst par fundamentālu valodas digitālo pamatresursu.

META-NET pārskata pētījumā (Vasiļjevs, Skadiņa, 2012) 2012. gadā norādīts, ka latviešu valodai nav publiski pieejams neviens sintaktiski marķēts korpuss. Tādējādi šāda korpusa radīšana un publiskošana, it īpaši atvērto datu veidā, sniedz jaunas iespējas valodas tehnoloģiju pētījumos un izstrādē, kā arī valodniecības studijās un pētījumos. Šāds korpuss ļauj valodniekiem pārbaudīt praksē teorijā aprakstīto izpratni par sintaktiskajām parādībām latviešu valodā un precizēt to atbilstoši datos balstītiem novērojumiem.

2.1 Sintaktiski marķētā korpusa gramatikas modelis

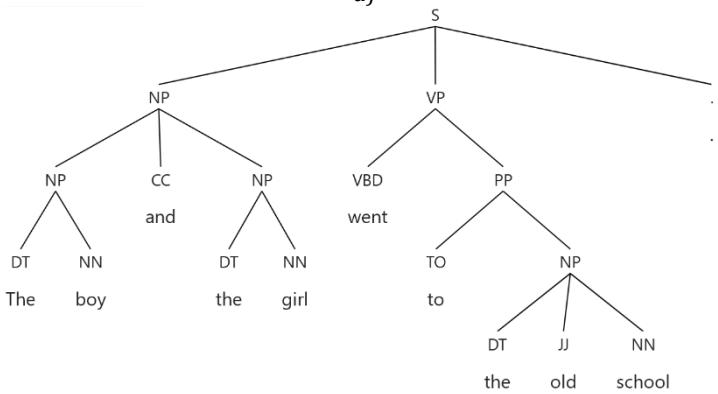
Balstoties uz veiksmīgajām hibrīdā gramatikas modeļa iestrādēm *Čankera* izveidē *SemTi-Kamols* projektā un konsultējoties ar lingvistiem, tika nolemts šo modeli tālāk attīstīt sintaktiski marķētā korpusa vajadzībām ar ilgtermiņa mērķi modelēt visas veidojamajā korpusā sastopamās latviešu valodas konstrukcijas.

"Latviešu valodas sintaktiski marķētā korpusa" (*Latvian Treebank*, LVTB) gramatikas modelis ir veidots kā divu pasaulē plaši lietotu sintaktisko modeļu – atkarību gramatikas un frāzes struktūras gramatikas – hibrīds.

Teikuma struktūras formālie attēlojumi no matemātiskā viedokļa ir grafi, tāpēc turpmāk darbā lietota grafu teorijas terminoloģija. Grafu veido divu tipu elementi – galīgs skaits **virsotņu** (angl. *node*, *vertex*) un **šķautnes** (angl. *edge*), kas šīs virsotnes pa pāriem savieno (katras divas virsotnes – ne vairāk kā viena šķautne). Parasti teikuma struktūra ir **sakņots koks** (angl. *rooted tree*) – grafs bez cikliem ar vienu īpaši atzīmētu virsotni, ko sauc par **sakni** (angl. *root*). Katrai virsotnei pievienotās virsotnes dalās divās grupās – vecāki un bērni. **Vecāks** (angl. *parent*) ir tāda virsotne, kuras attālums (šķautņu skaita ziņā) līdz saknei ir mazāks nekā dotajai virsotnei, un tāda katrai koka virsotnei, izņemot sakni, ir tieši viena (sakne ir vienīgā virsotne, kam vecāka nav). **Bērni** (angl. *children*) ir pārējās virsotnes – tām attālums līdz saknei ir lielāks nekā dotajai virsotnei. Virsotnes, kurām bērnu nav, sauc par **lapām** (angl. *leaves*). Virsotnes, kas ir dotās virsotnes vecāki, vecāku vecāki utt. (būšanas vecākam transitīvais slēgums), sauc par dotās virsotnes **priekštečiem** (angl. *ancestors*). Virsotnes, kas ir dotās virsotnes bērni, bērnu bērni utt. (būšanas bērnam transitīvais slēgums), sauc par dotās virsotnes **pēctečiem** (angl. *descendants*). Atsevišķos gadījumos grafs, kas attēlo viena teikuma struktūru, var būt nesakarīgs, t.i., grafs var būt vairāku koku kopa – **mežs** (angl. *forest*), taču praktiskie lietojumi no šādām struktūrām bieži vairās.

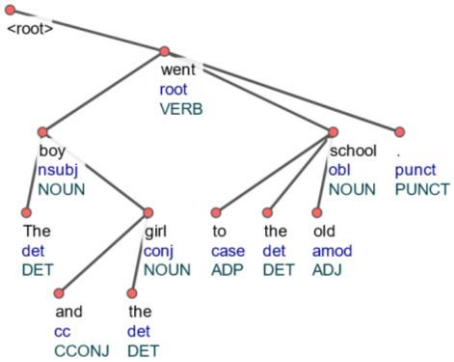
Frāzes struktūras gramatikā (angl. *phrase structure grammar*) (Chomsky, 1957) par teikuma struktūras pamatvienību tiek uzskatīta frāze. Katra frāze sastāv no **tekstvienībām** (angl. *tokens*) – vārdiem vai pieturzīmēm – un/vai citām frāzēm, un teikums tiek attēlots ar koku, kura sakne ir frāze, kas atbilst visam teikumam, savukārt lapas – tekstvienības. Šāda koka piemērs dots 2.a attēlā. Pamata modelī frāzēm tiek uzstādīta prasība būt **nepārtrauktām** (angl. *continuous*), t.i., ja apakškokam, kura saknē ir frāze X, pieder gan i-tā, gan j-tā tekstvienība teikumā, tad arī visām tekstvienībām intervālā $[i, j]$ ir jāpieder tam pašam apakškokam. Praksē šis ierobežojums mēdz tikt pārkāpts, lai saglabātu līdzīgu marķējumu līdzīgām valodas parādībām, piemēram, ja pārfrāzētam teksta fragmentam *the school they went to* ‘skola, uz [kuru] viņi gāja’ vēlas saglabāt attēlā dotajam piemēram analoģisku frāžu dalījumu, tad nākas veidot **pārtrauktu frāzi** (angl. *discontinuous phrase*) – starp darbības vārda frāzes (VP) daļām *went to* ‘gāja uz’ un *the school* ‘skola’ atrodas *they* ‘viņi’.

a)



Avots: *Berkley Neural Parser* demonstrācija https://parser.kitaev.io/

b)



Avots: *UDPipe* parsētāja demonstrācija http://lindat.mff.cuni.cz/services/udpipe/

2. attēls. Piemērs teikuma marķējumam atbilstoši frāžu gramatikas modelim (a) un atkarību gramatikas modelim (b).
Teikums: *The boy and the girl whent to the old scool.* 'Zēns un meitene gāja uz veco skolu.'

Atkarību gramatika (angl. *dependency grammar*) (Mel'čuk, 1988) par teikuma struktūras pamatvienību uzskata vārdu un teikumu modelē ar orientētām, binārām attieksmēm starp vārdiem – **atkarībām** (angl. *dependencies*). Par vienkārša teikuma struktūras centrālo elementu uzskata izteicēju, tam tālāk var būt atkarīgie, un katram atkarīgajam tālāk atkal var būt atkarīgie. Teikuma struktūra ir koks, un katra no tā virsotnēm atbilst vienam vārdam. Korpusu marķēšanas vajadzībām modelis var tikt papildināts tā, lai kokā kā virsotnes tiktu iekļautas visas teikumu veidojošās

tekstvienības, ieskaitot pieturzīmes (Hajič et al., 2000). Šāda koka piemērs dots 2.b attēlā. Atkarību gramatikā balstītu modeli izmanto arī lietuviešu valodas, kas ir latviešu valodai tipoloģiski vistuvākā dzīvā valoda, sintaktiski marķētais korpusā ALKSNIS[9] (Bielinskienė et al., 2016), taču tā izstrāde tika sākta vēlāk – pēc tam, kad latviešu valodas sintaktiski marķētā korpusa modelis jau bija izvēlēts.

Mūsdienu atkarību gramatikas izcelsme meklējama L. Tenjēra gramatikas modelī (Tesnière, 1959). Viņa darbā papildus atkarību attieksmēm teikumā tiek definētas arī citas attieksmes, piemēram, **transferences** (fr. *translation*) operācija, kas no prievārda un ar to saistītā pamatvārda veido vienu "virtuālu" vārdu, kas tālāk teikumā funkcionē kā īpašs pamatvārda locījums. Atsevišķa konstrukcija – **junkcija** (fr. *jonction*) – paredzēta vienlīdzīgu teikuma locekļu attēlošanai. Par koka virsotnēm šajā modelī tiek uzskatīti nevis vārdi, bet ***nucléus*** (fr.) – elementi, kas var būt vai nu viens vārds, vai ar aprakstītajām operācijām iegūts vairāku vārdu savienojums.

Frāzes struktūras gramatikas un atkarību gramatikas marķējumi formālā līmenī ir pielīdzināmi viens otram un pārveidojami no viena marķēšanas modeļa otrā un atpakaļ, ja vien frāžu marķējumā katrai frāzei ir norādīts galvenais elements. Tad katru frāzi var pārveidot par atkarību komplektu starp galveno vārdu un pārējiem frāzes elementiem, savukārt katru atkarību var uztvert kā divu elementu frāzi, kuras galvenais elements ir atkarības neatkarīgais elements. Pārtrauktām frāzēm frāzes struktūras gramatikā atbilst neprojektīvas šķautnes atkarību gramatikā. Pēc Nivre, Nilsson (2005): ja virsotņu (tekstvienību) pāri *v* un *w* savieno šķautne, tā ir **neprojektīva** (angl. *non-projective*) tad un tikai tad, ja kāda no tekstvienībām, kas teikumā atrodas starp *v* un *w*, nav nedz *v*, nedz *w* pēctecis. Par **neprojektīvu koku** (pretstatā projektīvam, angl. *projective*) sauc tādu koku, kurā ir vismaz viena neprojektīva šķautne. Var viegli pārliecināties, ka, veicot transformāciju starp modeļiem augstāk aprakstītajā veidā, neprojektīva šķautne kļūst par pārtrauktu frāzi un otrādi.

SemTi-Kamola hibrīdajā gramatikas modelī tā sākotnējā izstrādes stadijā tika izmantotas atkarības un x-vārdi – L. Tenjēra *nucléus* līdzīgas konstrukcijas atsevišķu vārdu savienojumu attēlošanai (Nešpore et al., 2010). Šajā darbā hibrīdais gramatikas modelis ir būtiski papildināts un pilnveidots, lai novērstu trūkumus, kas tika apzināti korpusa marķēšanas gaitā.

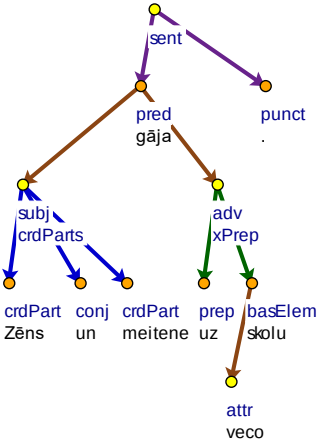
Pilnveidotais LVTB modelis (Pretkalniņa et al., 2011a; Pretkalniņa, Levāne-Petrova, 2011) ir hibrīds attiecībā pret iepriekš aprakstītajiem sintakses modeļiem – tas teikuma struktūru attēlo kā atkarību koku, kurā

[9] CLARIN-LT repozitorijā http://hdl.handle.net/20.500.11821/21

dažas no virsotnēm var būt frāzēm līdzīgas konstrukcijas (sk. piemēru 3. attēlā). Šīm konstrukcijām var būt gan kopīgi atkarīgie, kas attiecināmi uz visu konstrukciju, gan individuāli atkarīgie, kas attiecināmi uz atsevišķu konstrukcijas sastāvdaļu. Katra frāzes sastāvdaļa ir vai nu tekstvienība, vai cita frāze, tādējādi LVTB hibrīdmodelis ir vispārinājums, par kura speciālgadījumiem var uzskatīt gan frāžu, gan mūsdienu atkarību modeļus.

Marķējot korpusu, hibrīdmodeļa atkarību relācijas tiek izmantotas, lai attēlotu sintaktiskā pakārtojuma attieksmes, piemēram, *maza <u>māja</u>*; *<u>iet</u> skolā*; *ilgi <u>gulēt</u>* (neatkarīgais komponents norādīts ar pasvītrojumu).

Frāzes veida konstrukcijas modelī tiek iedalītas trīs grupās: x-vārdi, sakārtojuma konstrukcijas un pieturzīmju konstrukcijas.



3. attēls. Piemērs teikuma marķējumam atbilstoši LVTB modelim, apzīmējumi: zaļās šķautnes – x-vārda (frāzes) sastāvdaļas, zilās šķautnes – sakārtojuma konstrukcijas (frāzes) sastāvdaļas, violetās šķautnes – pieturzīmju konstrukcijas (frāzes) sastāvdaļas, brūnās šķautnes – atkarības

Ar x-vārdiem šobrīd attēlo konstrukcijas, ko latviešu valodā izsaka ar analītiskajām formām, piemēram, saliktus izteicējus, prievārdu konstrukcijas. Šīm konstrukcijām raksturīga stingra iekšējā vārdu secība, kā arī stingri definēts elementu skaits un veids. Piemēram, prievārda konstrukcijas veido viens prievārds un viens nomens, un prievārds nosaka to, vai prievārds ir pirms nomena vai pēc – tādi prievārdi kā *ap*, *uz*, *pār* tiek lietoti pirms nomena, bet *dēļ*, *labad* – pēc nomena (lai gan ir atsevišķi prievārdi, kam valodas materiālā sastopami lietojumi abās pozīcijās – *dēļ*, *pēc*). Šāda x-vārdu izpratne precizē sākotnējā *SemTi-Kamola* modelī paredzēto x-vārdu ideju: lai gan sākotnējā *SemTi-Kamola* modelī x-vārdos bija iekļautas arī dažas citas konstrukcijas, piemēram, vienlīdzīgi teikuma

locekļi un divdabja teicienu pamatelementi, tomēr sākotnējā *SemTi-Kamola* modeļa autori uzskatīja, ka visu pieturzīmju iekļaušana x-vārdos neatbilst iecerētajai idejai.Latviešu valodā pieturzīmes ļauj spriest par gramatisko struktūru, tāpēc tika nolemts, ka sintaktiski marķētajā korpusā jāmarķē arī pieturzīmes, turklāt, ja tas ir iespējams, atainojot pieturzīmju likšanas motivāciju. Kā šīs problēmas atrisinājums tika radīts otrs frāzes veida konstrukciju tips – pieturzīmju konstrukcija. Šī konstrukcija satur pamatelementu – (visbiežāk vienu) vārdu vai frāzi – un pieturzīmes, kas teikumā lietotas šī pamatelementa dēļ. Piemēram, teikumā *Viņš, ēzdams ķiršus, nosmērēja kreklu.* ir pieturzīmju konstrukcija, kas satur *ēzdams* kā pamatvārdu un abus komatus, jo tie tiek lietoti *ēzdams* veidotā divdabja teiciena dēļ.

Pilnveidotajā LVTB modelī tiek lietots vēl viens frāzes veida konstrukciju tips: līdzīgi kā oriģinālajā L. Tenjēra atkarību gramatikā arī šeit tiek šķirta atsevišķa konstrukcija koordinētiem elementiem. Šī konstrukcija LVTB modelī tiek konsekventi lietota gan vienlīdzīgu teikuma locekļu attēlošanai, piemēram, *zēns un meitene*, gan vienlīdzīgu teikuma daļu attēlošanai, piemēram, *zēns ir mājās, bet meitene iet uz skolu* un *zēns ir mājās, jo viņam šorīt bija iesnas un māte atļāva neiet uz skolu*. Sakārtojuma konstrukcijas dažos aspektos strukturāli ir līdzīgas abiem augstāk aprakstītajiem frāžu tipiem, tomēr sakārtojuma konstrukciju struktūra nav tik stingri fiksēta kā x-vārdu struktūra, un tās var saturēt arī pieturzīmes, ja tādas atdala vienlīdzīgos teikuma locekļus vai teikuma daļas.

Vēl LVTB modelis paredz šādu papildinformāciju: atkarību attieksmēm ir dažādi veidi jeb sintaktiskās lomas (piemēram, apstāklis, apzīmētājs, papildinātājs, situants, determinants u.c.) un katram frāzes veida konstrukciju tipam ir vairāki paveidi. X-vārdi paveidos sadalīti atbilstoši to uzbūves atšķirībām (prievārdiskās konstrukcijas ir viens veids, salikti izteicēji vēl viens), pieturzīmju konstrukcijas – pēc pamatelementa veida un pieturzīmju lietojuma motivācijas. Sakārtojuma konstrukciju paveidi parāda, vai šādi marķētā konstrukcija ir vienlīdzīgi teikuma locekļi vai teikuma daļas.

Modeļa formālā struktūra tika fiksēta pirmajā korpusa marķēšanas mēnesī, savukārt noteiktu valodas konstrukciju marķējumu lingvisti, kas strādā ar korpusu, turpina precizēt visā marķēšanas gaitā, kā rezultātā var mainīties frāzes veida konstrukciju un atkarību tipi, un šādos gadījumos tehniskajiem atbalsta risinājumiem ir jābūt viegli pielāgojamiem. Korpusa marķēšanas gaitā nākas saskarties ar latviešu valodas sintakses teorijas nepilnībām – dažādiem neskaidriem robežgadījumiem un teorijā nepilnīgi aprakstītām parādībām. Daļa korpusa veidošanā konstatēto teorētisko problēmu apkopotas (Pretkalniņa, Rituma, 2012). Modeļa lingvistiskās detaļas sīkāk aprakstītas (Rituma et. al, 2019). Kopumā novērojams, ka

modelis pēc daudzām attīstības iterācijām ir nostabilizējies, tāpēc uzskatāms par pabeigtu.

2.2 Korpusa izveide

Šajā sadaļā aprakstīta LVTB izveide. Vispirms tiek raksturoti nepieciešamie rīki un datu formāti, pēc tam – korpusa izveides gaita.

2.2.1 Rīki un formāti sintaktiski marķētajam korpusam

Tā kā LVTB datiem jābūt ērti glabājamiem, apskatāmiem un rediģējamiem, bija nepieciešams radīt vai pielāgot atbalsta rīkus un datu glabāšanas formātus.

Kā pamata rīku komplekts korpusa manuālajai apstrādei tiek lietots Kārļa Universitātē izstrādātais *TrEd toolkit* (Hajič et al., 2001) un kopā ar to arī datu metaformāts *Prague Markup Laguage* (PML) (Pajas, Štěpánek, 2006). PML ir šīs rīku kopas vietējais (angl. *native*) datu formāts. Šāda izvēle tika izdarīta *TrEd toolkit* un PML plašās funkcionalitātes dēļ: ir izstrādāts vizuālas rediģēšanas rīks *TrEd*, meklēšanas valoda kokveida struktūrām PML-TQ ar realizāciju (Štěpánek, Pajas, 2010), masveida apstrādes rīks *bTrEd*[10] u.c. Tie ir aprobēti, marķējot vairākus korpusus, arī lielus – Prāgas Atkarību korpusu (*Prague Dependency Treebank*, PDT) (Hajič et al., 2000), Prāgas Arābu valodas atkarību korpusu (*Prague Arabic Dependency Treebank*) (Hajič et al., 2004), Slovēņu valodas atkarību korpusu (*Slovene Dependency Treebank*) (Džeroski et al., 2006) u.c. Turklāt Kārļa Universitāte LINDAT/CLARIN iniciatīvas ietvaros piedāvā šādu datu publicēšanas pakalpojumu[11].

Tāpat vērtīgi ir arī tas, ka PML standarts ļauj tekstam pievienoto marķējumu sadalīt vairākos līmeņos un katra līmeņa datus glabāt atsevišķā failā – tas ļauj vienā līmenī glabāt morfoloģisko marķējumu (vienas tekstvienības ietvaros) un citā – sintaktisko, tādējādi veidojot vienotu glabāšanas standartu gan morfoloģiski, gan sintaktiski marķētajiem korpusiem. Šādi veidotai struktūrai ir vieglāk pievienot jaunus marķējuma līmeņus, ja nākotnē rodas vēlme sintaktiski marķēto korpusu papildināt ar augstāka līmeņa marķējumu. Papildus tam PML ir salīdzinoši ērti lietojams arī ārpus *TrEd toolkit* rīkiem, jo PML ir XML (*eXtensible Markup Language*)

[10] *bTrEd* tehniskā dokumentācija
https://ufal.mff.cuni.cz/pdt2.0/doc/tools/tred/btred.html

[11] Kārļa Universitātes sintaktisko korpusu repozitorijs
https://lindat.mff.cuni.cz/repository/xmlui/discover?filtertype=subject&filter_relational_operator=equals&filter=treebank

metaformāta apakšformāts, tāpēc PML ir apskatāms ar XML redaktoriem un apstrādājams ar XML apstrādes rīkiem. Tas savukārt nodrošina šādā formātā esošu datu lietojamību nākotnē šobrīd vēl neapzinātos uzdevumos.

Lai gan *TrEd toolkit* tiek postulēts kā izmantotajām sintaktisko koku struktūrām universāls rīku komplekts, tomēr līdz šim tas pamatā lietots tieši atkarību korpusiem, tāpēc, lai pielāgotu to LVTB vajadzībām, tika definēts jauns PML datu formāts (konkrētais formāts, kas atbilst vispārīgajai PML specifikācijai – metaformātam), kas paredzēts tieši LVTB ar 2.1. sadaļā aprakstītajam modelim atbilstošu marķējumu. Pēc analoģijas ar PDT arī topošā latviešu valodas sintaktiski marķētā korpusa marķējums tiek sadalīts trīs līmeņos – dalījums tekstvienībās, morfoloģiskais marķējums un sintaktiskais marķējums. Tekstvienību un morfoloģiskā marķējuma līmeņi veidoti maksimāli tuvi PDT atbilstošajiem līmeņiem, pārņemot aprobētu praksi un nodrošinot vieglāku saprotamību pētniekiem, kas strādājuši ar PDT. Sintaktiskā marķējuma līmenim par pamatu izmantots PDT analītiskā līmeņa formāts, kas attēlo atkarību kokus. Pielāgojot to LVTB vajadzībām, tas papildināts ar frāzes tipa konstrukcijām un tukšām virsotnēm (virsotnēm bez atbilstības ar kādu tekstvienību; PDT tādu nav) vārdu izlaidumu (redukciju, angl. *ellipsis*) attēlošanai. Arī papildināšana ar frāzes tipa virsotnēm praksē veikta, ieviešot īpaša veida tukšas papildvirsotnes, kuras kalpo frāzes kā vienota kopuma attēlošanai.

Lai adekvāti varētu šo formātu izmantot, tā īpatnības tika definētas *PML Schema* standartam [12] atbilstošās PML shēmās un tika radīts arī paplašinājuma modulis grafiskajam redaktoram *TrEd* (Pretkalniņa et al., 2011b). Paplašinājuma modulis satur gan dažādas ar taustiņu īsceļiem pieejamas funkcijas, kas padara ērtāku manuālo sintaktisko marķēšanu, gan stila lapas vizualizācijai, ko lieto gan *TrEd*, gan arī Kārļa Universitātes CLARIN/LINDAT repozitorijs, kas piedāvā šāda veida datus turēt publiski pieejamus un katram interesentam vaicājamus.

Papildus tika izveidota XSL (*Extensible Stylesheet Language*) transformācija, kas ļauj korpusa datus pārveidot *Tiger XML* formātā (Mengel, Lezius, 2000), ko izmanto Štutgartes Universitātes izstrādātais *TigerSearch*[13] un citi rīki.

[12] *PML Schema* tiek definēta *PML* specifikācijā 6. sadaļā "*PML schema file*" https://ufal.mff.cuni.cz/jazz/pml/doc/pml_doc.html#pml-schema

[13] *TigerSearch* mājaslapa http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/tigersearch.html

2.2.2 Korpusa iteratīvā izveide un attīstība

LVTB marķēšana sākta 2010. gadā. Dažādu projektu ietvaros korpuss ir pieaudzis līdz pat 17 tūkstošiem teikumu 2022. gadā un tādējādi kļuvis par būtisku latviešu valodas datorlingvistikas resursu.

Raksturojot LVTB attīstību, nosacīti var runāt par divām fāzēm – eksperimentālu sākotnējo marķēšanas fāzi (angl. *pilot project*) un masveida korpusa paplašināšanas fāzi no apmēram 2016. gada.

Sākotnēji eksperimentālajā fāzē korpusu marķēja viens sintakses speciālists un automatizētas priekšmarķēšanas iespējas bija minimālas. Šādi vairāku gadu laikā korpuss sasniedza apmēram 1500 teikumu apjomu, bet pēc morfoloģiskā tagotāja (Paikens et al., 2013) izveides un integrācijas marķēšanas procesā korpuss sasniedza 5000 teikumu 2014. gadā (Rituma et. al, 2019).

2016. gadā sākās LVTB masveida paplašināšana. ERAF projekta "Daudzslāņu valodas resursu kopa teksta semantiskai analīzei un sintēzei latviešu valodā" (2016–2019) (sk. arī 3.2.1. sadaļu) laikā korpuss sasniedza 10 tūkstošus teikumu. Pēc tam LVTB kvantitatīva un kvalitatīva paplašināšana notiek Valsts pētījumu programmās "Humanitāro zinātņu digitālie resursi" (2020–2022) un "Letonika latviskas un eiropeiskas sabiedrības attīstībai" (2022–2024), un darba rakstīšanas brīdī korpuss ir sasniedzis 17 tūkstošu teikumu apjomu. LVTB ir balstīti dažādi lingvistiski pētījumi (piemēram, Lokmane, Saulīte, 2023a; Lokmane, Saulīte 2023b, Rituma et al., 2023), tas palīdz pierādīt darba 1. hipotēzi, jo hibrīdais marķēšanas modelis labāk atbilst latviešu valodniecības tradīcijai un tāpēc ir vieglāk uztverams pētniekiem Latvijā.

Aptuvenu ieskatu par iegūtā korpusa apjoma nozīmīgumu var gūt, aplūkojot gan vēsturiski svarīgus korpusus, gan dažādu valodu sintaktiski marķēto korpusu izmērus Universālo atkarību (*Universal Dependencies*, UD) iniciatīvas (sk. 3. nodaļu detalizētam UD aprakstam) 2022. gada pavasara laidienā [14]. Šajā UD laidienā UDLV-LVTB ir sasniedzis 282 tūkstošus tekstvienību, un laidienā ir 32 korpusi, kam apjoms ir vismaz 250 tūkstoši sintaktisko vārdu [15] katrā. Ja valodām, kam UD ir vairāki korpusi, skaita kopējo korpusu apjomu, tad no laidienā iekļautajām 130 valodām 30 ir vismaz 250 tūkstoši sintaktisko vārdu lieli korpusi. Lielākie individuālie UD

[14] UD v2.10, pieejams LINDAT repozitorijā http://hdl.handle.net/11234/1-4758

[15] UD marķēšanas paradigma paredz noteiktām valodām noteiktas tekstvienības dalīt vairākos sintaktiskajos vārdos, piemēram, angļu saīsinājumi *mom's* un *don't* tiek uzskatīti katrs par diviem sintaktiskajiem vārdiem, tādēļ dažu valodu korpusos šādu "vārdu" skaits pārsniedz tekstvienību skaitu. Latviešu valodas korpusam šie skaitļi sakrīt.

korpusi šajā versijā ir vācu valodai – *UD_German-HDT* (3,455 milj. sint. v.), krievu valodai – *UD_Russian-SynTagRus* (1,517 milj.), čehu valodai – *UD_Czech-PDT* (1,509 milj.), japāņu valodai – *UD_Japanese-BCCWJ* (1,253 milj.) un *UD_Japanese-BCCWJLUW* (995 tūkst.), kā arī islandiešu valodai – *UD_Icelandic-IcePaHC* (985 tūkst.).

2010. gadā, kad tika uzsākti ar promocijas darbu saistītie pētījumi, pasaules lielākos korpusus salīdzināt bija daudz grūtāk, jo tie marķēti atbilstoši dažādiem formālajiem gramatikas modeļiem (t.sk. mazliet atšķīrās arī dalīšana tekstvienībās), taču trīs nozīmīgākie bija jau pieminētais čehu valodas PDT (1,5 milj.) (Hajič et al., 2000), kā arī angļu *Penn Treebank* (3 milj., sintaktiski marķēto korpusu pētījumu pamatlicējs) (Marcus et al., 1994) un vācu *Tiger Treebank* (900 tūkst.) (Brants et al., 2002). No vienas puses, arī citu valodu pieredze liecina, ka vajadzība pēc tik apjomīgiem korpusiem ir mazinājusies. Ticamākie iemesli tam ir dziļo neironu tīklu (angl. *deep neural networks*) valodas modeļu straujā attīstība un arvien plašāks starpvalodu mašīnmācīšanās metožu (angl. *transfer learning*) lietojums, kas ļauj mērķvalodas modelēšanu pilnveidot, izmantojot citu valodu datus. No otras puses, ņemot vērā pasaules pieredzi šobrīd aktīvākajā starptautiskajā datorlingvistikas sintakses kopienā UD, latviešu valodai ir izveidots zinātniski vērtīgs piemērota lieluma resurss, par ko liecina arī rezultāti parsētāju būvēšanas sacensībās (sk. 0. sadaļu).

2.3 Korpusa nozīme un ietekme

Šajā sadaļā aprakstīta LVTB publiskošana un tālākā izmantošana parsētāju būvēšanas pētījumos.

2.3.1 Pieejamība

Korpusa dati tiek marķēti publiski nepieejamā repozitorijā, bet korpuss tiek publiskots versiju veidā. Kopš 2018. gada korpusa versijas tiek publicētas Kārļa Universitātes LINDAT/CLARIN repozitorijā reizi pusgadā atbilstoši UD korpusu versiju grafikam kopā ar 3. nodaļā aprakstīto atvasināto latviešu valodas UD korpusu[16]. Tur tiek nodrošināta korpusa citējamība ar paliekošām saitēm (angl. *persistent linking*), kā arī tiešsaistes meklēšana korpusā, tādējādi nodrošinot šo korpusu derīgumu zinātniskajai citēšanai. Pirmajā publicētajā korpusa versijā ir 7,7 tūkstoši teikumu,

[16] Visas publicētās LVTB versijas kopā ar 3. nodaļā aprakstītā UD korpusa versijām pieejamas repozitorija sadaļā https://lindat.mff.cuni.cz/services/pmltq/#!/treebanks, atlasot pēc latviešu valodas.

jaunākajā (2022. gada novembrī) – gandrīz 17 tūkstoši teikumu [17]. Lai informētu Latvijas pētniekus par šo resursu, ir organizēti arī informatīvie semināri[18].

2.3.2 Korpusā balstīti parsētāju pētījumi

Ņemot vērā UD korpusa straujo attīstību un plašo lietojumu (sk. 3. nodaļu), šajā sadaļā iekļauto pētījumu aprakstam ir vairāk retrospektīva nozīme, jo UD kā vienojošā reprezentācija ir pavērusi iespējas daudzām valodām izmantot vienotus rīkus un vienotas parsētāju būves metodes, neveicot specifisku pielāgošanos (angl. *feature extraction*) katrai valodai. Tāpat pētniecības jomas attīstības rezultātā ir radušās jaunas parsētāju būves tehnoloģijas un latviešu valodas parsētāju izstrādē ir iesaistījušās arī starptautiskas pētnieku grupas (sk. 0. sadaļu).

Šeit minētajiem pētījumiem bija nozīmīga loma tālāk aprakstītā UD korpusa tapšanā: tika risināta korpusa transformāciju problemātika, kā arī gūts sākotnējais priekšstats par atkarību attēlojuma ietekmi uz parsētāju darbību un tādu rīku darbību, kas lieto parsētājus (*downstream application*). Turklāt daudzveidīgie atkarību attēlojumi, ko ir iespējams iegūt no LVTB ar transformāciju palīdzību, apstiprina darba 1. hipotēzi, ka hibrīdais modelis nodrošinās plašāku korpusa izmantojamību nekā atkarību modelis. Datu transformācijas, kas šajos pētījumos tika atzītas par perspektīvām, daudzos aspektos ir līdzīgas UD marķējuma modelim, tādējādi iezīmējot UD korpusa izveidi kā loģisku darba turpināšanas virzienu.

Tajā pašā laikā jāpiezīmē, ka jaunākās pētniecības tendences darba pabeigšanas brīdī liecina par hibrīdo modeļu reaktualizāciju. Nivre et al. (2022) aplūko plašu tipoloģiski dažādu valodu kopu (tā ietver arī 3. nodaļā aprakstīto latviešu valodas UD korpusu), t.sk. atsaucoties uz publikāciju (Bārzdiņš et al. 2007), kas kalpojusi arī par pamatu 2.1. sadaļā aprakstītajam modelim, un secina, ka L. Tenjēra *nucleus* elementiem līdzīgu konstrukciju izmantošana parsētāju uzbūvē sniedz nelielu, taču statistiski nozīmīgu uzlabojumu parsēšanas precizitātē.

Turpmākā 2.3.2. sadaļa strukturēta šādi: salīdzinošie parsētāju pētījumi aprakstīti 2.3.2.4. sadaļā, savukārt priekšnosacījumi šiem pētījumiem skaidroti 2.3.2.1.–2.3.2.3. sadaļās: vispirms 2.3.2.1. sadaļā raksturota pētījumiem nepieciešamā parsētāja sistēmas izvēle (atbilstoša tā

[17] Apkopojums par korpusa versijām un izmantotā marķējuma apraksti pieejami arī LU MII resursā http://sintakse.korpuss.lv/versions.html

[18] Piemēram, CLARIN iniciatīvā https://www.clarin.lv/lv/clarin-latvija-seminari/31-clarin-seminars-par-latviesu-valodas-sintaktiski-marketa-korpusa-izmantosanu

brīža modernākajām metodēm), tad 2.3.2.2. sadaļā definētas metrikas rezultātu salīdzināšanai un, visbeidzot, 2.3.2.3. sadaļā aprakstītas datu transformācijas, kas nepieciešamas, lai lietotu izvēlēto parsētāju sistēmu.

2.3.2.1 Parsētāja apmācības sistēmas izvēle

Veidojot no datiem inducētu parsētāju, bija nepieciešams saskaņot datu marķējuma modeli ar parsētājā izmantoto. Tā kā LVTB modelim kā hibrīdmodelim atbilstoša parsētāju apmācības sistēma nav izstrādāta (jo šāda veida sintakses modeļi līdz šim ir maz lietoti), tika apskatītas iespējas atbilstoši hibrīdmodelim marķētos datus pārveidot tā, lai tie būtu izmantojami kādai jau izstrādātai parsētāju apmācības sistēmai.

Atbilstoši marķēšanā plaši izmantotajiem modeļiem arī parsētāju apmācības sistēmās pamatā izmanto vai nu frāzes struktūras gramatiku, vai atkarību gramatiku. Lai gan parsētāju sistēmas, kas izmanto frāzes struktūras gramatiku (Collins, 2002), bija pazīstamas jau senāk, šīs pētījuma fāzes sākšanas laikā (2012. gads) labākos rezultātus sniedza atkarību gramatikā balstītais parsētājs (Bohnet, Nivre, 2012), kas vienlaicīgi veic sintaktisko un morfoloģisko analīzi.

Sākot šo pētījuma fāzi, latviešu valodas manuāli sintaktiski marķētais korpuss vēl bija salīdzinoši mazs (53 225 tekstvienību) un atsevišķi bija pieejams lielāks manuāli morfoloģiski marķētais korpuss (109 311 tekstvienību, ieskaitot sintaktiski marķēto korpusu). Tāpēc tika nolemts sākotnēji veikt morfoloģisko un sintaktisko marķēšanu secīgi, nevis reizē – lai varētu iegūt būtiski augstākas precizitātes morfoloģisko marķējumu, jo morfoloģiskā marķētāja apmācībai būtu iespējams izmantot gandrīz trīs reizes vairāk datu. Labāka morfoloģiskās marķēšanas precizitāte ļauj sasniegt arī labāku sintaktiskās marķēšanas precizitāti.

Atkarību modelī balstītās parsētāju sistēmas tika atzītas par piemērotākām ne tikai tam laikam augstās precizitātes dēļ, bet arī tādēļ, ka atsevišķas atkarību modelī balstītās sistēmas realizē neprojektīvu koku apstrādi ar tam īpaši paredzētu algoritmu (Nivre, 2009). Šis aspekts latviešu valodas gadījumā ir svarīgs tāpēc, ka valodām ar salīdzinoši brīvu vārdu secību neprojektīvi koki veidojas biežāk (McDonald et al., 2005). Kā alternatīva apskatāma iespēja lietot atkarību parsētāju, kas apstrādā tikai projektīvus kokus, bet tad jāizmanto projektivizācijas transformācijas datu pirmsapstrādei un pēcapstrādei (Nivre, Nilsson, 2005).

Atkarību modelī balstītajās parsētāju sistēmās parsēšanai tobrīd tika izmantotas divas pamatmetodes – pāreju parsētāji (angl. *transition based parser*) (Nivre, 2009) un grafu parsētāji (angl. *graph based parser*) (Koo, Collins, 2010), taču grafu parsētāju galvenais konceptuālais trūkums šo pētījumu kontekstā ir nespēja konstruēt neprojektīvus kokus.

Pāreju parsētājiem ir arī praktiskas priekšrocības. Pirmkārt, ātrdarbība: grafu parsētājam asimptotiskais izpildes laika novērtējums pret teikuma garumu ir $O(n^3)$ vai $O(n^4)$ atkarībā no izmantoto apakšstruktūru sarežģītības (Koo, Collins, 2010), bet pāreju parsētājam – $O(n)$ projektīvam (Nivre, 2009) un $O(n^2)$ neprojektīvam (Nivre, 2003). Otrkārt, plaši lietotajai pāreju parsētāja sistēmai *MaltParser*[19] (Nivre et al., 2007b) ir izstrādāts parametru optimizēšanas rīks *MaltOptimizer*[20] (Ballesteros, Nivre, 2012), kas automatizē mašīnmācīšanās parametru pārlasi, tādējādi samazinot nepieciešamo darba apjomu, kas jāveic, lai veiksmīgi apmācītu sistēmu.

2.3.2.2 Parsētāju vērtēšanas metrikas

Darbā parsētāju salīdzināšanai un izvērtēšanai tiek lietotas šādas metrikas.

Nemarķētas piesaistes kritērijs (angl. *unlabeled attachment score*, UAS) norāda, kādai daļai tekstvienību parsētāja norādītais vecāks kokā sakrīt ar doto standartu, piemēram, cilvēka marķēto.

Lomu precizitāte (angl. *label accuracy*, LA) norāda, kādai daļai tekstvienību norādītā loma (kā atkarības atkarīgajam elementam) sakrīt ar doto standartu, piemēram, cilvēka marķēto.

Marķētās piesaistes kritērijs (angl. *labeled attachment score*, LAS) norāda, kādai daļai tekstvienību ar doto standartu sakrīt gan parsētāja norādītais vecāks, gan loma (kā atkarības atkarīgajam elementam).

Ja nav norādīts citādi, metriku vērtības šajā darbā tiek dotas procentos.

2.3.2.3 Datu transformācijas

Lai izmantotu LVTB datus *MaltParser* apmācībai, marķējumu bija nepieciešams pārveidot: gan no hibrīdā sintakses modeļa uz atkarību modeli, gan no PML XML formāta uz *CoNLL* tabulāro datu formātu[21]. Šai vajadzībai radītie rīki pieejami tiešsaistē[22]. Šajā sadaļā galvenā uzmanība tiks veltīta transformācijām no LVTB modeļa marķējuma uz atkarību marķējumu (Pretkalniņa, Rituma, 2014), jo datu formāta transformācijas ir relatīvi vienkāršas – no PML XML failiem jāizgūst nepieciešamie lauki un tie jāpieraksta tabulārā formātā atbilstoši *CoNLL* formāta prasībām.

---

[19] *MaltParser* mājaslapa http://www.maltparser.org/

[20] *MaltOptimizer* mājaslapa http://nil.fdi.ucm.es/maltoptimizer/install.html

[21] *CoNLL-X Shared Task: Multi-lingual Dependency Parsing* mājaslapas *Wayback Machine* arhīvs https://web.archive.org/web/20160814191537/http://ilk.uvt.nl/conll/#dataformat

[22] *GitHub* repozitorijs *CorporaTools* https://github.com/LUMII-AILab/CorporaTools

Lai pārveidotu korpusa marķējumu, pietiek definēt, kā pārveidojamas šīs LVTB modeļa pamata konstrukcijas:

- atkarību attieksmes starp tekstvienībām;
- atkarību attieksmes, kurās vecāks un/vai bērns ir frāzes tipa konstrukcija;
- frāzes tipa konstrukcijas.

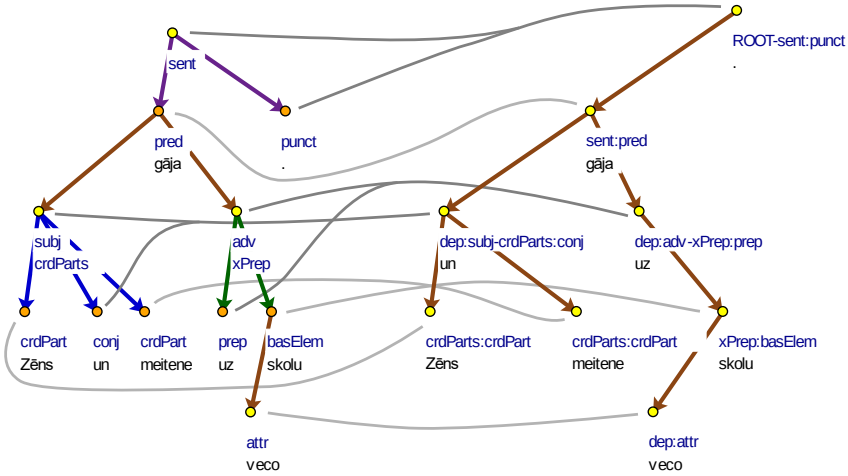
Izmantojot 2.1. sadaļā aprakstītās pamatidejas, tika nolemts izmantot transformēšanas stratēģiju, kas atbilst šādām prasībām:

- atkarību attieksmes starp tekstvienībām atainot identiski oriģinālam;
- frāzes tipa konstrukcijas atainot par atkarību koka fragmentiem, kas ir sakņoti koki;
- atkarību attieksmes, kurās vecāks vai bērns ir frāzes tipa konstrukcija, frāzes tipa konstrukciju(-as) aizstāt ar iepriekšējā punktā izveidoto koka fragmentu saknes virsotni.

Veicot transformāciju uz atkarību attēlojumu, informācija, kas vairs netiek attēlota strukturāli, ir jāzaudē vai jāiekodē lomās. Balansējot informācijas zaudējumus ar lomu komplekta sarežģītību, tiek izveidota šāda lomu kodēšanas sistēma:

- atkarību šķautņu marķējums tiek papildināts ar prefiksu, kas norāda, vai atkarības neatkarīgais elements oriģinālajā marķējumā ir tekstvienība (prefikss *dep*) vai frāzes tipa konstrukcija (prefikss *phdep*);
- elementam, kas kļūst par frāzes tipa konstrukcijas attēlojošā apakškoka sakni, tiek veidota salikta loma, kas sastāv no:
  1. prefiksa, kas norāda, vai attiecīgā konstrukcija oriģinālajā marķējumā ir atkarīga no tekstvienības (prefikss *dep*) vai no frāzes tipa konstrukcijas (prefikss *phdep*), un atkarību lomas, kas frāzes tipa konstrukcijai piešķirta oriģinālajā marķējumā;
  2. frāzes tipa un elementa lomas frāzē (šī lomas daļa netiek iekļauta gadījumos, ja šim elementam loma, kas izveidojas 1. punktā, ir sintaktiski pieļaujama, arī elementam esot ārpus frāzes);
- pārējās frāzes sastāvdaļas tiek marķētas ar saliktu lomu, kas sastāv no frāzes tipa un elementa lomas frāzē;
- informācija par tukšajām virsotnēm, kas attēlo vārdu izlaidumus, tiek atmesta.

Transformācijas piemērs dots 4. attēlā. Šāds apraksts definē transformācijai izvēlētās vispārīgās īpašības, taču, lai iegūtu precīzi definētu transformāciju, ir jānorāda, kā transformējams katrs no korpusā izmantotajiem frāzes veida konstrukciju paveidiem.



4. attēls. Sintakses koka transformēšanas piemērs

Konsultējoties ar valodniekiem [23], lielākajai daļai frāzes veida konstrukciju tika viennozīmīgi definētas transformācijas, taču tika apzinātas arī atsevišķas frāzes veida konstrukciju grupas, kurās valodiskās zināšanas nesniedz viennozīmīgu lēmumu par optimālo transformāciju. LVTB ir četras šādas grupas.

- Saliktie izteicēji (sintaktiski marķētajā korpusā apzīmēti ar x-vārda paveidu *xPred*):
  - darbības vārdu saliktie laiki, piemēram, *ir gājis*;
  - modālie izteicēji, piemēram, *varēja gulēt*, *gribēja ēst*, *gadījās pakrist*;
  - sastata izteicēji, piemēram, *ir gudrs*, *bija skolotājs*, *būs auksti*.
- Vienlīdzīgi teikuma locekļi un teikuma daļas (sintaktiski marķētajā korpusā – visi sakārtojuma konstrukciju paveidi).
- Frāzes veida konstrukcijas, kuru mērķis ir piesaistīt kokam interpunkcijas zīmes (sintaktiski marķētajā korpusā – visi pieturzīmju konstrukciju paveidi), piemēram:
  - palīgteikuma pieturzīmju konstrukcijas,
  - iespraudumu un iestarpinājumu pieturzīmju konstrukcijas,
  - uzrunas pieturzīmju konstrukcijas.

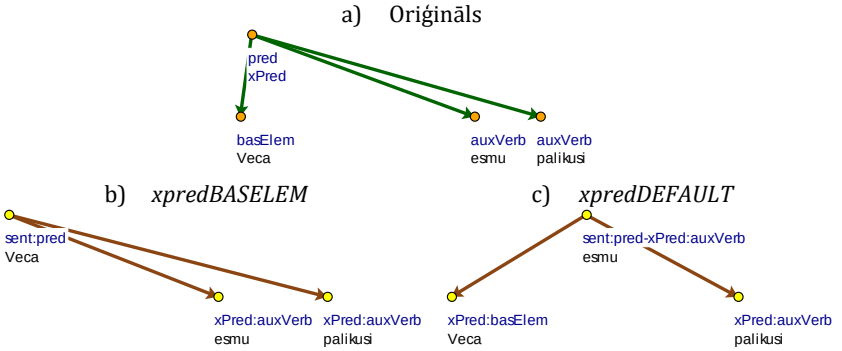
Šīm konstrukcijām tika izveidotas transformāciju alternatīvas, lai tālāk pētītu izvēļu ietekmi uz parsētāju izveidi un lietojamību.

[23] Īpašs paldies Laurai Ritumai, Baibai Saulītei, Guntai Nešporei-Bērzkalnei un Ilzei Lokmanei.

2.3.2.3.1 Salikto izteicēju transformācijas

Katrā saliktā izteicējā ir viens pamatelements (korpusā loma *basElem*), kas apzīmē semantiski galveno elementu, un viens vai vairāki palīgdarbības vārdi (korpusā loma *auxVerb*) un/vai modificētāji (korpusā loma *mod*). Piemēram, frāzē *ir skolotājs* lietvārds *skolotājs* ir semantiski galvenais vārds un tādēļ pamatelements, savukārt *ir* ir palīgdarbības vārds, bet frāzē *peldēt gribi* pamatelements ir darbības vārds *peldēt*. Ņemot vērā, ka konstrukcijā ir iespējamas šādas un tikai šādas sastāvdaļas, tika izvēlēts izskatīt šādas transformācijas (sk. 5. attēlu):

- *xpredBASELEM*[24] – par frāzei atbilstošā apakškoka sakni tiek izvēlēts elements ar lomu *basElem*; savukārt pārējie frāzes elementi (korekta koka gadījumā – ar lomām *auxVerb*, *mod*) tiek pakārtoti izvēlētajam saknes elementam (sk. 5.b attēlu);
- *xpredDEFAULT* – par frāzei atbilstošā apakškoka sakni tiek izvēlēts lineāri pirmais (t.i., pirmais attiecībā pret tekstvienību secību tekstā) elements ar lomu *auxVerb* vai *mod*, pārējie tiek pakārtoti izvēlētajam saknes elementam (sk. 5.c attēlu).



5. attēls. Saliktu izteicēju transformāciju piemēri,
apzīmējumi: zaļās šķautnes – x-vārda (frāzes) sastāvdaļas, brūnās šķautnes – atkarības

Transformāciju *xpredDEFAULT* būtu iespējams uzlabot, pilnveidojot kritērijus, pēc kuriem vairāku *auxVerb* un/vai *mod* lomu gadījumā izvēlas saknes elementu, jo lineāri pirmā elementa izvēle nenodrošina, ka frāzes elements tiks izvēlēts tā, lai attēlotu vienas un tās pašas attieksmes, piemēram, frāzē *esmu palikusi veca* tiktu izvēlēts *esmu*, bet frāzē *veca*

[24] Šeit un turpmāk – transformāciju nosaukumi saskaņoti ar publikācijās un *CorporaTools* repozitorijā izmantotajiem.

palikusi esmu – *palikusi*. Tomēr to šajā pētījuma fāzē nebija iespējams veikt bez sintaktiski marķētā korpusa manuālas papildināšanas ar papildu marķējumu, kas palīdzētu veikt šādu izvēli.

2.3.2.3.2 Sakārtojuma konstrukciju transformācijas

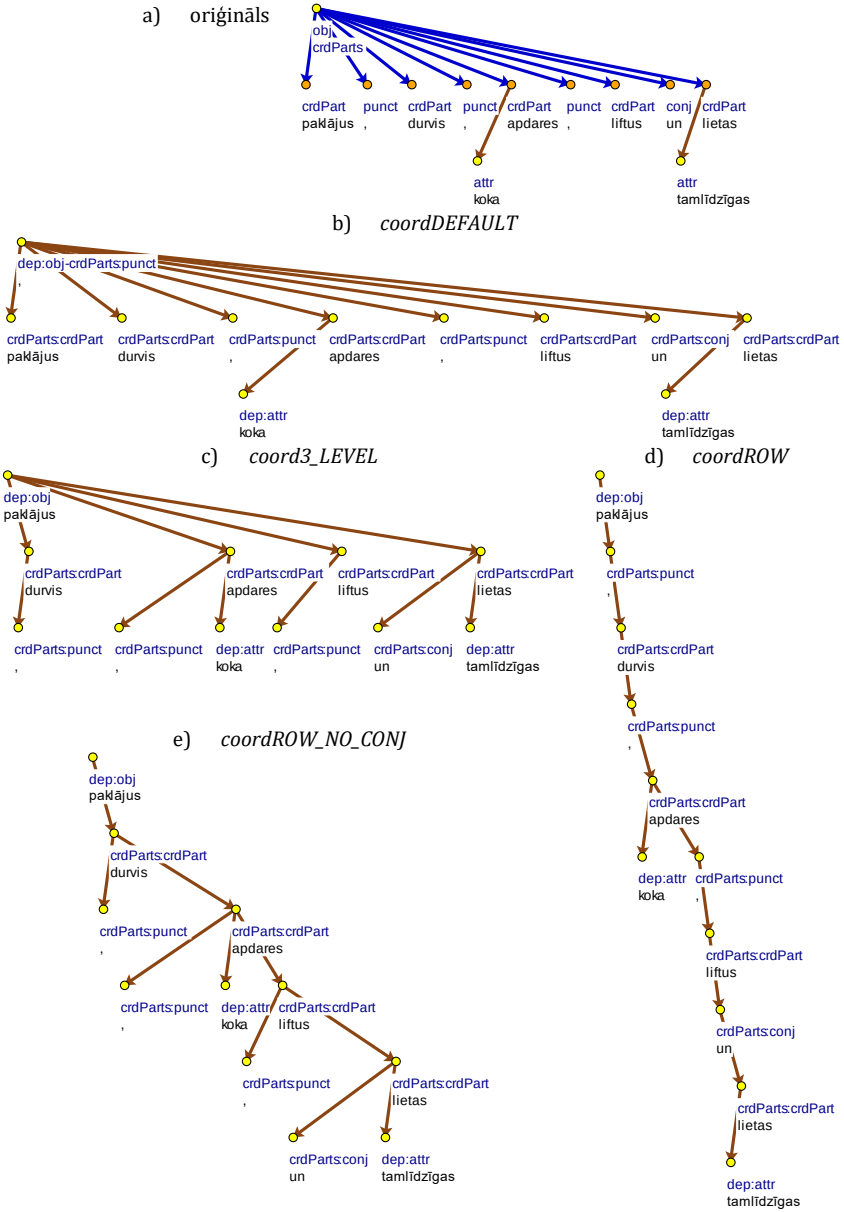
Sakārtojuma konstrukcijas sastāv no diviem vai vairāk koordinētajiem elementiem (korpusā ar lomu *crdPart*), kurus atdala pieturzīmes (korpusā ar lomu *punct*) un/vai saikļi (korpusā ar lomu *conj*). Piemēram, frāzē *sēdēja, ēda un dzēra* koordinētie elementi ir darbības vārdi *sēdēja*, *ēda*, *dzēra*, pēdējos divus koordinētos elementus atdala saiklis *un*, bet pirmos divus – pieturzīme komats. Saiklis var atrasties arī pirms pirmā koordinētā elementa, kā piemērā *gan zēni, gan meitenes*.

Sakārtojuma konstrukciju attēlojums plaši variē dažādos atkarību korpusos, atšķiroties gan pamata lēmumos, gan niansēs. Popel et al. (2013) piedāvā sakārtojuma konstrukciju attēlojumu iedalīt trijās saimēs atkarībā no koordinēto elementu konfigurācijas:

- Prāgas saime – visi koordinētie elementi ir bērni kādam no atdalošajiem saikļiem vai pieturzīmēm,
- Maskavas saime – koordinētie elementi veido atkarību virkni,
- Stenfordas saime – pārējie koordinētie elementi ir pakārtoti pirmajam vai pēdējam koordinētajam elementam.

Tika izveidotas transformācijas (sk. 6. attēlu), kuru rezultātā iegūtie attēlojumi pārstāv katru no šīm saimēm.

- *coord3_LEVEL* – Stenfordas saimes pārstāve – par frāzei atbilstošā apakškoka sakni tiek izvēlēts lineāri pirmais elements ar lomu *crdPart*, kā tā bērni tiek pievienoti pārējie elementi ar lomām *crdPart*. *conj* un *punct* elementi tiek pievienoti kā bērni tuvākajam *crdPart* elementam, pirms kura tie teikumā atrodas (sk. 6.c attēlu).
- *coordDEFAULT* – Prāgas saimes pārstāve – par frāzei atbilstošā apakškoka sakni tiek izvēlēts saiklis, kas atrodas starp lineāri pirmo elementu ar lomu *crdPart* un otro. Ja tur šāda saikļa nav, kā, piemēram, frāzē *zēni, meitenes un suņi*, tad tiek izvēlēta starp lineāri pirmajiem diviem elementiem ar lomu *crdPart* esošā pieturzīme. Visi pārējie sakārtojuma konstrukcijas elementi tiek pievienoti kā bērni izvēlētajai apakškoka saknei (sk. 6.b attēlu).
- *coordROW_NO_CONJ* – Maskavas saimes pārstāve – par frāzei atbilstošā apakškoka sakni tiek izvēlēts lineāri pirmais elements ar lomu *crdPart*. Katrs nākamais elements ar lomu *crdPart* tiek pievienots kā bērns iepriekšējam. Elementi ar lomām *conj* un *punct* tiek pievienoti kā bērni tuvākajam elementam ar lomu *crdPart,* pirms kura tie teikumā atrodas (sk. 6.e attēlu).



6. attēls. Sakārtojuma konstrukcijas transformāciju piemēri, apzīmējumi: zilās šķautnes – sakārtojuma konstrukcijas (frāzes) sastāvdaļas, brūnās šķautnes – atkarības

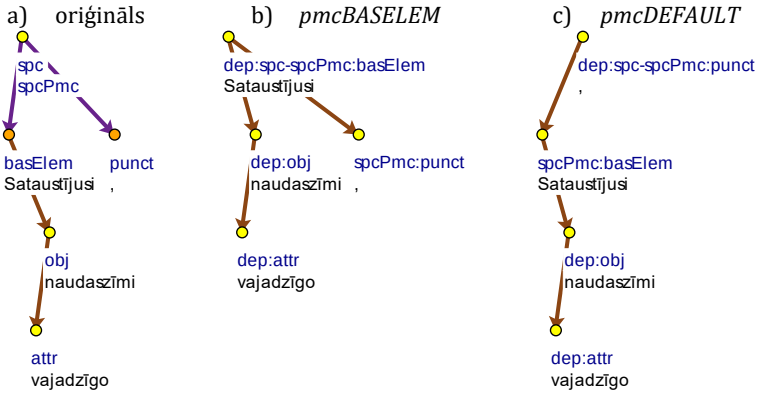
- *coordROW* – Maskavas saimes pārstāve – par frāzei atbilstošā apakškoka sakni tiek izvēlēts lineāri pirmais elements ar lomu *crdPart*. Katrs nākamais frāzes elements tiek pievienots kā bērns iepriekšējam. Ja pirms pirmā elementa ar lomu *crdPart* atrodas elements ar lomu *conj*, to pievieno pirmajam elementam ar lomu *crdPart* (sk. 6.d attēlu).

2.3.2.3.3 Pieturzīmju konstrukciju transformācijas

Katrā pieturzīmju konstrukcijā ir viens pamatelements (korpusā loma *basElem*), kurš kopā ar elementiem, kas apakškokā atrodas zem tā, izraisa attiecīgo pieturzīmju lietojumu, un viena vai vairākas pieturzīmes (korpusā loma *punct*). Vēl šajā konstrukcijā var ietilpt saikļi (korpusā loma *conj*), piemēram, gadījumos, kad ar pieturzīmēm atdalītu palīgteikumu ievada saiklis, kā arī patvaļīgs skaits elementu bez skaidri definētas sintaktiskās lomas (korpusā *no*), piemēram, uzrunas, iespraudumi, partikulas. Piemēram, teikumā *Bet, Anna, aizver durvis!* teikuma pieturzīmju konstrukcija sastāv no izsaukuma zīmes, saikļa *bet*, pamatelementa *aizver* un elementa ar lomu *no*, kas tālāk satur citu pieturzīmju konstrukciju (uzrunu) ar pamatelementu *Anna*.

Ņemot vērā, ka konstrukcijā būtiskākās sastāvdaļas ir *basElem*, *punct* un *conj* (ja tāds ir), tika izvēlēts izskatīt šādas transformācijas (sk. 7. attēlu):

- *pmcBASELEM* – par frāzei atbilstošā apakškoka sakni tiek izvēlēts elements ar lomu *basElem*; savukārt pārējie frāzes elementi tiek pakārtoti izvēlētajam saknes elementam (sk. 7.b attēlu);



7. attēls. Pieturzīmju konstrukciju transformāciju piemēri, apzīmējumi: violetās šķautnes – pieturzīmju konstrukcijas (frāzes) sastāvdaļas, brūnās šķautnes – atkarības

- *pmcDEFAULT* – par frāzei atbilstošā apakškoka sakni tiek izvēlēts lineāri pirmais elements ar lomu *conj* vai, ja tāda nav, tad pirmais elements ar lomu *punct*, pārējie tiek pakārtoti izvēlētajam saknes elementam (sk. 7.c attēlu).

2.3.2.3.4 Kopsavilkums

Katra frāzes transformācija ir neatkarīga no citām, tāpēc izskatītos frāžu transformāciju variantus ir iespējams patvaļīgi kombinēt. Tādējādi ir iespējams iegūt 16 dažādas transformācijas, ko izmantot sintaktiski marķētā korpusa datu pārveidošanai, lai tos tālāk padarītu izmantojamus atkarību parsētāja apmācībai. Izmantojot šīs transformācijas, tiek iegūti 16 korpusa varianti, kuros tie paši teksti marķēti dažādos veidos. Taču, lai noskaidrotu iegūto korpusa variantu savstarpējās priekšrocības, bija nepieciešami praktiski eksperimenti, jo nebija pietiekamu teorētisko lingvistisko apsvērumu, kas dotu skaidru atbildi.

2.3.2.4 Parsētāja apmācība un transformāciju salīdzināšana

Pirmie parsētāju apmācības eksperimenti tika veikti, manuāli salīdzinot *MaltParser* sistēmā realizētos parsēšanas algoritmus (Pretkalniņa, Rituma, 2013), taču tālākajai sistēmas parametru atlasei izmantota *MaltOptimizer* sistēma (Ballesteros, Nivre, 2012), kas triju soļu procesā veic piemērotāko parametru atlasi. Pirmajā solī *MaltOptimizer* aprēķina vispārējus rādītājus par korpusu, piemēram, tekstvienību skaitu, neprojektīvo koku īpatsvaru. Otrajā solī tiek izvēlēts parsēšanas algoritms. Trešajā solī atbilstoši parsēšanas algoritmam tiek izvēlētas apmācības pazīmes – noteiktām tekstvienībām var lietot vārdformu, pamatformu, morfoloģisko informāciju un *CoNLL* tabulārajā formātā paredzēto pazīmju (*FEATS*) kolonnu, kas ļauj iekļaut jebkādu papildinformāciju.

Eksperimenti tika veikti, izmantojot gan korpusā esošo manuāli veidoto morfoloģisko marķējumu, gan automātiski veidotu morfoloģisko marķējumu. Salīdzinošajos eksperimentos uzsvars likts uz automātiski veidota morfoloģiskā marķējuma izmantošanu, jo tas labāk atbilst parsētāja iespējamajiem lietojumiem – manuāla morfoloģiskā marķēšana prasa laiku un cilvēkresursus, tāpēc parsētājus visplašāk lieto tekstam ar automātiski iegūtu morfoloģisko marķējumu. Automātiski marķētās morfoloģijas izmantošana radīja nelielu precizitātes kritumu – ap 1–2 procentpunktiem (Pretkalniņa, Rituma, 2013).

Lai iegūtu labāku izpratni par izveidoto sintaktiskā marķējuma transformāciju ietekmi uz parsēšanu, tika veikti divu tipu eksperimenti. Parsētāju ārējā novērtēšana (angl. *extrinsic evaluation*) tika veikta, vadoties

pēc rezultātiem, ko sniedz rīki, kas izmanto parsētājus. Parsētāju iekšējā novērtēšana (angl. *intrinsic evaluation*) tika veikta, salīdzinot parsēšanas precizitāti fiksētām testa datu kopām un atsevišķām konstrukcijām tajās.

2.3.2.4.1 Ārējais novērtējums

Lai gan sintaktiskam parsētājam ir plaši lietojumi lingvistikas pētījumu sagatavošanā un datu indeksācijā, tomēr plašākais parsētāju lietojums ir saistāms ar integrēšanu dažādās rīku sistēmās, kurās sintaktiskās analīzes rezultāti tiek izmantoti kā ieejas dati citiem rīkiem. Tāpēc ir būtiski izvērtēt ne tikai to, kā sintaktiskā reprezentācija ietekmē parsēšanas precizitāti, bet arī to, kā tā ietekmē to rīku precizitāti, kas izmanto parsēšanas rezultātus. Šai vajadzībai tika veikti detalizēti eksperimenti ar 16 aprakstītajām transformācijām un ar trim latviešu valodai izveidotiem rīkiem, kas darbojas dažādos semantiskās analīzes aspektos: semantisko lomu marķētāju, koreferenču risinātāju un nosaukto entitāšu atpazinēju.

Semantisko lomu atpazīšana (angl. *semantic role labeling*) (piem., Barzdins et al., 2014) ir semantiskās analīzes uzdevums, kas attiecīgajā tobrīd latviešu valodai izveidotajā rīkā darbojas teikuma līmenī. Šajā rīkā tiek lietota *FrameNet*[25] (Ruppenhofer, 2010) metode – atpazīt galīgu skaitu situāciju (angl. *semantic frame*), ko veido situāciju izsaucēji (*frame target*) un situāciju elementi (angl. *frame element*). Izmantotais latviešu valodas rīks paredzēts 26 ziņu tekstiem raksturīgām situācijām, piemēram, *dzimšana*, *pirkšana/pārdošana*, *stāšanās amatā*, *uzņēmuma dibināšana,* un atpazīšanu veic divās fāzēs, vispirms atrodot situāciju izsaucējus, pēc kuriem tiek identificētas situācijas, un tad identificētajām situācijām meklējot iespējamo elementu realizācijas. Piemēram, ja teikumā *Uzņēmums "Laima" iegādājies ražošanas tehniku 30 tūkst. eiro apjomā* šādi jāmarķē semantiskās lomas, tad vispirms var atrast, ka vārds *iegādājies* izsauc *pirkšanas/pārdošanas* situāciju, un pēc tam – ka *pirkšanas/pārdošanas* situācijai ir pieļaujams elements *pircējs* (piemērā – *uzņēmums "Laima"*) un elements *pirkums* (*ražošanas tehnika*).

Semantisko lomu marķēšanas rīks pirmajā fāzē kā pazīmes izmantoja atkarību koka lomas, savukārt otrajā fāzē – gan lomas, gan sintaktiskā koka strukturālās īpatnības. Tāpat atkarību koka struktūra nosaka situāciju elementu robežas, ja situācijas elementu apraksta vairāki vārdi. Aplūkotajā piemērā elements *pirkums* būtu *ražošanas tehniku 30 tūkst. eiro apjomā*, marķēšanas rīks kā situācijas elementu atzīmētu *tehniku* un pārējie vārdi kā

[25] *FrameNet* projekta mājaslapa https://framenet.icsi.berkeley.edu

situācijas elementa sastāvdaļas ir identificējami kā vārda *tehniku* bērni vai pēcteči atkarību kokā.

Koreferenču risināšana (angl. *coreference resolution*) (piem., Znotins, Paikens, 2014) ir semantiskās analīzes uzdevums, kas risināms teksta ietvaros. Šī uzdevuma mērķis ir identificēt teikuma fragmentus – pieminējumus (angl. *mention*), kas atsaucas uz vienu personu vai lietu (referentu). Piemēram, tekstā *Jānis ir centīgs jaunietis, viņš vienmēr izpilda mājasdarbus* koreferenču risināšana būtu identificēt, ka *Jānis* un *viņš* ir viena referenta dažādi pieminējumi. Koreferenču risināšanā kā pazīmes tiek izmantotas atkarību struktūras īpatnības.

Nosaukto entitāšu atpazīšana (angl. *named entity recognition*) (piem., Paikens et al., 2012; Znotins, Paikens, 2014) ir semantiskās analīzes uzdevums, kas risināms gan teikuma, gan teksta ietvaros. Šī uzdevuma mērķis ir identificēt vietvārdus, personu vārdus, uzņēmumu nosaukumus un citas nosauktās entitātes (angl. *named entity*), kas iekļautas attiecīgās sistēmas tvērumā[26]. Nosaukto entitāšu atpazīšana latviešu valodai pamatā izmanto šablonus un nosaukumu sarakstus, taču eksperimentu vajadzībām tā tika papildināta ar sintaktisko pazīmju lietojumu – atkarību lomām un atsevišķām atkarību struktūras īpatnībām (piemēram, apskatāmajai virsotnei tuvākais priekštecis atkarību kokā, kurš ir lietvārds).

Eksperimentu **rezultāti** publicēti Pretkalniņa et al. (2014). Aplūkojot vairāku semantikas rīku rezultātus, tika secināts, ka dažādiem semantikas uzdevumiem piemērotākās atkarību reprezentācijas var atšķirties.

- Nosaukto entitāšu atpazīšanai sintaktisko pazīmju izmantošana nedeva manāmu rezultātu precizitātes uzlabojumu, tāpēc ir iespējams to lietot rīkplūsmās (angl. *pipeline*) pat pirms parsētāja.
- Koreferenču risināšana vislabākos rezultātus deva, izmantojot parsētājus ar vienlīdzīgo teikuma locekļu atveidojumu *coordDEFAULT*, par spīti tam, ka šie parsētāji paši par sevi deva būtiski sliktākus sintaktiskās parsēšanas rezultātus.
- Semantisko lomu atpazīšanas uzdevumam sintaktiskās informācijas izmantošana deva būtisku uzlabojumu otrajā solī, t.i., meklējot situācijas elementus – šeit labākos rezultātus deva parsētājs ar *coordROW_NO_CONJ* vienlīdzīgo teikuma locekļu atveidojumu.

Tādējādi, lai iegūtu tā brīža resursiem optimālus semantisko rīku darbības rezultātus, nepietika izvēlēties parsētāju ar augstākajiem precizitātes rādītājiem pat tad, ja precizitātes rādītāji atšķiras par vairāk nekā pieciem procentpunktiem (~10%). Tas parāda, ka izvēle sintaktiski

[26] Latviešu valodai izstrādātais nosaukto entitāšu atpazinējs kā nosauktās entitātes neatpazīst datumus un laika identifikatorus, lai gan reizēm arī šāda informācija tiek iekļauta atpazīšanas problemātikā.

marķētajā korpusā izmantot LVTB hibrīdo marķējuma modeli ir ļoti veiksmīga – no šī attēlojuma viegli var iegūt dažādas atkarību reprezentācijas un tādējādi pielāgot katram parsētāju lietojumam vispiemērotāko atkarību struktūru.

Tajā pašā laikā jāņem vērā, ka rezultāti tika gūti no neliela datu korpusa un apmācības gaitā bija vērojamas lielas rezultātu svārstības – lai gan tie attiecīgajā pētījuma posmā sniedza būtisku ieskatu tālākajai pētījuma attīstībai, nav droši no tiem izdarīt secinājumus par lieliem korpusiem un citām parsēšanas metodēm.

2.3.2.4.2 Iekšējais novērtējums

Par atkarību reprezentācijas piemērotību dažādiem uzdevumiem var spriest ne tikai pēc precizitātes noteiktu uzdevumu veikšanā, bet arī pēc tā, cik precīzi parsētāju sistēma iemācās tādu vai citādu noteiktas sintaktiskās struktūras reprezentāciju. Tāpēc tika veikta otra eksperimentu sērija, kurā parsētāju rezultātos salīdzina precizitāti struktūrām ar neviennozīmīgu reprezentāciju – sakārtojuma konstrukcijām, saliktajiem izteicējiem un pieturzīmju konstrukcijām. Rezultāti publicēti (Pretkalniņa, Rituma, 2014).

Parsētāju novērtēšanai tika uztaisīta virkne eksperimentu un rezultāti tika salīdzināti, izmantojot LAS, UAS un LA metrikas trijos veidos:

1. parsētāja rezultātiem kopumā;
2. interesējošās frāzes sastāvdaļām šādām frāžu grupām (katrai grupai atsevišķi):
   - pieturzīmju konstrukcijām,
   - sakārtojuma konstrukcijām,
   - saliktiem izteicējiem;
3. frāžu atkarīgajiem šādām frāžu grupām (katrai grupai atsevišķi):
   - pieturzīmju konstrukcijām,
   - sakārtojuma konstrukcijām,
   - saliktiem izteicējiem.

Lai iegūtu uzskaitāmās tekstvienības 2. un 3. grupas metrikām, tika izmantots korpusa oriģinālais hibrīdais marķējums. Par frāzes sastāvdaļu tiek uzskatīta:

a. tekstvienība, kas hibrīdajā marķējumā ir marķēta kā frāzes sastāvdaļa,
b. tekstvienība, kas tad, ja frāzes sastāvdaļa pati ir frāze, pēc transformācijas uz atkarību reprezentāciju kļūst par apakšfrāzi reprezentējošā apakškoka sakni.

Par frāzes atkarīgo tiek uzskatīta:

a. tekstvienība, kas hibrīdajā marķējumā ir marķēta kā frāzes atkarīgais,

b. tekstvienība, kas tad, ja frāzes atkarīgais ir frāze, pēc transformācijas uz atkarību reprezentāciju kļūst par frāzi reprezentējošā apakškoka sakni.

Apkopojot sastāvdaļu atpazīšanas precizitāti dažādajiem frāzes veida konstrukciju atkarību attēlojumiem, tika novērots *xpredDEFAULT* un *pmcDEFAULT* parsētāju pārākums, kas apstiprināja arī vispārējos rezultātos novēroto tendenci. Analizējot sakārtojuma konstrukciju sastāvdaļu atpazīšanas rezultātus, bija novērojams, ka *coordDEFAULT* parsētāju rezultāti viennozīmīgi ir sliktāki. Šī tendence atspoguļojās arī vispārējos rezultātos. Gan sastāvdaļu atpazīšanas, gan vispārējās precizitātes testos redzams Maskavas saimes atkarību struktūru pārākums – labākos rezultātus sniedza *coordROW* un *coordROW_NO_CONJ*, taču rezultāti neapliecināja, ka kāds no šiem attēlojumiem būtu viennozīmīgi labāks nekā otrs.

Analizējot frāžu atkarīgo atpazīšanu, grūtības sagādāja pieturzīmju konstrukciju un sakārtojuma konstrukciju atkarīgo elementu mazais skaits korpusā – attiecīgi 2,5 un 1% tekstvienību[27], tāpēc šo atkarīgo atpazīšanas precizitāte bija zema. Par vienlīdzīgo teikuma locekļu konstrukcijām nācās secināt, ka korpusa apjoms tobrīd bija pārāk mazs, lai pilnvērtīgi iemācītos atšķirt sakārtojuma konstrukciju kopējos atkarīgos un viena koordinētā elementa atkarīgos, jo LA visiem sakārtojuma konstrukciju atkarīgo atpazīšanas veidiem bija ļoti zems (15–30%). Par pieturzīmju konstrukciju atkarīgajiem jāņem vērā, ka *pmcDEFAULT* gadījumā atbilstošā atkarību apakškoka sakne ir saiklis vai pieturzīme, bet *pmcBASELEM* apskatāmais koka struktūras fragments ir līdzīgs gadījumiem, kad pieturzīmju konstrukcijas pamatelements tiek lietots ar tādu pašu atkarīgo, bet ārpus pieturzīmju konstrukcijas. Tāpēc tika secināts, ka pietiekami liela korpusa gadījumā lietot *pmcDEFAULT* tipa attēlojumu teorētiski būtu informatīvāk, bet mazam korpusam (kā eksperimenta laikā pieejamajam) ir labāk lietot *pmcBASELEM*, ko parsētājs vieglāk iemācās.

Savukārt, analizējot salikto izteicēju atkarīgos, kuriem piemēru korpusā ir vairāk, atpazīšanas rezultāti demonstrē tendenci par labu *xpredDEFAULT*. Tas kopā ar sastāvdaļu atpazīšanas un vispārīgajiem

[27] Eksperimentu veikšanas brīdī pieejamie dati LVTB bija šādi: 15,3% tekstvienību ietilpa sakārtojuma konstrukcijās, 31,9% tekstvienību – pieturzīmju konstrukcijās un 7,1% – saliktajos izteicējos. Attiecīgo frāžu atkarīgie bija salīdzinoši reti – visā korpusā ir 1% sakārtojuma konstrukciju atkarīgo, 7% salikto izteicēju atkarīgo un 2,5% pieturzīmju konstrukciju atkarīgo (par pieturzīmju konstrukciju atkarīgajiem kļūst elementi, kas attiecināmi uz visu teikuma daļu, nevis uz atsevišķu teikuma locekli, parasti atbilstoši latviešu valodniecības teorijai tie ir situanti un determinanti, kā arī dažu tipu palīgteikumi).

rezultātiem skaidri liecina par labu *xpredDEFAULT* tipa attēlojumu izmantošanai.

Parsētāju varianti, kuriem par labu liecina atsevišķo valodas parādību analīze, ir starp labākajiem arī vispārīgajos testos. Salīdzinot ar parsētāju vispārīgajiem rezultātiem, kas iegūti iepriekšējā eksperimentu sērijā (sk. 2.3.2.4.1. sadaļu un (Pretkalniņa et al., 2014)), vērojams, ka korpusa mazā apjoma dēļ rezultātiem raksturīga diezgan augsta nestabilitāte, taču kopumā šie sākotnējie pētījumi deva vērtīgu ieskatu parsētāju būvēšanā un palīdzēja sagatavoties nākamajā nodaļā aprakstītajiem pētījumiem.

3 Latviešu valodas Universālo atkarību korpuss

2014. gadā, par pamatu ņemot *Universal Dependency Treebank* projektu (McDonald et al., 2013) un *HamleDT* projektu (Rosa et al., 2014), rodas Universālo atkarību (*Universal Dependencies*, UD) iniciatīva – Joakima Nivres (*Joakim Nivre*) koordinēts projekts ar mērķi piedāvāt valodneatkarīgu (angl. *language-independent*) ietvaru sintaktiski marķētu korpusu veidošanai[28]. UD iniciatīva par savu mērķi izvirza marķējumu ar augstu lingvistisko precizitāti, marķējuma saskaņotību dažādām valodām (angl. *cross-lingual consistence*), piemērotību ātrai manuālai un automātiskai marķēšanai, pieejamību (saprotamību) potenciālajiem datu izmantotājiem ārpus lingvistu loka, kā arī noderīgumu tālākai izmantošanai valodas sapratnes (angl. *natural language understanding*) rīkos (Nivre et. al., 2016). Šī mērķa sasniegšanai UD iniciatīva izstrādā vadlīnijas atkarībās balstītam marķējuma modelim, definē izmantojamās lomas un morfoloģiskās kategorijas, kā arī iespējas veidot valodspecifiskus (angl. *language-specific*) modeļa paplašinājumus un vadlīnijas. Reizi pusgadā publicējot jaunu datu versiju, iniciatīvā veidotais datu klāsts tiek strauji attīstīts: UD versijā 1.2 2015. gada novembrī ir iekļauti 37 korpusi 33 valodām, versijā 2.2 2018. gada jūlijā – 112 korpusi 71 valodai, versijā 2.11. 2022. gada novembrī – 243 korpusi 138 valodām[29].

Dalība UD iniciatīvā ar vadlīnijām atbilstoši marķētu latviešu valodas korpusu paplašina dažādas pētījumu un izmantojuma iespējas – modeļa valodneatkarība ļauj lietot latviešu valodas datus kopā ar citu valodu datiem salīdzinošajos pētījumos, kā arī izmantot dažādus citu pētnieku izstrādātus rīkus, kas paredzēti šim gramatikas modelim un datu formātam. Sintaktiski marķēts korpuss, kas izmantojams parsētāju apmācībai, ir būtisks šī darba mērķis, un šis mērķis salāgojas ar UD nolūku labi derēt ātrai automātiskai marķēšanai. Savukārt UD orientācija gan uz lingvistisko precizitāti, gan lietojamību valodas sapratnes uzdevumos atbilst 2.3.2.4. sadaļā fiksētajai nepieciešamībai sintaktisko reprezentāciju izvēlēties ne tikai tā, lai to būtu viegli iemācīties parsētājam, bet arī tā, lai parsētāja dotie rezultāti būtu pietiekami informatīvi tālākai izmantošanai. Tāpēc tika nolemts izpētīt iespējas veidot pielāgotu transformāciju, kas ļautu "Latviešu valodas sintaktiski marķēta korpusa" (*Latvian Treebank,* LVTB) datus publicēt UD iniciatīvas ietvaros.

[28] UD iniciatīvas mājaslapa https://universaldependencies.org/

[29] UD v2.11 pieejama LINDAT repozitorijā http://hdl.handle.net/11234/1-4923

3.1 Korpusa izveide

Pirmo reizi kopā ar citiem UD korpusiem korpuss latviešu valodai (*Latvian UD Treebank*, UDLV-LVTB) publicēts versijā 1.3 2016. gada maijā (Pretkalniņa et al., 2016). Sākotnējā versijā korpusā bija aptuveni tūkstoš teikumu (LVTB ziņu tekstu daļa), taču versijā 2.0 tiek publicēts transformēts viss tobrīd marķētais LVTB materiāls. 2022. gada novembra versijā 2.11 UDLV-LVTB korpusa apjoms ir sasniedzis 285 tūkstošus tekstvienību un 16,9 tūkstošus teikumu.

UDLV-LVTB satur šādu marķējumu (iekavās doti atbilstošo lauku nosaukumi UD izmantotajā *CoNLL-U* tabulārajā datu formātā):

- teksts ir sadalīts tekstvienībās un teikumos;
- katrai tekstvienībai ir norādīta pamatforma (*LEMMA* – vēlams, bet neobligāts lauks atbilstoši UD vadlīnijām);
- katrai tekstvienībai ir norādīta morfoloģiskā informācija – morfoloģiskais tags no oriģinālā marķējuma (*XPOS* – vēlams, bet neobligāts lauks), vārdšķiras identifikators atbilstoši UD specifikācijai (*UPOS* – obligāts lauks) un morfoloģisko pazīmju izvērsums atbilstoši UD specifikācijai (*FEATS* – vēlams, bet neobligāts lauks);
- teikuma ietvaros atkarību struktūra – katrai tekstvienībai norādīts tās vecāks atkarību kokā (*HEAD* – obligāts lauks) un atkarības tips jeb sintaktiskā loma (*DEPREL* – obligāts lauks);
- sākot ar versiju 2.1, teikuma ietvaros tiek norādīta arī paplašinātā atkarību struktūra, kas satur tālākai izmantošanai valodas sapratnes uzdevumos noderīgas papildu saites (DEPS – vēlams, bet neobligāts lauks).

Marķējums tiek iegūts, LVTB datus apstrādājot ar speciāli UD vajadzībām radītu transformāciju LVTB2UD. Tas ļauj viegli papildināt UD korpusu, kad LVTB tiek papildināts ar jauniem datiem. Transformācijas pirmkods ir pieejams tiešsaistē [30]. Pārejot no versijas 1.4 uz 2.0, UD vadlīnijās tiek veiktas būtiskas izmaiņas un precizējumi, un transformācija tiek atbilstoši atjaunināta. Turpmāk, ja nav norādīts citādi, tiek aprakstīta transformatora darbība atbilstoši jaunākās versijas (2.11) vadlīnijām.

3.1.1 UDLV-LVTB izveides transformācija

UDLV-LVTB ir likumos balstīta transformācija, kas balstās 2.3.2.3. sadaļā aprakstītajā pieredzē, taču atšķirībā no sākotnējām

[30] *GitHub* repozitorijā *CorporaTools* mape LVTB2UD https://github.com/LUMII-AILab/CorporaTools/tree/master/LVTB2UD

transformācijām šī nav parametriska, tā veidota kā likumu komplekss, kura mērķis ir maksimāli precīzi aprakstīt, kā katra no LVTB konstrukcijām (dažādu tipu frāzes, atkarības) pārveidojama UD konstrukcijā, ieskaitot gan atbilstošu lomu piekārtošanu, gan strukturālas izmaiņas.

LVTB2UD ieejas datus apstrādā teikumu pa teikumam, katram teikumam veicot tālāk aprakstītās apstrādes darbības. Katra teikuma apstrāde ir algoritmiski neatkarīga no iepriekšējo un tālāko teikumu apstrādes.

3.1.1.1 Dalījums tekstvienībās

Pirmais solis katra teikuma apstrādē ir izsecināt, vai LVTB dalījums tekstvienībās atbilst UD vadlīnijām, un, ja nē, tad radīt izmainīto dalījumu tekstvienībās. Atšķirības ir retas, un tās iedalāmas divās grupās: redakcionālu kļūdu labojumi un "vārdi" ar atstarpēm.

Redakcionālas kļūdas, piemēram, liekas garumzīmes (piem., *gribās*) un nepareizi kopā vai atsevišķi sarakstīti vārdi (piem., *jā dara*, *kautkas*), LVTB, pārņemot PDT praksi, tiek labotas vienā PML (sk. 2.2.1. sadaļu) līmenī ar morfoloģisko marķējumu (tajā pašā failā), bet sintaktiskais marķējums tiek veidots nākamajā līmenī (citā failā) jau izlabotajam tekstam. UD metodika savukārt paredz izmantot tekstu ar visām redakcionālajām kļūdām un lieki atdalītu vārdu (*jā dara*) savienošanai ievieš atkarību lomu *goeswith*. Tā kā LVTB satur gan informāciju par oriģinālo tekstu, gan par redakcionālā labojuma veidu, tad šajā solī tiek rekonstruēts oriģinālteksts un vajadzības gadījumā arī pievienotas *goeswith* saites. Par tādām kļūdām kā izlaisti komati, kam UD nespecificē norādīšanas veidu, tiek ievietoti kodificēti komentāri atbilstošās tekstvienības laukā *MISC*, kas paredzēts nespecificētas informācijas nodošanai. Taču nākotnē šis atainojums var mainīties, ja UD specificē vienotu veidu šādas informācijas norādīšanai.

Sākotnējās LVTB un UD versijās bija plaša nesaderība "vārdu" ar atstarpēm lietojumā – UD tādus nepieļāva vispār, savukārt LVTB tos lietoja daudzviet – ar cipariem rakstītu skaitļu (*10 000*), saliktu saikļu un partikulu (*lai gan*) un saīsinājumu (*u. c.*) attēlojumā. Taču līdz ar versiju 2.0 LVTB un UD pieejas kļuva daudz tuvākas: UD ieviesa vadlīnijās iespēju lietot "vārdus" ar atstarpēm ar nosacījumu, ka tie ir uzskaitīti valodspecifiskajā dokumentācijā ar regulārajām izteiksmēm, savukārt LVTB UD ietekmē atteicās no saliktu saikļu un partikulu attēlošanas par vienotām tekstvienībām. Tā rezultātā gan LVTB, gan UDLV-LVTB kā tekstvienības ar atstarpēm lieto tikai atsevišķus saīsinājumus (*P. S.*, *N. B.* un tādus saīsinājumus kā *u.c.*, *v.tml.*, ja tie pierakstīti ar atstarpi) un ar ciparem pierakstītus skaitļus, un šajā aspektā īpaša pārveidošana vairs nav nepieciešama.

3.1.1.2 Sākotnējā morfoloģiskā informācija

Nākamais solis ir aizpildīt morfoloģijas laukus *LEMMA*, *XPOS*, *UPOS* un *FEATS*. Laukus *XPOS* un *LEMMA* attiecīgi aizpilda ar LVTB doto morfoloģijas tagu un lemmu. Laukus *UPOS* un *FEATS* aizpilda atbilstoši valodnieku izstrādātiem likumiem, kas par ieejas datiem izmanto lemmu un tagu. Atsevišķos gadījumos šī informācija nav pietiekama, lai noteiktu *UPOS*:

- UD ir nepieciešams LVTB vietniekvārdu dalījums determinētājos (*DET*, vietniekvārds, kas aizstāj īpašības vārdu, *tā māja*) un vietniekvārdos (*PRON*, vietniekvārds, kas aizstāj lietvārdu, *tas ir viņš*), kamēr LVTB šādu dalījumu nedod, jo latviešu valodā tas bieži netiek izteikts ar morfoloģiskiem rādītājiem;
- UD prasa šķirt apstākļa vārdus, kas ievada palīgteikumus (*SCONJ*), no citiem apstākļa vārdiem (*ADV*), kamēr LVTB šādu dalījumu nedod, jo latviešu valodā tas netiek izteikts ar morfoloģiskiem rādītājiem;
- UD prasa iespraudumiem no citām valodām norādīt vārdšķiru, kādu tie aizstāj dotajā teikumā, savukārt LVTB šādus gadījumus marķē kā bezmorfoloģijas elementus.

Kolonnas *FEATS* aizpildījumu pamatā nosaka morfoloģiskajā tagā iekļautās pazīmes, taču atsevišķos gadījumos pazīmes piešķir, arī vadoties pēc lemmu uzskaitījuma, piemēram, vairumam īpašības vārdu pazīme *Poss* (*possesive*, piederība) netiek aizpildīta, taču to norāda vārdiem *manējais*, *tavējais*. UD arī piedāvā vairākas pazīmes un pazīmju vērtības, kas latviešu valodai nav saistošas, jo neparādās kā morfoloģiskas kategorijas, piemēram, *Animacy* vai *Case=Erg*. Lai labāk varētu izsecināt dažas pazīmes, tika papildināts arī LVTB izmantotais marķējums – divdabju marķējums tika papildināts ar pakāpes un nolieguma norādēm, tādējādi dalība UD iniciatīvā ietekmē arī hibrīdmodeļa attīstību.

Šis morfoloģiskās marķēšanas solis kopā ar iepriekšējā sadaļā aprakstīto dalīšanu tekstvienībās ir veicams arī tad, ja tekstam pieejams tikai morfoloģiskais, bet ne sintaktiskais marķējums. Tādējādi šie rīki ir lietojami, arī lai padarītu morfoloģiskā tagotāja (Paikens et al., 2013) rezultātus pieejamus starptautiskiem projektiem.

3.1.1.3 UD sintaktiskās struktūras

Transformācijas svarīgākais solis ir UD sintaktisko struktūru secināšana no pieejamā LVTB marķējuma.

Salīdzinot UD pamata atkarību koku ar LVTB marķējumu, konstatēts, ka (1) LVTB lietotās atkarību attieksmes atbilst UD atkarību saiknēm, (2) LVTB frāzēm līdzīgās konstrukcijas pārsvarā atbilst saistītiem kokveida fragmentiem UD kokā un (3) LVTB frāzēm līdzīgo konstrukciju atkarīgie

atbilst attiecīgo UD koka fragmentu saknes atkarīgajiem. Tāpēc transformāciju var veidot kā rekursīvu algoritmu, kas katru frāzes veida konstrukciju vai atkarību pārveido, izmantojot tās tuvākajā apkaimē pieejamo informāciju. Izstrādātās transformācijas pamats ir koka apstaigāšana (angl. *tree travesal*), vispirms rekursīvi apstrādājot katras virsotnes bērnus (gan atkarīgos, gan, ja tā ir frāzes virsotne, arī frāzes sastāvdaļas) un tad pašu virsotni (angl. *postorder travesal*).

Apstrādājot katru virsotni, tās atkarīgajiem tiek piešķiras UD atkarību lomas, vadoties gan pēc atkarīgo LVTB lomām un morfoloģiskajām pazīmēm, gan pēc vecāka morfoloģiskajām pazīmēm. Apstrādājot frāzēm līdzīgo konstrukciju virsotnes, tiek izveidots atbilstošais atkarību koka fragments, kā arī katrai konstrukcijas sastāvdaļai piešķirta UD atkarību loma, vadoties pēc tās LVTB lomas un morfoloģiskā marķējuma, kā arī frāzei līdzīgās konstrukcijas kopējā marķējuma – tipa, lomas teikumā un frāzes morfosintaktiskā marķējuma.

Tukšās virsotnes vārdu izlaidumu marķēšanai jeb redukcijas virsotnes tiek apstrādātas šādi:

1. redukcijas virsotnes bez atkarīgajām virsotnēm uz UD pārveidotas netiek;
2. redukcijas virsotnēm ar atkarīgajām virsotnēm – viens no atkarīgajiem atbilstoši to potenciālajām UD lomām tiek izvēlēts par reducētās vienības aizvietotāju, šo virsotni pakārto reducētās vienības vecākam un tai pakārto pārējos reducētās vienības atkarīgos.

Darba gaitā tiek izstrādāts detalizēts LVTB un UD izmantoto lomu savstarpējais kartējums (angl. *mapping*; šajā gadījumā – attēlojums "daudzi pret daudziem")[31], taču tiek arī secināts, ka atsevišķos gadījumos LVTB marķējums neļauj precīzi noteikt UD lomu. (Pretkalniņa et al., 2016)

UD vadlīniju 2.0. versija apraksta neobligāti pievienojamu papildmarķējumu, kas tālāk atvieglotu UD datu izmantošanu dažādos programmiskos lietojumos – tā sauktās paplašinātās atkarības (angl. *enhanced dependencies*). Vadlīnijas piedāvā marķējumā iekļaut piecu veidu papildinformāciju, katrs no papildinformācijas elementiem ir neobligāts:

1. tukšas (angl. *null*) virsotnes, kas reprezentē reducētus izteicējus (angl. *elided predicates*);
2. šķautnes, kas sasaista katru no vienlīdzīgajiem teikuma locekļiem ar to kopīgo vecāku un kopīgajiem atkarīgajiem (pamata atkarību

[31] LVTB mājaslapā http://sintakse.korpuss.lv/ tiek uz katru versiju publicēta jaunākā versija, darba rakstīšanas brīdī tā ir http://sintakse.korpuss.lv/docs/v2-11/LV2UD_mapping.pdf

kokā šāda tieša šķautne ir tikai pirmajam no vienlīdzīgo locekļu virknes);

3. šķautnes, kas norāda kontrolētos un paceltos teikuma priekšmetus (angl. *controlled/rised subjects*), piemēram, teikumā *viņš grib ēst* teikuma priekšmetu *viņš* sasaista ne tikai ar *grib* (šķautne pamatkokā), bet arī ar *ēst* un teikumā *pavelkot virvi, viņa atsēja mezglu* teikuma priekšmetu *viņa* sasaista arī ar *pavelkot*;
4. papildu šķautnes, kas sasaista palīgteikumu ievadošus attieksmes vietniekvārdus ar virsteikuma locekli, uz kuru vietniekvārds referē;
5. lomā ietverama papildinformācija:
    a) locījums vai prievārds var tikt pievienots nevalentā nominālā teikuma locekļa lomai, piemēram, *nmod:loc* lietvārdam lokatīvā;
    b) palīgteikumu ievadošais apstākļa vārds var tikt pievienots palīgteikuma lomai, piemēram, *advcl:kad* apstākļa palīgteikumam, ko ievada apstākļa vārds *kad*.

Analizējot vadlīnijas un LVTB marķējumu, tika secināts, ka bez papildu koreferenču marķēšanas 4. punktu izpildīt nav iespējams, jo LVTB marķējumā īpaši nav atzīmēts, vai vietniekvārdi un apstākļa vārdi ievada palīgteikumu vai ne. Šī paša iemesla dēļ 5. punkta nosacījumus ir iespējams izpildīt tikai daļēji – tiek norādīta tikai informācija par locījumiem un prievārdiem (a), bet ne palīgteikumus ievadošie apstākļi (b).

Paplašināto atkarību vadlīniju 1. punktu realizēt ļauj LVTB izmantotā reducēto elementu attēlošanas stratēģija, kas ietver tukšu virsotņu veidošanu izlaisto elementu vietā, savukārt 2. punktam nepieciešamo informāciju nodrošina tas, ka LVTB vienlīdzīgus teikuma locekļus attēlo ar frāzei līdzīgu konstrukciju, tādējādi nošķirot atsevišķa vienlīdzīgā locekļa atkarīgos no kopējiem atkarīgajiem.

LVTB saliktajiem izteicējiem izmantotā frāzes veida konstrukcija *xPred* ļauj iegūt būtisku daļu 3. punktam nepieciešamo saikņu, jo tā grupē kopā garākas izteicēju virknes, piemēram, [*viņš*] *gribēja gulēt*. Taču latviešu valodā arī divdabju konstrukcijām var būt ar attiecīgās teikuma daļas izteicēju kopīgs teikuma priekšmets, piemēram, teikumos *māte sakās par dēlu neko nezinām* (Kalnača, Lokmane, 2018) un *žāvādamies viņš piecēlās*. Šādas situācijas pieejamais LVTB marķējums neļauj nošķirt no teikumiem, kuros divdabim ir cits, noklusēts, teikuma priekšmets, piemēram, *redzēju līstam* un *aizķerot trolejbusa vadus, cirka ēka var sabrukt*. Neskaidrās situācijās paplašināto atkarību grafa šķautnes netiek pievienotas, lai neradītu kļūdainu marķējumu.

Sākotnēji transformācija tika projektēta tikai pamata atkarību koka izveidei. Pēc UD v2 specifikācijas publiskošanas tā tika papildināta ar funkcionalitāti, kas nepieciešamās LVTB redukcijas virsotnes pārveido par

UD paplašināto atkarību redukcijas virsotnēm, lai tās kopā ar pamata atkarību koku veido paplašināto atkarību "mugurkaulu", un pēc tam izveidoto "mugurkaulu" papildina ar nepieciešamajām papildsaitēm (Pretkalnina et al., 2018), izmantojot "mugurkaulā" jau esošo informāciju. Šāds risinājums izrādās neoptimāls, jo būtībā tas nozīmē, ka no informatīvākas datu struktūras (LVTB oriģinālais koks) tiek izveidota mazāk informatīva struktūra (UD pamatkoks), kurai pēc tam jāpievieno informācija, lai iegūtu paplašināto atkarību grafu. Tāpēc projekta "Daudzslāņu valodas resursu kopa teksta semantiskai analīzei un sintēzei latviešu valodā" (*FullStack-LV*) laikā (sk. 3.2.1. sadaļu) pirms UD v2.4 versijas transformators tiek pārstrādāts (Gruzitis et al., 2018) tā, lai katrā LVTB koka apstaigāšanas solī tiktu izveidotas visas pamata un paplašināto atkarību saites uz līdz šim apstrādātajām virsotnēm[32].

Pēc uzlabojumiem pārveides **algoritms** darbojas tālāk minētajos soļos.

1. Priekšapstrāde:
    1.1. Augšupejošā veidā apstaigājot koku, tiek sastādīts kartējums (šajā gadījumā – attēlojums "viens pret daudziem"), kurā virsotņu identifikatori norāda uz sarakstiem ar vienlīdzīgajiem teikuma locekļiem un vienlīdzīgajām teikuma daļām, no kā attiecīgās virsotnes sastāv.
    1.2. Lejupejošā veidā apstaigājot koku, tiek sastādīts kartējums (šajā gadījumā – attēlojums "viens pret daudziem"), kurā teikuma priekšmetu / teikuma priekšmeta palīgteikumu virsotņu identifikatori norāda uz sarakstiem ar virsotnēm, uz kurām attiecīgās virsotnes attiecināmas kā teikuma priekšmets vai teikuma priekšmeta palīgteikums. Sākotnēji katrā šādā sarakstā iekļauj virsotnes pēc LVTB tiešajām saitēm, tad to papildina ar salikto izteicēju daļām, ja attiecīgā virsotne saistīta ar saliktu (t.i., vairākvārdu) izteicēju. Virsotnes katrā iegūtajā sarakstā tiek sakārtotas pēc to dziļuma kokā, t.i., pēc tā, cik garš ir īsākais ceļš no saknes līdz attiecīgajai virsotnei.
2. Koka apstaigāšana un UD struktūras izveide, apstrādājot katru virsotni:
    2.1. Pirms kārtējās virsotnes apstrādāšanas rekursīvi tiek apstrādāti visi tās atkarīgie, un, ja tā ir frāzes veida konstrukcijas virsotne, tad arī tās sastāvdaļas.

[32] *GitHub* repozitorijā *CorporaTools* mapē LVTB2UD klase *NewSyntaxTranslator* https://github.com/LUMII-AILab/CorporaTools/blob/9e0786fe7713528d47e0aaeb7123e1f4fba2591a/LVTB2UD/src/lv/ailab/lvtb/universalizer/transformator/syntax/NewSyntaxTransformator.java

2.2. Tiek noskaidrots, kura tekstvienība reprezentē doto virsotni pamata atkarību kokā un kura – paplašinātajā grafā:
   2.2.1. ja tā ir redukcijas virsotne, tad tiek izdarīta "redukcijas transformācija" – tiek izanalizēti atkarīgie un noteikts, kurš no tiem kļūs par vecāka aizvietotāju pamata atkarību kokā, kā arī tiek izveidota tukšā virsotne paplašinātajam atkarību grafam;
   2.2.2. ja tā ir frāzes veida konstrukcijas virsotne, tiek izdarīta "frāzes transformācija" – tiek izanalizētas sastāvdaļas, izveidotas UD grafa šķautnes starp tām un noteikts, kura no sastāvdaļām kļūst par šo frāzi reprezentējošā UD koka fragmenta sakni – pamatkokā frāzi reprezentēs attiecīgajai sastāvdaļai atbilstošā (iepriekšējos soļos rekursīvi noteikta) tekstvienība un paplašinātajā grafā – tā pati tekstvienība vai tukšā virsotne, ja tā ir izrādījusies redukcijas virsotne;
   2.2.3. ja tā ir vienkārša tekstvienības virsotne, tad gan UD pamatkokā, gan paplašinātajā grafā to reprezentē atbilstošā tekstvienība.
2.3. Gan pamatkokā, gan paplašinātajā atkarību grafā tiek izveidotas atkarības starp apstrādājamās virsotnes atkarīgajiem elementiem (tos pārstāvošajām UD virsotnēm) un šo virsotni pārstāvošajām UD virsotnēm. Apstrādājot atkarīgos, kam 1.2. punktā izveidotajā kartējumā ir norādīti papildu vecāki, tiek novilktas arī paplašinātā atkarību grafa šķautnes uz tiem.
2.4. Paralēli katru reizi, kad tiek veidota škautne, tiek aplūkots 1.1. punkta kartējums un izveidotas visas nepieciešamās paplašinātā atkarību grafa saites starp topošās atkarības vecāka un bērna vienlīdzīgajiem elementiem. Saikņu lomas tiek noteiktas pēc virsotņu apkaimes kokā, vecāka un bērna morfoloģiskajām īpašībām un apstrādes konteksta (frāzes, redukcijas vai atkarības apstrāde).
3. Koka apstrādes beigās izveidotā koka saknes virsotnei tiek piešķirta speciālā loma *root*.

3.1.1.4 Morfoloģijas pēcapstrāde

Pēc tam, kad ir iegūta UD sintaktiskā struktūra, ir iespējams precizēt morfoloģisko marķējumu – galvenokārt, mazināt nepareizi piešķirto PRON instanču skaitu, pārmarķējot par vārdšķiru DET tos vietniekvārdus, kas UD sintaktiskajā struktūrā nokļuvuši determinētāja lomā *det*.

3.1.1.5 Salīdzinājums ar iepriekšējo eksperimentu transformācijām

Tā kā UD pamatatkarību struktūra arī tiek iegūta, LVTB frāzes veida struktūras pārveidojot par sakņotiem apakškokiem, tad šajā sadaļā aprakstītās transformācijas strukturālos aspektus ir iespējams salīdzināt ar 2.3.2.3. sadaļā aprakstītajām transformācijām:

- pieturzīmju konstrukcijas tiek transformētas tāpat kā *pmcBASELEM* transformācijā (?);
- vienlīdzīgie teikuma locekļi tiek transformēti atbilstoši stratēģijai *coord3_LEVEL* (2.3.2.3.2) ar niansi gadījumos, ja garāku vienlīdzīgu teikuma daļu virkni atdala semikols – UD abpus semikolam esošās vienlīdzīgo teikuma daļu virknes liek apstrādāt kā atsevišķas virknes un tad pakārtot vienu otrai;
- saliktie izteicēji ar palīgvārdu *būt* sastata izteicēja vai salikta laika konstrukcijā, kā arī ar palīgvārdiem *tikt* un *tapt* salikta laika konstrukcijā tiek transformēti tāpat kā *xpredBASELEM* stratēģijā, bet pārējie – *xpredDEFAULT*;
- frāzes veida konstrukcijas, kas sastāv no viena funkcionālā vārda un viena pilnnozīmes vārda, piemēram, prievārdiskās konstrukcijas *xPrep* (*uz galda*, *aiz stūra*), tiek transformētas otrādi nekā 2.3.2.3. sadaļā aprakstītajās stratēģijās – tur par frāzes sakni kļūst funkcionālais vārds, bet UD – pilnnozīmes.

Salīdzinot ar 2.3.2.4. sadaļā novēroto, var redzēt, ka vairums UD transformācijas elementu ir tādi, ko parsētājiem ir vieglāk iemācīties, tāpēc datu pārvēršana UD formātā palīdz sasniegt promocijas darba mērķi – iegūt augstākas precizitātes parsētāju latviešu valodai.

3.1.2 UDLV-LVTB kvalitatīvais novērtējums

Rakstā Pretkalniņa et al. (2018) veikts manuāls paraugkopas izvērtējums, kas ļauj spriest par transformācijas kvalitāti. Cilvēkresursu ierobežojumu dēļ izvērtējums ir salīdzinoši neliels, līdz ar to nav uzskatāms par augsti reprezentatīvu, taču par spīti tam sniedz vērtīgu ieskatu transformācijas rezultātu kvalitātē. Tiek pārbaudīti 60 teikumi (aptuveni 800 tekstvienību, teikumi izvēlēti, proporcionāli atspoguļojot korpusā pārstāvētos teksta žanrus) un atrastajām UD marķējuma kļūdām tiek noteikts, vai tās radušās kļūdaina oriģinālmarķējuma dēļ, transformācijas nepilnību dēļ vai arī tāpēc, ka LVTB nav informācijas, kas ļautu noteikt pareizās saites un to lomas.

Pamata atkarību kokos tiek identificētas 19 nepareizas saites / saišu lomas, t.i. transformācijas LAS ir 97,6%, kas uzskatāms par ļoti labu

rezultātu. No atrastajām problēmām tikai viena ir tāpēc, ka LVTB nesatur nepieciešamo informāciju, savukārt sešas – oriģināldatu kļūdu dēļ. Pārējās 12 uzskatāmas par uzlabojamām transformācijas kļūdām.

Tajos pašos datos tiek novērtēta arī paplašināto atkarību grafu kvalitāte, taču, to darot, sakarā ar paplašināto atkarību neobligāto raksturu kā kļūdas netiek skaitītas paplašināto atkarību saišu grupas, kas vispār netiek marķētas, t.i., 3.1.1.3. sadaļā aprakstītās paplašināto atkarību specifikācijas grupas 4 un 5(b). Tiek identificētas trīs kļūdas oriģināldatu kļūdu dēļ, astoņas paplašināto atkarību saites ar nepareizām lomām (visām pievienots nepareizs locījums vai prievārds – paplašināto atkarību specifikācijas grupa 5(a)) un 15 trūkstošas vienlīdzīgo teikuma locekļu vai teikuma priekšmetu saites (attiecīgi paplašināto atkarību specifikācijas grupas 2 un 3). Lai gan testa datos netiek konstatēti gadījumi, kad kļūdas būtu radušās tāpēc, ka LVTB nesatur nepieciešamo informāciju, nav ticami, ka šādu gadījumu nav vispār, taču var spriest, ka tie ir salīdzinoši reti.

Veiktais novērtējums ir neliela apjoma datiem, taču ļauj pozitīvi raudzīties uz transformācijas precizitāti. Turklāt 3.1.1.3. sadaļā aprakstītā transformatora pārveide pirms UD v2.4 ir tieši orientēta uz būtiskāko trūkumu uzlabošanu – tā būtiski uzlabo ar vienlīdzīgiem teikuma locekļiem un teikuma priekšmetiem saistīto paplašināto atkarību saišu piešķiršanas mehānismu, tādējādi samazinot iespējamo kļūdu apjomu.

3.2 Korpusa nozīme un ietekme

Šajā sadaļā aprakstīta UDLV-LVTB tālākā izmantošana – gan pētījumu projektos, gan latviešu valodas parsētāju būvēšanai.

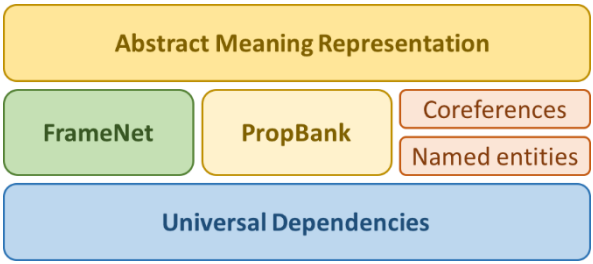
3.2.1 Korpusa dati kā pamats tālākiem pētījumiem

2017.–2019. gadā LU MII sadarbībā ar ziņu aģentūru LETA īstenoja projektu "Daudzslāņu valodas resursu kopa teksta semantiskai analīzei un sintēzei latviešu valodā"[33] (*FullStack-LV*), kurā būtiska nozīme ir UDLV-LVTB. Projekta laikā tika radīts 10 tūkstošus teikumu liels, līdzsvarots teksta korpuss, kas ir marķēts gan sintaktiski, gan vairākos līmeņos semantiski, izmantojot pasaulē plaši aprobētas sintaktiskās un semantiskās reprezentācijas, kas tika pielāgotas latviešu valodai. Teikuma semantiku (sk. 1. attēlu) šajā korpusā attēlo, izmantojot *FrameNet* (Ruppenhofer, 2010) un *PropBank* (Bonial et al., 2014) modeļus, bet teksta semantikas attēlošanai izmanto abstraktās nozīmes reprezentācijas (angl. *abstract meaning*

[33] Eiropas Reģionālās attīstības fonda (ERAF) praktiskas ievirzes pētījums 1.1.1.1/16/A/219

representation, AMR) (Banarescu et al., 2013) modeli. Tāpat korpusā marķē arī nosauktās entitātes (angl. *named entities*) un koreferences.

FrameNet un *PropBank* līmeņi tiek marķēti pēc UD pieejas sintaktiski marķētā korpusā. UD korpuss tiek automātiski atvasināts no LVTB, kas tiek marķēts manuāli atbilstoši 2.1. sadaļā aprakstītajam gramatikas modelim (sk. 8. attēlu). Paralēli tiek publicētas abas sintaktiski marķētā korpusa versijas. Tādējādi LVTB ir balstīti visi nākamie marķējuma līmeņi *FullStack-LV* daudzslāņu valodas resursu kopā. Projekta ietvaros tika būtiski paplašināta un uzlabota LVTB2UD transformācija, kā arī pats LVTB tika paplašināts līdz 13,6 tūkstošiem teikumu (Gruzitis et al., 2018). Viens no būtiskākajiem projekta sasniegumiem ir integrētā plaša izmantojuma rīkkopa dažādiem latviešu valodas sapratnes (angl. *natural language understanding*, NLU) uzdevumiem *NLP-PIPE*[34], kurā vienotā, konfigurējamā apstrādes rīkplūsmā (angl. *pipeline*) apkopotas nepieciešamās NLU komponentes, kas ir apmācītas ar *FullStack-LV* datiem (Znotiņš, Cīrule, 2018; Gruzitis, Znotins, 2018; Paikens, 2017).



8. attēls. *FullStack-LV* daudzslāņu teksta korpuss: zemāk attēlotie reprezentācijas slāņi tiek izmantoti par pamatu augstāk attēloto slāņu veidošanai (sk. plašāku kontekstu 1. att.)

LVTB un UDLV-LVTB reprezentāciju un korpusu pilnveide tiek turpināta 2020.–2024. gadā Valsts pētījumu programmās "Humanitāro zinātņu digitālie resursi" [35] un "Letonika latviskas un eiropeiskas

[34] Pirmkods pieejams *GitHub* repozitorijā https://github.com/LUMII-AILab/nlp-pipe, lejuplāde pieejama *Clarin.lv* repozitorijā http://hdl.handle.net/20.500.12574/4, demonstrācija pieejama https://nlp.ailab.lv

[35] Projekts "Humanitāro zinātņu digitālie resursi: integrācija un attīstība" Valsts pētījumu programmas "Humanitāro zinātņu digitālie resursi" ietvaros, projekta Nr.: VPP-IZM-DH-2020/1-0001

sabiedrības attīstībai" [36] atbilstoši jaunākajām UD vadlīnijām un jaunākajiem uz LVTB balstītajiem latviešu valodas gramatikas pētījumiem. Korpusu LVTB un UDLV-LVTB apjoms tiek palielināts līdz 17 tūkstošiem teikumu, kas arī apliecina resursa nozīmību.

3.2.2 Latviešu valodas parsētāju attīstība

Vēsturiski nozīmīgs punkts latviešu valodas parsētāju attīstībā ir 2016. gads, kad *Google* publisko *SyntaxNet* bibliotēku un parsētāja modeļus 40 valodām, kas apmācīti, izmantojot UD v1.3. Bibliotēka un moduļi ir brīvi pieejami[37], un 40 valodu skaitā ir arī latviešu valoda – šī ir pirmā reize, kad parsētāju latviešu valodai būvē starptautiski atzīta citvalstu pētnieku grupa, par pamatu izmantojot Latvijā sagatavotos datus. Latviešu valodas parsēšanas modulis ir apmācīts, izmantojot 3985 tekstvienības lielu korpusu[38], un dod 58,92% UAS, 51,47% LAS[39]. Angļu valodai *SyntaxNet* dod jaunu rezultātu rekordu (Andor et al., 2016).

2017. gadā jau notiek parsētāju veidošanas sacensības *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (Zeman et al., 2017), kurās kā datus izmanto UD v2.0 45 valodām, tai skaitā latviešu valodai. Labākos rezultātus[40] latviešu valodai uzrāda sacensībās uzvarējušais parsētājs *Stanford* – tas sasniedz 79,26% UAS un 74,01% LAS (Dozat et al., 2017). Labus rezultātus uzrāda arī otrs labākais parsētājs *C2L2* – 77,43% UAS un 71,35% UAS (Shi et al., 2017). Šajā pašā gadā latviešu

[36] Projekts "Mūsdienu latviešu valodas izstrāde un valodas tehnoloģiju attīstība (LATE)" Valsts pētījumu programmas "Letonika latviskas un eiropeiskas sabiedrības attīstībai" ietvaros, projekta Nr.: VPP-LETONIKA-2021/1-0006

[37] Apmācītie moduļi ir atrodami *GitHub* repozitorijā *tensorflow models* repozitorija vēsturē:
https://github.com/tensorflow/models/tree/a5d45f2ed20effaabc213a2eb9def291354af1ec/syntaxnet

[38] UD vadlīnijas https://universaldependencies.org/release_checklist.html#data-split nosaka, ka katrs korpuss tiek publicēts, norādot kanonisko datu sadalījumu apmācības, kalibrēšanas un novērtēšanas datu kopās, un rekomendē teikumus dažādās versijās nepārvietot starp sadalījuma kopām. UDLV-LVTB šo rekomendāciju pilda, proporcionāli palielinot visas dalījuma kopas, ja tiek publicēta lielāka korpusa versija.

[39] Rezultātu pārskats ir atrodams *GitHub*, *tensorflow models* repozitorija vēsturē:
https://github.com/tensorflow/models/blob/a5d45f2ed20effaabc213a2eb9def291354af1ec/syntaxnet/universal.md

[40] Visi rezultāti pieejami sacensību mājaslapā
https://universaldependencies.org/conll17/results.html

valodas UD korpuss tiek izmantots kā viens no korpusiem pētījumā (Nivre, Fang, 2017), kas argumentē, ka LAS metrika lielākoties dod labākus rezultātus analītisko valodu, piemēram, angļu, parsētājiem, un piedāvā alternatīvu metriku, tādējādi mazinot vairākas desmitgades ilgušo angļu valodas dominanci iegultajos rīku un metriku pieņēmumos nozarē.

2018. gadā šīs sacensības notiek vēlreiz, izmantojot UD v2.2 datus no 82 korpusiem 57 valodām. Latviešu valodai tobrīd pieejamais sintaktiski marķētais korpuss ir jau 81 tūkstoti tekstvienību liels (Zeman et al., 2018). Labākos rezultātus[41] latviešu valodai uzrāda parsētājs *HIT-SCIR*, sasniedzot 87,76% UAS un 83,97% LAS (Che et al., 2018). Otro labāko rezultātu dod parsētājs *Stanford* – 85,97% UAS un 81,85% LAS (Qi et al., 2018). Šajās sacensībās rezultāti tiek salīdzināti, izmantojot arī jaunu metriku MLAS, kas ir līdzīga LAS, taču ņem vērā arī morfoloģiskā marķējuma pareizumu. Izmantojot šo metriku, latviešu valodai labākos rezultātus sniedz parsētājs *Stanford* – 67,89%.

Gan LAS, gan MLAS metrikās latviešu valodai sasniegtie rezultāti ne tikai būtiski pārsniedz labāko parsētāju vidējos rezultātus pa valodām, tas ir, 75,84% LAS un 61,25% MLAS, bet arī pārsniedz vidējo rezultātu t.s. lielo korpusu grupā, kas ir 84,37% LAS, 72,67% MLAS. Lielo korpusu grupā šajās sacensībās ir ieskaitīts 61 korpuss, kuru apjoms ir vismaz 25 tūkstoši tekstvienību. No 2.2.2. sadaļā pieminētajiem lielākajiem UD korpusiem šajās sacensībās piedalās čehu valodas *UD_Czech-PDT* un krievu valodas *UD_Russian-SynTagRus* (tobrīd mazāks – ap 1 milj. tekstvienību), uz kuriem uztrenētie parsētāji sasniedz vienus no sacensību augstākajiem rezultātiem, pārsniedzot 90% LAS un 85% MLAS. Kopumā šo sacensību rezultāti apliecina, ka sacensībās latviešu valodai, pateicoties kvalitatīvajam korpusam, ir radīti augstas rezultativitātes parsētāji.

2020. gadā notiek paplašināto atkarību parsētāju veidošanas sacensības *IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies* (Bouma et al., 2020), un latviešu valoda ir starp 17 valodām, kuru dati tiek piedāvāti sacensību dalībniekiem. Latviešu valodai labākais parsētājs sasniedz 85% ELAS precizitāti[42] (ELAS – LAS atvasināta metrika, kas pielāgota tieši paplašināto atkarību grafu novērtēšanai). 2021. gadā sacensības notiek vēlreiz (Bouma et al., 2021), un rezultāti sasniedz pat 90,25% ELAS un 91,25% LAS precizitāti[43] latviešu valodai, pārsniedzot

[41] Visi rezultāti pieejami sacensību mājaslapā https://universaldependencies.org/conll18/results.html

[42] Sacensību rezultātu pārskats pieejams sacensību mājaslapā https://universaldependencies.org/iwpt20/Results.html

[43] Sacensību rezultātu pārskats pieejams sacensību mājaslapā https://universaldependencies.org/iwpt21/results.html

vidējos rādītājus, kas ir 89,24% ELAS un 89,81% LAS. Šajās divās sacensībās parsētāja precizitāte tiek mērīta paplašināto atkarību grafos, tāpēc, lai gan šeit dotie skaitļi nav precīzi salīdzināmi ar *CoNLL 2017-2018* rezultātiem, taču šie grafi satur vairāk informācijas kā pamatu atkarību koki, tāpēc būtībā šis analīzes uzdevums ir pat grūtāks. Tādējādi šie augstie rezultāti (salīdzinājumam vēl 2016. gadā Andor et al. par *SyntaxNet* ziņo, ka angļu valodas precizitāte ir virs 90% LAS pamata atkarībām, bet čehu valodas – mazliet zem šī rādītāja) pilnvērtīgi apstiprina darba 2. hipotēzi, ka vidēja izmēra korpuss (10–20 tūkstoši teikumu) ir pietiekams augstas kvalitātes *state-of-the-art* parsētāju izveidei.

Paralēli korpusā balstās arī Latvijā izstrādātie parsētāju un ar tiem saistīto tehnoloģiju pētījumi. 2016. gadā tiek publicēts pētījums, kurā izmantota tobrīd inovatīvā un ļoti aktuālā jēdzientelpas (angl. *word embeddings*) pieeja. Šajā pētījumā UDLV-LVTB korpusā apmācītais parsētājs sasniedz 74,9% UAS (Znotiņš, 2016). 2018. gadā *FullStack-LV* projektā NLP-PIPE rīkā iegūti rezultāti 81,2% UAS un 76,8% LAS (Znotiņš, Cīrule, 2018). Savukārt 2020. gadā LVBERT rīks nodrošina 89,9% LAS (Znotiņš, Bārzdiņš, 2020) un 2022. gadā (VPP, 2022) – 90,79% LAS. Analizējot parsētāja rezultātus atkarībā no apmācības datu apjoma (sk. 1. tabulu), ir novērojams būtisks precizitātes kāpums, ja salīdzina parsētājus, kas izmanto vienu vai divas piektdaļas datu. Salīdzinot parsētājus, kas izmanto divas, trīs vai četras piektdaļas korpusa, kāpums ir aptuveni puse procentpunkta, savukārt, pievienojot pēdējo korpusa piektdaļu, tiek iegūts niecīgs precizitātes uzlabojums.

Šie rezultāti ļauj secināt, ka ar šādu korpusa apjomu pietiek, lai pilnvērtīgi izmantotu šobrīd pieejamās parsētāju būvēšanas tehnoloģiju iespējas un iegūtu augstas kvalitātes parsētājus latviešu valodai. Tas nozīmē, ka izveidotais korpuss ir optimāls darbā izvirzīto uzdevumu izpildei.

1. tabula. LVBERT parsētāja precizitātes pieaugums atkarībā no apmācības datu apjoma (VPP, 2022)

Izmantotā apmācības datu kopas (*train*) daļa	LAS % parametru kalibrēšanas datu kopā (*dev*)	LAS % novērtēšanas datu kopā (*test*)
20%	84,71	85,08
40%	88,51	89,28
60%	89,54	89,94
80%	90,13	90,56
100%	90,31	90,79

Secinājumi

Promocijas darbā izvirzītais mērķis – apjomīga mašīnlasāma sintaktiski marķēta korpusa izveide un aprobēšana – ir sasniegts. Izvirzītās hipotēzes par hibrīda marķējuma modeļa priekšrocībām un kvalitatīva vidēja apjoma korpusa piemērotību augstas precizitātes (~90%) parsētāju apmācībai ir praktiski apstiprinātas.

Darba ietvaros secināts tālāk minētais.

- Izvēle veidot un attīstīt darbā aprakstīto hibrīdo gramatikas modeli ir izrādījusies ļoti veiksmīga, jo šis modelis ļauj reprezentēt latviešu valodas sintaktiskās parādības, saglabājot būtiskas nianses, kuras ne vienmēr ir iespējams precīzi attēlot tīrā atkarību vai frāžu struktūras gramatikas modelī.
    - Turklāt korpusa marķēšana bagātīgā hibrīdā formātā ļauj veidot pielāgotas transformācijas uz plaši lietotiem, bet ietvertās informācijas ziņā nosacīti vienkāršākiem formātiem (piem., UD), kā arī veikt pētījumus par piemērotākajiem atkarību attēlojumiem dažādām gramatikas parādībām.
    - Atkarību attēlojums var ietekmēt gan parsētāju precizitāti, gan teksta analīzes rīkus, kas izmanto parsētājus. Novērotā ietekme nav viennozīmīga – dažādi rīki dod priekšroku dažādiem atkarību attēlojumiem.
    - Salīdzinot ar vienkāršajiem un skaitļošanas ziņā efektīvajiem atkarību modeļiem, hibrīdu parsētāju algoritmu un mašīnmācīšanās modeļu attīstība ilgu laiku nav bijusi globāla aktualitāte, par to vairāk interesējušies pētnieki, kas pēta morfoloģiski bagātas valodas. Taču tagad jautājums par hibrīdu parsētājiem ir atkal aktualizējies plašāk un jaunākie eksperimenti uzrāda daudzsološus rezultātus (Nivre et al., 2022).
- UD gramatikas modelis ir ļoti veiksmīgs tālākiem pētījumiem un parsētāju apmācībai ne tikai tā starptautiskā rakstura dēļ, bet arī tāpēc, ka, vadoties pēc darbā veiktās parametriskās analīzes, tajā daudzviet izvēlēti tādi sintaktisko konstrukciju atainošanas paņēmieni, kuriem arī parsētāju apmācības sistēmas dod labākus rezultātus.
    - Hibrīdajam modelim, kas izmantots "Latviešu valodas sintaktiski marķētajā korpusā", un UD modelim ir pietiekami daudz kopīgo elementu, lai būtu iespējams izveidot augstas precizitātes transformāciju vismaz vienā virzienā. UD modelis ir

nosacīti vienkāršāks, tāpēc abpusēja transformācija netika veidota.
    - Hibrīdajam modelim un UD modelim atšķiras lomu izvēles un iedalījuma kritēriji – hibrīdajā modelī izmantoto sintaktisko lomu šķirums vairāk balstīts teikuma semantiskajā struktūrā, un lomas ir, piemēram, apzīmētājs/atribūts (*attr*), apstāklis (*adv*). Savukārt UD lomu izvēlē galvenokārt ņemti vērā morfosintaktiski kritēriji, piemēram, daudzos gadījumos tās ir grupētas pa vārdšķirām – nomināls modificētājs (*nmod*), adjektīvisks modificētājs (*amod*), adverbiāls modificētājs (*advmod*) utt.
    - Turklāt likumos balstīta transformēšana uz citu (šajā gadījumā UD) marķējuma modeli palīdz atklāt marķējuma kļūdas un nekonsekvences sākotnējā korpusā un tādējādi, izmantojot atgriezenisko saiti, uzlabot korpusa kvalitāti.
- Secināts, ka sasniegtais korpusa lielums (17 tūkstoši teikumu) ir pietiekams, lai izveidotu pasaules līmeņa precizitātes parsētāju – latviešu valodai labākais parsētājs *IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies* sacensībās sasniedz 91,25 precizitāti (LAS), kas būtiski pārspēj vidējo visu valodu rezultātu un ir pielīdzināms vidējam rezultātam lielo korpusu grupā.
    - Pateicoties korpusa apjomam, kvalitātei un savietojamībai, dažādas pētnieku grupas ir veiksmīgi izmantojušas UDLV-LVTB datus četrās *CoNLL* un *IWPT Shared Task* sacensībās 2017., 2018., 2020. un 2021. gadā augstas precizitātes parsētāju izstrādei un novērtēšanai. Tāpat korpusa dati ir sekmīgi lietoti parsētāju pētījumos Latvijā.
    - Analizējot parsētāja rezultātus atkarībā no apmācības datu apjoma, ir redzams, ka pieejamais korpusa apjoms ir pietiekams, lai pilnvērtīgi izmantotu šobrīd pieejamās parsētāju būvēšanas tehnoloģiju iespējas un iegūtu augstas kvalitātes parsētājus latviešu valodai.
    - Pārliecinošie latviešu valodas UD parsētāju mašīnapmācības rezultāti apliecina korpusa marķējuma kvalitāti un viendabību.
- Dalība UD iniciatīvā ir sekmējusi latviešu valodas un arī citu fleksīvu valodu resursu starptautisko atpazīstamību un veicina fleksīvām valodām piemērotāku rīku izveidi pētniecības jomā, kuras vēsturiskā izcelsme pamatā meklējama darbā ar analītiskajām valodām.

UNIVERSITY
OF LATVIA

FACULTY OF COMPUTING

Lauma Pretkalniņa

FORMAL MODEL OF LATVIAN GRAMMAR AND ITS IMPLEMENTATION IN A MACHINE-READABLE TREEBANK

SUMMARY OF DOCTORAL THESIS

Submitted for the Doctoral degree in Computer Science
Subfield of Computer and Systems Software

Riga, 2023

This doctoral thesis was developed at the University of Latvia,
Institute of Mathematics and Computer Science,
Artificial Intelligence Laboratory
over the period of time from 2011 to 2023

European Social Fund project "Support for Doctoral Studies at the University of Latvia" No. 2009/0138/1DP/1.1.2.1.2./09/IPIA/VIAA/004.

The thesis consists of an introduction, 2 chapters, conclusions and references.

Thesis form: a set of publications in the field of computer science, the subfield of computer and system software.

Thesis supervisor: assoc. prof. Dr. Normunds Grūzītis, University of Latvia

Reviewers:

1) Prof. Dr. Ģirts Karnītis, University of Latvia
2) Assoc. prof. Dr. Andrius Utka, Vytautas Magnus University
3) Dr. sc. comp Mārcis Pinnis, SIA "Tilde"

The defence of the thesis will take place at 16:00 on April 14, 2023 in an open session of Doctoral Council of the Field of Computer and Information science of the University of Latvia in Institute of Mathematics and Computer Science, University of Latvia, Riga, Raiņa bulvāris 29 room 413

The doctoral thesis and its summary can be found in the Library of the University of Latvia in Riga, 4 Kalpaka bulvāris.

UL Promotion Council of Computer Science,
Chairman of the Promotion Council: Jānis Bičevskis
Secretary of the Promotion Council: Ruta Ikauniece

Acknowledgements

I would like to thank everyone who has supported, inspired and advised me in the writing of this thesis; without your help this work would certainly not have been completed. I would like to thank my family and friends for their unending patience and support. I would like to thank my colleagues for inspiring me and sharing their knowledge. I would like to thank Normunds, who believed in my work even when I myself did not. I would like to thank Baiba and Madara for the readability of this work. I would like to thank Ingus for his expertise in foreign languages and mathematics. I would like to thank Jānis for his love and delicious lunches.

The work on this thesis was started with the support of the European Social Fund for the project "Support for Doctoral Studies at the University of Latvia" (2009/0138/1DP/1.1.2.1.2/09/IPIA/VIAA/004). Further fundamental research was funded within the framework of the State Research Programmes "National Identity (Language, Latvian History, Culture and Human Security) No. 3" and "Digital Resources for Humanities" (VPP-IZM-DH-2020/1-0001). Industry-driven research (in cooperation with the news agency LETA) was carried out within the following European Regional Development Fund projects: Information and Communication Technology Competence Centre study No. 2.7 "Exploring the Potential of Automatic Information Extraction for Latvian Media Monitoring" (KC/2.1.2.1.1/10/02/001) and industry-driven research project "Full Stack of Language Resources for Natural Language Understanding and Generation in Latvian (FullStack-LV)" (1.1.1.1/16/A/219). The direction of the research is continued in the project "Research of the Modern Latvian Language and Development of Language Technologies" within the State Research Programme "Letonika – Fostering a Latvian and European Society" (VPP-LETONIKA-2021/1-0006). The concluding part of the thesis was completed with the support of the European Social Fund for the project "Strengthening the Doctoral Capacity of the University of Latvia Within the Framework of the New Doctoral Model" (8.2.2.0/20/I/006).

Abstract

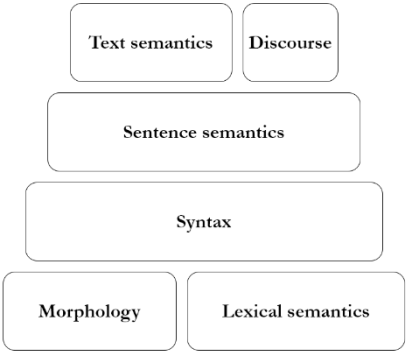
The given doctoral thesis describes the creation of a hybrid grammar model for the Latvian language, as well as its subsequent conversion to a Universal Dependencies (UD) grammar model. The thesis also lays the groundwork for Latvian language research through syntactically annotated texts. In this work, a fundamental Latvian language resource was developed and evaluated for the first time – a machine-readable treebank of 17 thousand syntactically annotated sentences. The sentences are annotated according to two syntactic annotation models: the hybrid grammar model developed in the thesis, and the internationally recognised UD model. Both annotated versions of the treebank are publicly available for downloading or querying online.

Over the course of the study, a set of tools and infrastructure necessary for treebank creation and maintenance were developed. The language coverage of the IMCS UL experimental hybrid model was extended, and the possibilities were defined for converting data annotated according to the hybrid grammar model to the dependency grammar model. Based on this work, a derived UD treebank was created.

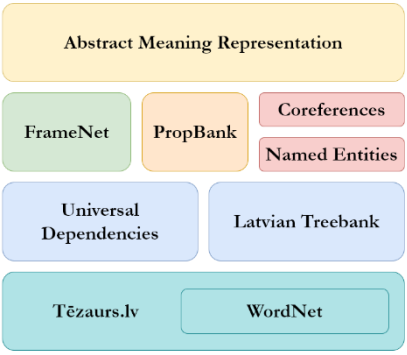
The resulting treebank has served as a basis for the development of high accuracy (91%) Latvian language parsers. Furthermore, the participation in the UD initiative has promoted the international recognition of Latvian and other inflective languages and the development of better-fitted tools for inflective language processing in computational linguistics, which historically has been more oriented towards analytic languages.

1 Introduction

This doctoral thesis was developed in the field of computational linguistics – an interdisciplinary field that deals with the modelling and processing of natural language using computational methods. The main task of computational linguistics is the automatic extraction of structured, machine-readable and machine-interpretable information from natural language[1], as well as the representation of machine-readable information (data) through natural language; thus the central aspects of computational linguistics are formed by analysis (natural language understanding, NLA) and synthesis (natural language generation, NLG). Language processing in both linguistics and computational linguistics tends to be viewed as a multi-level task (see Figure 1a), where grammatical and semantic information on each of the levels can be analysed using formal models of meaning representation (see Figure 1b; for the sake of clarity, representation models used in the projects related to the doctoral thesis are mentioned here, although they are not the only ones).



(a) Levels of text analysis (by Grūzītis, 2011)



(b) Selected representations (by Grūzītis et al., 2018)

Figure 1: Language processing and analysis

Numerous problems of modern computational linguistics and their practical solutions derive from language analysis on semantic levels. For

[1] Although this work focuses on the text and does not address the problem of speech recognition and synthesis, it should be noted that the result of speech recognition is a text that requires further processing and analysis, and the input data for speech synthesis is also a text.

example, the operation of virtual assistants popular today (*Siri*[2], *Alexa*[3], etc.) is typically ensured by text classification and information extraction. To solve such tasks, different approaches can be distinguished, depending on what intermediate steps and resources are used to obtain the solution. One of such approaches is carrying out the analysis step by step, starting with the analysis on the lowest i.e., morphological level and continuing with a syntactic analysis based on morphological analysis, etc., until the desired level of analysis or meaning representation of the text is reached. Another approach is immediately addressing the final problem without a direct and thorough analysis of the lower levels of language. This second approach solves only the specific task at hand, and the solution may be difficult to generalise for other tasks on the same level of analysis. For example, lists of keywords or word embeddings may serve as a sufficient solution for a text classification task, but such a resource is not enough to retrieve factual information from the text. The development of the resources described in the first approach requires more time and human resources, but the application of solutions developed by such a method is subsequently wider and will continue to be useful in the development of new applications. Therefore, the development of wider-coverage solutions on lower language levels is a fundamental investment in addressing problems on higher levels of a language later on. The research conducted in this doctoral thesis, as well as its results are oriented towards the step-by-step text analysis i.e., the first approach.

The lowest levels of text processing consist of the division of a text into tokens and morphological analysis. Morphology is a subfield of linguistics that studies and describes the grammatical structure of words; however, in computational linguistics morphological analysis often implies the analysis of not only words, but also punctuation, i.e. all tokens of text. There is a multitude of individual problem cases on this level as well, but a fundamental, comprehensive solution for the Latvian language already exists, which uses a lexicon to generate analysis variants (Paikens, 2007), and a statistical, corpus-data-trained tagger with a 98% accuracy rate of part of speech recognition and a 93% accuracy rate of full morphological annotation including case, gender and number for nouns and adjectives, tense, person and number for verbs, etc. (Paikens et al., 2013; Paikens, 2017).

On the next – syntactic – level, formal connections between words, i.e. the elements of a sentence, are considered, while the semantic levels of a

[2] *Apple Siri* homepage: https://www.apple.com/siri/
[3] *Amazon Alexa* homepage: https://alexa.amazon.com/

sentence and text examine the meaning of a sentence[4] or text as a whole. This doctoral thesis is devoted to language analysis on a syntactic level. Prior to the start of this research, the scope of syntactic analysis tools and resources for the Latvian language were quite limited – there was a lack of an effective wide-coverage parser and a syntactically annotated corpus i.e., a treebank, for the creation of a machine-learning parser. There was an experimental rule-based partial parser that used a set of manually compiled rules (Bārzdiņš et al., 2007), but the set of rules only covered extended simple sentences, and the obtained results were ambiguous: each sentence was offered all possible analysis variants corresponding to the morphological characteristics of the tokens in the sentence. There were also existing studies devoted to specific problems, such as spell-checking (Deksne, Skadiņš, 2011) or controlled natural languages (Paikens, Grūzītis, 2012).

The syntax-level technologies successfully developed in the given doctoral thesis, and the Universal Dependencies treebank in particular, form the necessary basis for further, higher-level research (see Figure 1b).

1.1 Research Problem

The research problem defined in this doctoral thesis is the development and evaluation of a complete (wide-coverage) formal model of the Latvian grammar, including the creation of a large, machine-readable treebank that corresponds to the previously mentioned model.

1.2 Topicality of the Research Problem

The results of this doctoral thesis open up a wide range of opportunities for new tools and research.

- Through the use of the treebank, it is possible to create various new language analysis tools, with specific emphasis on high-precision syntactic parsers for the Latvian language and the opportunity to participate in shared parser building tasks.
- The treebank, which is publicly available in two data formats annotated according to two syntactic annotation models, allows the previously impossible option of selecting Latvian language data for research according to syntactic criteria.
- A widely used international standard – UD – has been adapted for the Latvian language. In addition, Latvian is the first Baltic language for

[4] For the sake of brevity, unless otherwise indicated, here and further the notion *sentence* includes both predicative and non-predicative units (utterances).

which such an extensive resource is being developed. Thus, participation in the UD initiative promotes international cooperation and allows the dissemination of insights into the peculiarities and needs of inflective languages with a rich morphology in what is historically a more Anglocentric environment in computational linguistics.

- New tools for processing the Latvian language are important not only for scientific development, but also for the general public – they contribute to a wider integration of the Latvian language into electronic media and the survival of Latvian in the Digital Age.

1.3 Goals and Objectives of the Study

The general objective of the study is to start a new direction of research of the Latvian language, namely, the studies of computational linguistics that are based on syntactically annotated texts. The aim of this research is to develop and evaluate a fundamental language resource so far unprecedented for the Latvian language i.e., a machine-readable treebank. To achieve this aim, the following tasks have been set:

- to create methods and the necessary technical framework for the creation of a Latvian treebank, including the comparison of different syntactic annotation models;
- to design an experimental treebank prototype;
- to develop data transformations for the use of a treebank in various formats, including those in line with the internationally recognised UD approach;
- to prepare the treebank data for application in parser training, including international shared tasks, and to ensure public data access for various studies.

1.4 Hypotheses

The following hypotheses have been proposed in the thesis:

1) a hybrid grammar model of dependencies and phrase structures will expand the usability of the Latvian treebank in comparison to dependency grammar model;

2) a high-quality, medium-sized (approx. 10-20 thousand sentences[5]) treebank of the Latvian language will serve as the basis for the creation of state-of-the-art parsers.

The hypotheses proposed in this thesis have been confirmed through practical studies.

1.5 Research Methods

The following research methods have been used in the doctoral thesis:

- literature review – in order to identify the perspective research methods and experience with other language, various scientific publications and, in some cases, the source code of open sourced tools have been analysed;
- iterative development and adaptation – the tools and algorithms developed over the course of the thesis were implemented, evaluated and iteratively refined based on their practical application;
- quantitative evaluation – the transformation algorithms developed during the thesis were evaluated using metrics approved in the field of research;
- controlled experiments – variants of algorithms were compared in controlled environment experiments with the help of the quantitative evaluation method, analysing their effective differences and accuracy;
- error analysis – where possible, manual error analysis was also performed randomly within the results of the algorithms in order to get a better idea of potential problems and their types.

1.6 Main Results

The main results of the doctoral thesis are as follows:

- a developed set of tools and the infrastructure for the creation of the Latvian Treebank (LVTB), including defined extensions of the IMCS UL experimental hybrid model of Latvian grammar necessary for a wide language coverage.

[5] A retrospective look at the UD multilingual treebank data and the *CoNLL* 2018 results show that the average volume of a 'large' corpus is 12.5 thousand sentences and the average accuracy rate of the best parser for this type of corpus is 84.37%. The organizers of this shared task considered a corpus to be 'large', when it had enough data to distinguish separate machine-learning datasets not only for training and testing, but also for development (*dev*) purposes.

- a complete analysis of possibilities regarding the conversion of data annotations according to the hybrid grammar model into annotations according to the dependency model, as well as a study of the effect of various transformations on the accuracy and further use of parsers. The creation of a transformation that is capable of deriving the Latvian UD Treebank (UDLV-LVTB) from the data annotated according to the hybrid grammar model.
- the most important indirect result – a new direction of interdisciplinary research for the Latvian language and the basis for the development of fundamental language technologies: (1) the corpus – LVTB and UDLV-LVTB (Latvian part of UD versions v1.3–v2.11); (2) parsers for the Latvian language.

1.7 Practical Significance and the Evaluation of Results

Using the syntax model and annotation infrastructure created in this thesis, a treebank consisting of 17 thousand sentences has been created, which is publicly available in both hybrid[6] and UD representation[7].

The treebank has served as the basis for cooperation between IMCS UL and the news agency LETA in the ERDF industry-driven research project "Full Stack of Language Resources for Natural Language Understanding and Generation in Latvian" (2017-2019; see Section 3.2.1), in which a multilayer syntactically and semantically annotated corpus (sembank) have been created; in addition, the toolkit NLP-PIPE (Znotiņš, Cīrule, 2018) has been created on its basis for a thorough analysis of Latvian texts (according to Figure 1b). In turn, the NLP-PIPE toolkit is used by LETA, the National Library of Latvia, etc. to develop their own products and services. The treebank also serves as the basis for further linguistic and language technology research in the State Research Programmes "Digital Resources for Humanities" (2020-2022) and "Letonika – Fostering a Latvian and European Society" (2022-2024).

The derived UD treebank has also been used by foreign research teams to develop syntactic parsers for the Latvian language. It has been used for the training of the Google SyntaxNet parser, as well as in four international shared tasks for UD parser training, in which the created parsers for Latvian exceeded the accuracy rate of 85%; it was also used in

[6] CLARIN-LV repository http://hdl.handle.net/20.500.12574/63
[7] LINDAT/CLARIN-CZ repository http://hdl.handle.net/11234/1-4758

the creation of the widely used multilingual parser UDPipe[8] (Straka et al. 2016; Straka 2018). Thus, the results of the thesis have served as the basis for the creation of high-efficiency parsers for the Latvian language (see Section 0). The UD treebank is also being used in studies that develop metrics and tools more suitable for inflective languages (as opposed to the historically dominant English, which is an analytic language), such as CLAS metrics (Nivre, Fang, 2017).

1.8 Publications of Results of the Study

The thesis is designed as a set of publications, combining 11 publications of the author that address issues related to treebank creation and the development of parsers. The study was developed in the Artificial Intelligence Laboratory of the Institute of Mathematics and Computer Science (IMCS) of the University of Latvia (UL) within several projects and research programmes over the period of time from 2010 to 2023. The findings described in the thesis are the result of collective work, in which the author has lead the described study or significantly participated in the achievement of the given results (see table "Promocijas darba autora personiskais ieguldījums" on page 5 of the thesis).

The results of the study have been published in 7 publications indexed in the Elsevier Scopus and Thomson Reuters Web of Science databases.

- Saulīte, B., Darǵis, R., Grūzītis, N., Auziņa, I., Levāne-Petrova, K., **Pretkalniņa, L.,** Rituma, L., Paikens, P., Znotiņš, A., Strankale, L., Pokratniece, K., Poikāns, I., Bārzdiņš, G., Baklāne, A., Saulespurēns, V., Ziediņš, J. (2022). *Latvian National Corpora Collection – Korpuss.lv.* Proceedings of 13th International Conference on Language Resources and Evaluation (LREC 2022), Marseille, pp. 5123–5129 (*Scopus*).
- Gruzitis, N., **Pretkalnina, L.,** Saulite, B., Rituma, L., Nespore-Berzkalne, G., Znotins, A., Paikens, P. (2018). *Creation of a Balanced State-of-the-Art Multilayer Corpus for NLU*. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, pp. 4506–4513 (*Scopus* and *WOS).*
- **Pretkalniņa, L.**, Rituma, L., Saulīte, B. (2018). *Deriving enhanced Universal Dependencies from a hybrid dependency-constituency treebank.* Proceedings of the 21sh International Conference "Text, Speech, and Dialogue" (TSD), LNCS, Vol. 11107, Springer Link, pp. 95–105 (*Scopus* and *WOS).*

[8] *UDPipe* website: https://ufal.mff.cuni.cz/udpipe/2

- **Pretkalniņa, L.**, Rituma, L., Saulīte, B. (2016). *Universal Dependency Treebank for Latvian: a Pilot.* Proceedings of the 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2016), Frontiers in Artificial Intelligence and Applications, Vol. 289, IOS Press, pp. 136–143 (*Scopus* and *WOS).*
- **Pretkalniņa, L.**, Rituma, L. (2014). *Constructions in Latvian Treebank: the Impact of Annotation Decisions on the Dependency Parsing Performance.* Proceedings of the 6th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2014), Frontiers in Artificial Intelligence and Applications, Vol. 268, IOS Press, pp. 219–226 (*Scopus* and *WOS*).
- **Pretkalniņa, L.**, Znotiņš, A., Rituma, L., Goško, D. (2014). *Dependency parsing representation effects on the accuracy of semantic applications – an example of an inflective language.* Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, pp. 4074–4081 (*Scopus* and *WOS).*
- **Pretkalniņa, L.**, Rituma, L. (2012). *Syntactic Issues Identified Developing the Latvian Treebank.* Proceedings of the 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2012), Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 185–192 (*Scopus* and *WOS).*

The results of the study have also been published in 4 other international peer-reviewed publications:

- **Pretkalniņa, L.**, Rituma, L. (2013) *Statistical Syntactic Parsing for Latvian.* Proceedings of the 19th Nordic Conference of Computational Linguistics, NEALT Proceedings Series, Vol. 16, Oslo, pp. 279–289.
- **Pretkalniņa, L.**, Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011). *A Prague Markup Language Profile for the SemTi-Kamols Grammar Model.* Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011), Riga, pp. 303–306.
- **Pretkalniņa, L.**, Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011). *Towards a Latvian Treebank.* M.Á. Mora, M. Carrió Pastor, (ed.): Actas del 3 Congreso Internacional de Lingüística de Corpus. Tecnologias de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus, Candel, Valence, pp. 119–127.
- **Pretkalniņa, L.**, Levāne-Petrova, K., (2011). *Preparatory Work for Latvian Treebank.* Proceedings of International Conference CORPUS LINGUISTICS – 2011, St. Petersburg, pp. 53–58.

The author has presented the results of the study at 10 international conferences:

- 21st International Conference "Text, Speech, and Dialogue" (TSD), Brno, 2018;
- 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Riga, 2016;
- 6th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Kaunas, 2014;
- 9th International Conference on Language Resources and Evaluation (LREC), Reykjavik, 2014;
- 19th Nordic Conference of Computational Linguistics (NODALIDA), Oslo, 2013;
- 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT), Tartu, 2012;
- 17th International Scientific Conference "The Word: Aspects of Research", Liepaja, 2012;
- 18th Nordic Conference of Computational Linguistics (NODALIDA), Riga, 2011;
- International Conference CORPUS LINGUISTICS – 2011, St. Petersburg, 2011;
- 3rd International Conference on Corpus Linguistics, Valende, 2011.

Results related to the study have also been presented at two local conferences:

- 73rd UL conference "Latviešu valodas sintaktiski anotētā korpusa attēlošana universālā atkarību formātā", 2015;
- 72nd UL conference "Marķējuma transformācijas sintaktiski marķētā latviešu valodas tekstu korpusā", 2014.

Over the course of doctoral studies, the author of the thesis has also participated in the creation of the following international peer-reviewed publications that are indirectly related to the topic of the thesis:

- Paikens, P., Klints, A., Lokmane, I., **Pretkalniņa, L.**, Rituma, L., Stāde, M., Strankale, L., (2023). *Latvian WordNet*. Proceedings of the 12th Global Wordnet Conference (GWC2023), *to be published.*
- Paikens, P., Rituma, L., **Pretkalniņa, L.** (2022). *Towards Word Sense Disambiguation for Latvian.* Baltic Journal of Modern Computing, 10(3), 402–408 (*Scopus*).
- Paikens, P., Grasmanis, M., Klints, A., Lokmane, I., Pretkalnina, L., Rituma, L., Stade, M., Strankale, L. (2022). *Towards Latvian WordNet.* Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022), Marseille, pp. 2808–2815 (*to be indexed in the* Scopus *database*).

- Paikens, P., Gruzitis, N., Rituma, L., Nespore, G., Lipskis, V., **Pretkalnina, L.**, Spektors, A. (2019). *Enriching an Explanatory Dictionary with FrameNet and PropBank Corpus Examples.* Proceedings of the 6th Biennial Conference on Electronic Lexicography (eLex), pp. 922–933 (*Scopus*).
- **Pretkalnina, L.**, Paikens, P. (2018). *Extending Tezaurs.lv Online Dictionary into a Morphological Lexicon*. Proceedings of the 8th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2018), Frontiers in Artificial Intelligence and Applications, Vol. 307, pp. 120–125 (*Scopus*).
- Saulīte, B., **Pretkalniņa, L.**, Spektors, A. (2017). *Pirmās konjugācijas darbības vārdi Tēzaurā.* Vārds un tā pētīšanas aspekti : rakstu krājums 21 (1/2), Liepāja, 122.–129. pp.
- Spektors, A., Auzina, I., Dargis, R., Gruzitis, N., Paikens, P., **Pretkalnina, L.**, Rituma, L., Saulite, B. (2016). *Tezaurs.lv: The Largest Open Lexical Database for Latvian.* Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, pp. 2568–2571 (*Scopus*).
- Paikens, P., Rituma, L., **Pretkalniņa, L.** (2013). *Morphological Analysis with Limited Resources: Latvian Example.* Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), NEALT Proceedings Series, Vol. 16, Oslo, pp. 267–277.

2 A Hybrid Grammar Model of the Latvian Language and Latvian Treebank

A treebank is an essential language resource that enables the creation of a data-driven syntactic parser, as well as provides an opportunity for new ways of linguistic research.

During the creation of a syntactic analyser i.e. parser, information on the syntactic structure of the analysed language is collected; i.e., it is necessary to define the model in which the syntactic information will be represented, and to gather data on the nature of a particular language. Once the model of the syntactic analysis is formulated, the information on of the peculiarities of a language can be assembled and compiled both in person, in the form of rules, or by a computer through machine learning (e.g. using statistical methods or neural networks) from an accordingly prepared dataset.

Before the start of this thesis, the Latvian language did not have a treebank and subsequently lacked the possibility of developing statistic parsers. However, the approach of building a rule-based parser had been attempted in the State Research Programme "Scientific Foundations of Information Technologies", in the project "Research and Development of the Semantic Web Technologies for Latvia" (*SemTi-Kamols*) (2005–2009), in which a hybrid Latvian syntax model was based on the dependency grammar theory by Lucien Tesnière (Tesnière, 1959) (see Section 2.1 for more details). On the basis of this syntax model, the syntactic analyser *Čankeris* was later created (Bārzdiņš et al., 2007; Nešpore et al., 2010). *Čankeris* was a partial parser that was able to analyse most of the simple extended sentences, as well as clauses corresponding to simple sentences in compound sentences through a brute force search by using rules written by linguists. For each processed sentence, *Čankeris* provided all formally possible analysis variants, since at the time of its creation there was no existing Latvian treebank available, which would allow to evaluate the more likely variants. With such an approach, the number of proposed analysis variants tends to increase exponentially depending on the length of the sentence.

The examination of the results generated by *Čankeris* revealed a number of problems inherent in many rule-based parsers. The most significant of these problems is scaling difficulty: as the number of rules increases, the average number of analysis options for a sentence also increases significantly (exponentially); because of this the processing speed decreases and the interaction of rules becomes more complex, problematic to track and more difficult to debug.

With the increase in the number of proposed analysis variants, one is faced with the perplexing question of how to choose one analysis variant from many others and offer it to the user or utilise it further. A reasonable course of action would be to choose the right variant from all the options, or, if there is no right variant, select the least erroneous one. However, in language there is a high probability of situations when several grammatically correct analysis variants are possible within the given text fragment and the expected human interpretation of it is not unambiguously determinable only based on the knowledge of grammar. For example, in the case of the compound nouns *sieviešu ādas zābaki* ('women's leather boots') and *liellopa ādas zābaki* ('cow leather boots'), it is the real-life knowledge that helps one conclude that in the first case the boots are most likely intended for women, whereas in the second case the boots are made of cowhide, and therefore not intended to be worn by cattle. Integrating broad-scale real-life knowledge into rule-based parsers would be extremely difficult, but statistical data on what structures are more common for a particular language material (e.g. that "women's boots" is a much more commonly used wording than "cow boots") can help achieve better results. Thus, the conclusion was made that even if other scalability problems were solved, the further development of *Čankeris* into a wide-coverage, workable syntax parser would nonetheless require a treebank from which to derive such statistical information.

On the other hand, if one chooses to build a parser on the basis of statistical regularities calculated from data or neural networks trained on data, a treebank that serves as a source of these regularities immediately becomes a fundamental and basic digital language resource for this.

A 2012 META-NET review study (Vasiļjevs, Skadiņa, 2012) indicated that not a single treebank is publicly available for the Latvian language. Therefore, the creation and publication of such a resource, especially in the form of open data, provides new opportunities not only for the research and development of language technologies, but for the studies and research of linguistics as well. Such a resource allows linguists to test their understanding of theoretical information on syntactic phenomena of the Latvian language in practice and to improve it according to data-driven observations.

2.1 The Treebank's Grammar Model

Based on the more successfully applied aspects of the hybrid grammar model in *Čankeris* created in the *SemTi-Kamols* project, as well as consultations with linguists, it was decided to further develop this model for

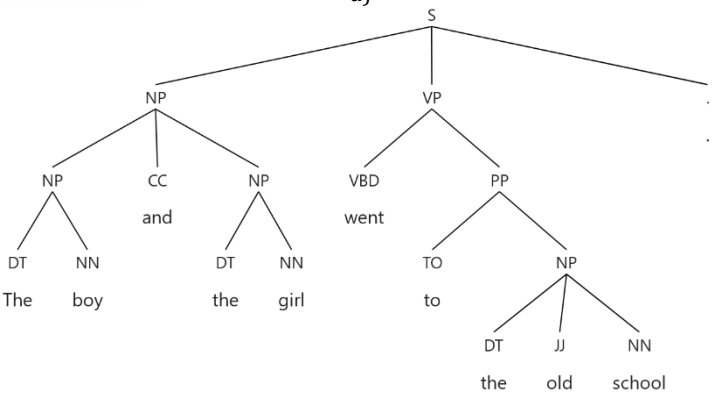
the needs of the treebank with the long-term goal of modelling all the Latvian constructions found in the prospective corpus.

The grammar model of the Latvian Treebank (LVTB model) is designed as a hybrid of two syntactic models widely used globally: dependency grammar and phrase structure grammar.

From a mathematical viewpoint, the formal representations of sentence structure are graphs, therefore terminology from graph theory is used further on in the thesis. The graph is made up of two types of elements – a finite number of **nodes** and **edges** that form pairs from the nodes (each pair of nodes is connected by no more than one edge). Most often, the structure of the sentence is a **rooted tree** – a graph without cycles with one specifically marked node that is called **the root**. The nodes attached to any given node are divided into two groups: parents and children. A **parent** is a node whose distance (in terms of the number of edges) from the root is less than the distance of a given node; there is exactly one parent for each node of the tree (except the root, which is the only node that does not have a parent). The other nodes are called **children** – they are located at a greater distance from the root than the given node. Nodes that do not have children are called **leaves**. Nodes that are the parents of the given node, parents of parents, etc. (the transitive closure of being a parent) are called the **ancestors** of the given node. Nodes that are the children of a given node, children of children, etc. (the transitive closure of being a child) are called the **descendants** of the given node. In some cases, a graph representing the structure of a single sentence may be disjoined i.e., the graph may be a cluster of several trees – **a forest**; however practical implementations often try to avoid such structures.

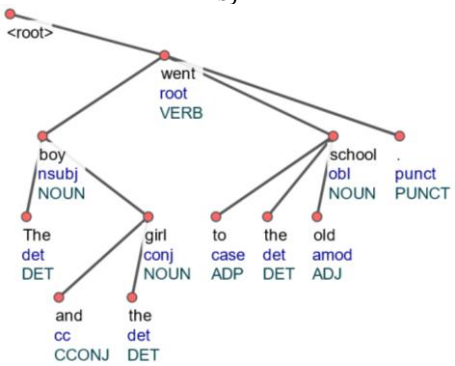
The phrase structure grammar model (Chomsky, 1957) considers a phrase to be the basic unit of sentence structure. In this model, each phrase consists of **tokens** – words or punctuation marks – and/or other phrases, thus a sentence is modelled as a tree, the root of which is the phrase corresponding to the entire sentence, and the leaves are tokens. An example of such a tree is given in Figure 2a. In the basic model, phrases are required to be **continuous**, i.e., if the subtree with the phrase X in its root contains both the i-th and j-th tokens in a sentence, then all the tokens within the interval $[i, j]$ must also belong to the same subtree. In practice, this condition tends to be overlooked in order to maintain the homogeneity of annotation for similar linguistic phenomena. This is the case with the paraphrased fragment of text *skola, uz [kuru] viņi gāja* ('the school they went to'): if the goal is to maintain phrase division equivalent to the example in the image, **a discontinuous phrase** must be formed, where between the verb phrase (VP) parts *gāja uz* ('went to'), and *skola* ('school'), the word *viņi* ('they') is located.

a)



Source: *Berkley Neural Parser* demo https://parser.kitaev.io/

b)



Source: *UDPipe* parser demo http://lindat.mff.cuni.cz/services/udpipe/

Figure 2: An example of sentence annotation according to the phrase structure grammar model (a) and the dependency grammar model (b).
Sentence: *Zēns un meitene gāja uz veco skolu.* 'The boy and the girl went to the old school.'

Dependency grammar (Mel'cuk, 1988), on the other hand, considers a word to be the basic unit of sentence structure and therefore models the sentence using directed, binary relations between words, i.e. **dependencies**. The predicate is considered the central element of a simple sentence structure; this predicate can have dependents, and each dependent can have further dependents as well. The structure of the sentence is depicted as a tree, with each of its nodes corresponding to one word. For corpus annotation purposes, this model can be supplemented so

that the tree includes nodes containing all of the tokens, including punctuation marks, that make up the sentence (Hajič et al., 2000). An example of such a tree is given in Figure 2b. A dependency grammar based model is used by Lithuanian (language typologically closest to Latvian) treebank ALKSNIS[9] (Bielinskienė et al., 2016), however, the development of the mentioned treebank started later – after the model for the Latvian treebank had already been chosen.

The origins of modern dependency grammar can be traced back to Tesnière's grammar model (Tesnière, 1959). In his work, apart from dependencies, other relations within a sentence are also defined, such as the **transfer** (Fr. *translation*) operation, when a preposition and its corresponding content word form a single 'virtual"' word, which functions as a special case of the content word further on in the sentence. A separate construction – a ***jonction*** (Fr.) – represents coordinated parts of a sentence. The nodes of the tree in this model are considered not to be words, but ***nucléus*** (Fr.)., namely, elements that can be either a single word or a combination of several words obtained through the operations described above.

Phrase structure grammar and dependency grammar annotations are mutually comparable on a formal level and can be converted from one annotation model to another, as long as a head word is specified for each phrase in the phrase structure annotation. As a result, each phrase can be transformed into a set of dependencies between the head word and the other elements of a phrase, while each dependency can be perceived as a two-element phrase, in which the independent element of the dependency becomes the phrase's head word. Discontinuous phrases in phrase structure grammar correspond to non-projective edges in dependency grammar. According to Nivre, Nilsson (2005), if the node (token) pairs *v* and *w* are connected by an edge, it is considered a **non-projective edge** if and only if any of the tokens between *v* and *w* in the sentence are not descendants of neither *v* nor *w*. **A non-projective tree** (as opposed to a projective one) is a tree that contains at least one non-projective edge. It is easy to determine that, when transforming between models in the manner described above, a non-projective edge becomes a discontinuous phrase and vice versa.

At its initial stage of development, the hybrid grammar model of *SemTi-Kamols* employed dependencies and x-words – Tesnière's *nucléus*-like constructs for representing some word groups (Nešpore et al., 2010). In this doctoral thesis, the hybrid grammar model has been significantly expanded and refined to eliminate the shortcomings that were identified during the treebank annotation process.

[9] CLARIN-LT repository http://hdl.handle.net/20.500.11821/21

The improved LVTB model (Pretkalniņa et al., 2011a; Pretkalniņa, Levāne-Petrova, 2011) is a hybrid in relation to the previously described syntax models – it represents sentence structure as a tree of dependencies, in which some of the nodes may also be phrase-like constructions (see example in Figure 3). These constructions can have common dependents that are related to the entire structure, as well as individual dependents that relate to an individual component of the construction. Each component of the phrase is either a token or another phrase, therefore the LVTB hybrid model is a generalisation, the instances of which can be considered both the phrase structure model and the modern model of dependencies.

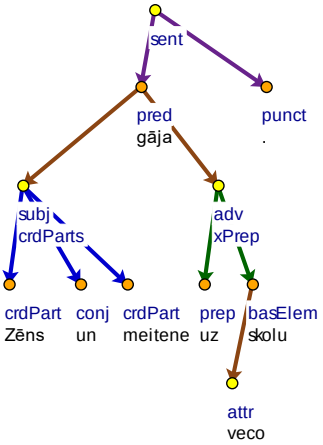


Figure 3: An example of sentence annotation according to the LVTB model, legend: green edges – components of the x-word (phrase), blue edges – components of the coordination construction (phrase), purple edges – components of punctuation mark construction (phrases), brown edges – dependencies

When annotating the treebank, the dependency relations of the hybrid model are used to represent syntactic relations of subordination, such as *maza <u>māja</u>* ('small <u>house</u>'), *<u>iet</u> skolā* ('<u>go</u> to school'), *ilgi <u>gulēt</u>* ('<u>sleep</u> for long') (the independent component is underlined).

Phrase-style constructions are divided into three groups in the model: x-words, coordination constructions, and punctuation mark constructions.

X-words currently represent constructions that are expressed in Latvian by analytic forms, such as complex predicate and prepositional constructions. These constructions are characterised by a strict internal word order, as well as a precise number and type of elements. For example, preposition constructions consist of one preposition and one nominal, and

the type of preposition determines whether it is placed before the nominal or after it – such Latvian prepositions as *ap* ('around'), *uz* ('on', 'to'), *pār* ('over') are used before the nominal, whereas *dēļ* ('because of'), *labad* ('for the sake of') are used after the nominal (although there are also certain prepositions that can be found in both positions in the actual language material, e.g. *dēļ* ('because of') and *pēc* ('by', 'after')). Such reinterpretation clarifies the idea of x-words in the initial *SemTi-Kamols* model: even though the x-words in the original *SemTi-Kamols* model included some additional constructions, e.g. coordinated parts of a sentence and the main word of participial clause, the authors of the original *SemTi-Kamols* model believed that the inclusion of all punctuation marks in x-words did not correspond to the intended idea.

In Latvian, punctuation marks help evaluate grammatical structure, therefore it was decided that punctuation should also be annotated in the treebank, if possible, noting the motivation for the use of the punctuation marks as well. As a solution to this problem, a second type of phrase-like constructions was defined, namely, the punctuation mark construction (PMC). This type of construction consists of a base element – a word or a phrase (most often one) – and punctuation marks used in the sentence because of that base element. As an example, this sentence can be examined: *Viņš, ēzdams ķiršus, nosmērēja kreklu* ('While eating cherries, he got his shirt dirty.'). There is a PMC that contains a base element (word) and two commas, which are used because of the participial clause formed by the participle *ēzdams.*

In the improved LVTB model, another type of phrase-like construction is used: similar to the original Tesnière's dependency grammar, a separate construction for coordinated elements is distinguished here. This construction is consistently used in the LVTB model both to represent coordinated parts of a sentence, such as *boy and girl,* and to represent coordinated clauses, for example, *zēns ir mājās, bet meitene iet uz skolu* ('the boy is at home, but the girl is going to school') and *zēns ir mājās, jo viņam šorīt bija iesnas un māte atļāva neiet uz skolu* ('the boy is at home, because he had a runny nose this morning and his mother allowed him not to go to school'). In some aspects, coordination constructions are structurally similar to the two phrase types described above, however, the structure of the coordination construction is not as strictly fixed as the structure of x-words, and may also contain punctuation marks as they separate coordinated parts of a sentence or clauses.

The LVTB model also supports the following additional information: dependency relations have different types or syntactic roles (e.g., adverbial modifier, attribute, grammatical object, situant, determinant, etc.), and each kind of phrase-style construction has several subtypes. X-words are

categorised according to the differences in their structure (prepositional constructions represent one type, complex predicates represent another), whereas PMCs are categorised according to the type of base element and the motivation for the use of punctuation marks. The types of coordination constructions show whether the annotated construction represents coordinated parts of a sentence or clauses.

Defining the formal structure of the model was finished in the first month of treebank annotation, meanwhile the annotation of certain language constructions is being continually fine-tuned by linguists working on the treebank throughout the annotation process, which may result in changes in the phrase-style construction and dependency types; in such cases the technical solutions must be easily adjustable. Over the course of treebank annotation, one must face the shortcomings of the syntax theory of the Latvian language, including various vague borderline cases and other phenomena incompletely described in the theory. Some of the theoretical problems identified in the formation of the treebank have been summarised (Pretkalniņa, Rituma, 2012) and the linguistic details of the model have been described in more detail (Rituma et. al, 2019). In general, it has been observed that the model has been stabilised after many development iterations, so it is considered complete.

2.2 The Creation of the Treebank

This section describes the creation of Latvian Treebank (LVTB). Firstly, the necessary tools and data formats have been described, followed by the description of the process of creating the treebank itself.

2.2.1 Tools and Data Formats for Creating the Treebank

Since the LVTB data should be easy to store, browse and edit, it was necessary to create or adapt the support tools and data storage formats that would ensure this.

As the basic toolkit for manual processing of the treebank, the *TrEd* toolkit developed at Charles University (Hajič et al., 2001) and the Prague Markup Language (PML) data metaformat (Pajas, Štěpánek, 2006) were used. The native format of this toolkit is PML. The described decision was made due to the wide functionality of *TrEd* toolkit and PML – among others, a visual editing tool for *TrEd* has been developed, a search language for tree-like structures PML-TQ with implementation already exists (Štěpánek,

Pajas, 2010), and a batch processing tool *bTrEd*[10] is also available. These tools have been practically appraised by annotating several corpora, including large ones, such as Prague Dependency Treebank (PDT) (Hajič et al., 2000), Prague Arabic Dependency Treebank (Hajič et al., 2004), Slovene Dependency Treebank (Jeroski et al., 2006), among others. Furthermore, Charles University offers the publishing services of such data as part of the LINDAT/CLARIN initiative[11].

It is also a valuable addition, that the PML standard allows the division of annotations attached the text into several levels and the storage of the data of each level in a separate file; this allows morphological annotation (within one token) to be stored on one level and syntactic annotation on another, thus creating a single storage standard for both morphologically and syntactically annotated corpora. It would be easier to add new levels of annotation to such a structure if the need arises in the future to supplement the treebank with a higher level of annotations. Furthermore, PML is also relatively easy to use outside of the TrEd toolkits as PML is a subformat of XML (eXtensible Markup Language) metaformat, therefore PML can be viewed with XML editors and processed with XML processing tools as well. This, in turn, ensures the usability of data in this format for future tasks that have not yet been defined.

Although the *TrEd* toolkit is postulated as a universal set of tools for working with any kind of the previously described syntactic tree structures, generally it has so far been used specifically for dependency corpora, so in order to adapt it to LVTB needs, a new PML data format was defined (a specific format that meets the general PML specification i.e. metaformat) and designed specifically for annotating LVTB corresponding to the model described in Section 2.1. By analogy with PDT, the annotation of the future Latvian Treebank is also divided into three levels: tokenization, morphological annotation and syntactic annotation. The levels of tokenization and morphological annotation are designed to be as close as possible to the corresponding levels of PDT, thus adopting approved practices and providing easier comprehensibility for researchers who have previously worked with PDT. The syntactic annotation level is based on the PDT analytic level format, which represents dependency tree structures. For the purpose of adapting it to LVTB needs, the format is supplemented with

[10] *bTrEd* technical documentation
https://ufal.mff.cuni.cz/pdt2.0/doc/tools/tred/btred.html
[11] Treebank Repository of Charles University:
https://lindat.mff.cuni.cz/repository/xmlui/discover?filtertype=subject&filter_relational_operator=equals&filter=treebank

phrase-style structures and empty nodes (not corresponding to any token; PDT does not have such nodes) to depict cases of word omission (ellipsis). Supplementation with phrase-style nodes has also been carried out in practice by introducing a special type of empty auxiliary nodes, which serve as representations of a phrase as a single, unified unit.

In order to be able to use this format adequately, its properties were defined in PML schemas corresponding to the PML Schema Standard[12]. An extension for the graphic editor *TrEd* was also created (Pretkalniņa et al., 2011b). The extension contains a variety of features accessible through keyboard shortcuts that make manual syntactic annotation more convenient, as well as a style sheet visualisation used by both *TrEd*, and the Charles University CLARIN/LINDAT repository, which offers the possibility of keeping this type of data publicly available and accessible to any interested party.

Additionally, the XSL (*Extensible Stylesheet Language*) transformation was created, enabling the conversion of corpus data into *Tiger XML* format (Mengel, Lezius, 2000), which is used by *TigerSearch*[13] and other tools developed by the University of Stuttgart.

2.2.2 Iterative Creation and Development of the Treebank

The annotation of LVTB started in 2010. Within various projects, the size of the treebank has increased to 17 thousand sentences in 2022 and thus has become an important resource for computational linguistics of the Latvian language.

When describing the development of LVTB, one can define two general phases: the pilot project of initial annotation and the massive expansion phase of the treebank starting from approximately 2016.

Initially, in the pilot project, the treebank was annotated by one syntax specialist and the possibilities for automated pre-annotation were minimal. Thus, over several years, the treebank reached the size of approximately 1500 sentences, but after the creation and integration of the morphological tagger (Paikens et al., 2013) into the annotation process, the size of the treebank increased to 5000 sentences in 2014 (Rituma et. al, 2019).

In 2016, the massive expansion of LVTB began. During the ERDF project "Full Stack of Language Resources for Natural Language

[12] *PML Schema* is defined in the PML specification, the Section 6:"PML schema file" https://ufal.mff.cuni.cz/jazz/pml/doc/pml_doc.html#pml-schema

[13] *TigerSearch* website: http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/tigersearch.html

Understanding and Generation in Latvian" (2016-2019) (see also Section 3.2.1), the corpus reached the size of 10 thousand sentences. After that, the quantitative and qualitative expansion of the LVTB was carried out within the State Research Programmes "Digital Resources for Humanities" (2020-2022) and "Letonika – Fostering a Latvian and European Society" (2022-2024), and at the time of writing the thesis, the treebank has reached the size of 17 thousand sentences. LVTB has served as a basis for various research studies (e.g., Lokmane, Saulīte, 2023a; Lokmane, Saulīte 2023b, Rituma et al. 2023), thus, helping to prove the 1st hypothesis of this thesis seeing as a hybrid grammar model better corresponds to Latvian linguistical tradition and therefore is easier to perceive for researchers in Latvia.

An approximate insight into the significance of the resulting size of the treebank can be gained by looking at both historically important corpora and the sizes of treebanks of other languages in the spring 2022 release[14] of the Universal Dependencies (UD) initiative (see Chapter 3 for detailed information on UD). In this UD release, the size of UDLV-LVTB has reached 282 thousand tokens, and the release contains a total of 32 corpora, the volume of which is at least 250 thousand syntactic words[15] each. If one measures the total corpus size for languages that have multiple UD corpora, out of the 130 languages included in the release, 30 have corpus sizes of at least 250 thousand syntactic words. In this version the largest individual UD treebanks have been created for German – *UD_German-HDT* (3,455 million synt. w.), Russian – *UD_Russian-SynTagRus* (1,517 million), Czech – *UD_Czech-PDT* (1,509 million), Japanese – *UD_Japanese-BCCWJ* (1,253 million) and *UD_Japanese-BCCWJLUW* (995 thousand), as well as Icelandic – *UD_Icelandic-IcePaHC* (985 thousand).

In 2010, when the research related to this doctoral thesis was commenced, it was much more difficult to compare the world's largest corpora, as they were annotated according to different formal grammatical models (including a slightly different tokenization system), but the three most significant corpora were the already mentioned Czech PDT (1.5 million) (Hajič et al., 2000), as well as the English Penn Treebank (3 million, pioneer of Treebank studies) (Marcus et al., 1994) and the German Tiger Treebank (900 thousand) (Brants et al. 2002). On the one hand, the experience of other languages also shows that the need for such large

[14] UDv2.10, available in the LINDAT repository http://hdl.handle.net/11234/1-4758

[15] The UD annotation paradigm defines that for certain languages some tokens should be divided into several syntactic words, for example, the English abbreviations *mom's* and *don't* are considered to be two syntactic words each, so in some language corpora the number of such 'words' exceeds the number of tokens. For Latvian corpus, however, these numbers match.

treebanks has diminished. The most plausible reasons for this are the rapid development of deep neural network language models, as well as the growing use of transfer learning methods, which help improve the modelling of the target language by using data from other languages. On the other hand, according to the global experience in the currently most active international community of computational linguistics – UD, a scientifically valuable resource of a suitable size has been created for the Latvian language, which is also reinforced by the results of several shared parser building tasks (see Section 0).

2.3 The Importance and Impact of the Treebank

This section describes the publication of LVTB and the further use of the resource in developing parsers.

2.3.1 Availability

The data of the treebank is annotated in a publicly inaccessible repository, but the corpus itself is available in the form of publicly released versions. Since 2018, the versions of the treebank have been published every six months in the Charles University LINDAT/CLARIN repository according to the UD release schedule, together with the derived UD corpus of the Latvian language [16] described in Chapter 3. There, the persistent linking of the treebank, as well as an online search option is provided, thus also ensuring the validity of these corpora for scientific citation. The first published version of the treebank contains 7.7 thousand sentences, the latest (November 2022) – almost 17 thousand sentences [17]. In order to inform Latvian researchers about this resource, informative seminars [18] have also been organised.

2.3.2 Corpus-Driven Parser Research

Given the rapid development and widespread use of UD treebanks (see Chapter 3), the description of the studies included in this section has a

[16] All published versions of LVTB, as well as the versions of the UD treebank described in Chapter 3 are available in the following repository section: https://lindat.mff.cuni.cz/services/pmltq/#!/treebanks, by selecting Latvian

[17] A brief description of the corpus versions and a summary of the used annotations are also available on the IMCS UL resource http://sintakse.korpuss.lv/versions.html

[18] For example, via the CLARIN initiative: https://www.clarin.lv/lv/clarin-latvija-seminari/31-clarin-seminars-par-latviesu-valodas-sintaktiski-marketa-korpusa-izmantosanu

more retrospective meaning, since the unifying representation that the UD ensures has opened up opportunities for many languages to use common tools and uniform methods of building parsers without carrying out feature extraction for each language. Additionally, due to the development of the research field, new technologies for constructing parsers have emerged, and international groups of researchers have also been involved in the development of parsers for Latvian (see Section 0).

The studies described here played an important role in the creation of the UD treebank described below: the problem of treebank transformations was addressed and an initial insight was gained into the impact of the dependency representation on the operation of parsers and their downstream application. Furthermore, the varied LVTB dependency representations available through the transformations confirm the 1st hypothesis of this thesis, namely, that a hybrid model would expand the usability of treebank compared to a dependency model. The data transformations that were recognised as potentially useful in these studies are in many respects similar to the UD annotation model, thus highlighting the creation of the UD treebank as a logical continuation of the work.

At the same time, it should be noted that the latest research developments at the time of completion of this thesis indicate a re-actualization of hybrid models. Nivre et al. (2022) have examined a broad set of typologically different languages (including the Latvian language UD treebank described in Chapter 3) and cited the publication (Barzdiņš et al. 2007), which also served as the basis for the model described in Section 2.1. They have concluded that the use of Tesnière's *nucleus* elements in the developments of parsers provides a small but statistically significant improvement in parsing accuracy.

The following Section 2.3.2 is structured as follows: the comparative parser studies are described in Section 2.3.2.4, while the prerequisites for these studies are explained in Sections 2.3.2.1 – 2.3.2.3; first, Section 2.3.2.1 describes the choice of the parser system required for research (corresponding to the cutting-edge methods of the time), then Section 2.3.2.2 defines the metrics for comparing results and, finally, Section 2.3.2.3 describes the data transformations required to enable the use of the chosen parser system.

2.3.2.1 Selection of a Parser Training System

When creating a data-driven parser, it is necessary to align the data annotation model with the one used in the parser. Since the LVTB model is a hybrid model and it does not have a parser training system in place (as this type of syntax models have been little used so far), the possibilities were

considered for converting the data annotated according to the hybrid model to be used for an already developed parser training system.

In line with the models widely used in treebank annotation, parser training systems also essentially employ either a phrase structure grammar or a dependency grammar model. Although parser systems that use the phrase structure grammar model (Collins, 2002) have been known for a long time, at the start of this phase of the study (2012), the best results were given by parsers based on dependency grammar (Bohnet, Nivre, 2012), which simultaneously perform a syntactic and morphological analysis.

At the beginning of this phase of the study, the manually annotated treebank of the Latvian language was still relatively small (53,225 tokens) and there was a larger, separate (manually) morphologically annotated corpus (109,311 tokens, including the treebank). Therefore, it was decided to initially carry out morphological tagging and syntactic parsing sequentially, rather than simultaneously, in order to be able to generate a higher accuracy rate for morphological tagging, since it would be possible to use almost three times as much data for the training of the morphological tagger. Higher accuracy of morphological tagging also improves the accuracy of syntactic parsing.

Parser systems based on the dependency model were found to be more effective not only because of the high accuracy at the time, but also because certain systems that are based on the dependency model implement non-projective tree processing with an algorithm specially designed for that purpose (Nivre, 2009). In the case of the Latvian language, this aspect is important because non-projective trees are formed more often for languages with a rather free word order (McDonald et al., 2005). As an alternative, the possibility of using a dependency parser that processes only projective trees should be considered, however, in that case, projectivization transformation would be a necessary step for the pre-processing and post-processing of data (Nivre, Nilsson, 2005).

In the parser systems that are based on the of dependency model, two main methods were used for parsing at the time – transition-based parsers (Nivre, 2009) and graph-based parsers (Koo, Collins, 2010), however, in the context of these studies, the main conceptual disadvantage of graph-based parsers is the inability to construct non-projective trees.

In addition, transition-based parsers have two practical advantages. First of the advantages is the performance: for the graph-based parser, the asymptotic estimate of execution time versus sentence length is $O(n^3)$ or $O(n^4)$, depending on the complexity of the employed substructures (Koo, Collins, 2010), whereas for the transition-based parser the estimate is $O(n)$ for projective parser (Nivre, 2009) and $O(n^2)$ for non-projective parser

(Nivre, 2003). Secondly, a parameter optimisation tool *MaltOptimizer*[19] (Ballesteros, Nivre, 2012) has been developed for the widely used transition parser system *MaltParser*[20] (Nivre et al., 2007b); this tool automates the rereading of machine learning parameters, thereby reducing the amount of work required to successfully train the system.

2.3.2.2 Parser Evaluation Metrics

The following metrics are used to compare and evaluate parsers.

Unlabelled attachment score (UAS) indicates the fraction of the total tokens for which the parent in the tree specified by the parser coincides with the assigned standard e.g., annotated by a specialist.

Label accuracy (LA) indicates the fraction of the total tokens for which the given role (as the dependent) coincides with the assigned standard e.g., annotated by a specialist.

Labelled attachment score (LAS) indicates the fraction of the total tokens for which the given standard coincides with both the parent and the role (as the dependent) assigned by the parser.

Unless otherwise indicated, the metric values in this thesis are given as a percentage.

2.3.2.3 Data Transformations

In order to use LVTB data for the training of *MaltParser*, it was necessary to convert the annotations both from hybrid syntax model to the dependency model and from PML XML to the *CoNLL* tabular data format[21]. The tools created for this aim are available online[22]. This section focuses on the transformations from the LVTB model annotations to that of dependency annotations (Pretkalniņa, Rituma, 2014), as data format transformations are relatively simple – the necessary fields must be extracted from PML XML files and recorded in a tabular format according to the requirements of the *CoNLL* format.

In order to transform the corpus annotation, it is enough to define how these basic constructions of the LVTB model need to be transformed:

- dependency relations between tokens;

[19] *MaltOptimizer* website: http://nil.fdi.ucm.es/maltoptimizer/install.html
[20] *MaltParser* website: http://www.maltparser.org/
[21] *CoNLL-X Shared Task: Multi-lingual Dependency Parsing* website archived at *Wayback Machine*: https://web.archive.org/web/20160814191537/http://ilk.uvt.nl/conll/#dataformat
[22] *GitHub* repository *CorporaTools*: https://github.com/LUMII-AILab/CorporaTools

- dependency relations in which the parent and/or child is a phrase-style construction;
- phrase-style constructions.

Using the basic concepts described in Section 2.1, it was decided to use a conversion strategy that meets the following requirements:

- the dependency relations between tokens before and after conversion are identical;
- phrase-style constructions are converted to fragments of the dependency tree that are rooted trees;
- in dependency relations, in which the parent or child is a phrase-style construction, the phrase-style construction(s) is replaced with the root node of the tree fragments created in the previous point.

When transforming to a dependency representation, information that is no longer represented structurally is lost or encoded into roles. By balancing information loss with the complexity of the role set, the following role encoding system is created:

- the annotation of the dependency edges is supplemented with a prefix indicating whether the dependency head in the original annotation is a token (prefix *dep*) or a phrase-style construction (prefix *phdep*);
- for an element that becomes a root of the subtree representing a phrase-style structure, a composite role is formed, consisting of:
    1. a prefix indicating whether the structure in question depends on a token (prefix *dep*) or on the phrase-style construction (prefix *phdep*) in the original annotation, as well as a dependency role that was assigned to the phrase-style structure in the original annotation;
    2. a phrase type and role of the element within the phrase (this part of the role is not included in cases where the role that occurs in Point 1 is syntactically viable for that element even when the element is not included in the phrase);
- the remaining components of the phrase are annotated with a compound role, consisting of the type of phrase and the role of the element in the phrase;
- information on the empty nodes representing word omissions is discarded.

An example of the transformation can be seen in Figure 4. The given description defines the general characteristics chosen for the transformation, but in order to obtain a fully defined transformation, it is necessary to indicate how each of the phrase-like construction types used in the corpus should be transformed.

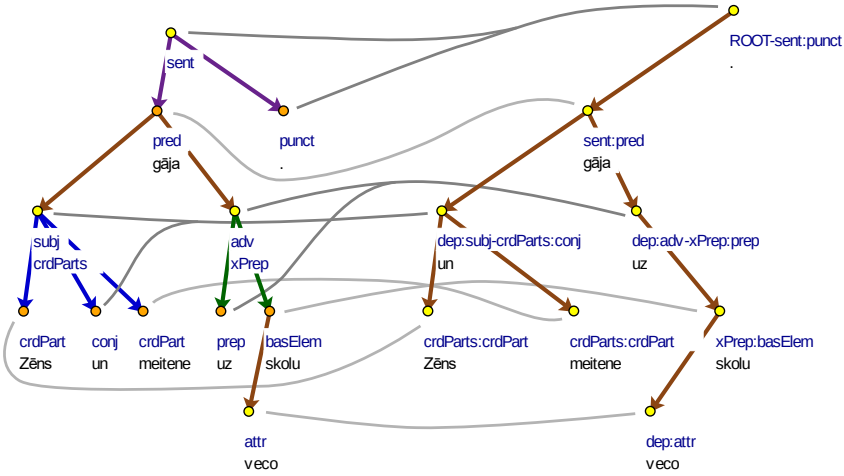


Figure 4: An example of a syntax tree transformation

After consulting linguists[23], transformations were unambiguously defined for most phrase-style constructions, however, certain individual groups of phrase-style constructions were also identified in which linguistic knowledge does not give a clear-cut decision on the optimal method of transformation. LVTB has four such groups:

- complex predicates (denoted by the x-word type *xPred* in the treebank);
  - perfect verb tenses e.g., *ir gājis* ('have gone');
  - modal predicates e.g., *varēja gulēt* ('could sleep'), *gribēja ēst* ('wanted to eat'), *gadījās pakrist* ('happened to fall');
  - compound predicates e.g., *ir gudrs* ('is smart'), *bija skolotājs* ('was a teacher'), *būs auksti* ('[it] will be cold');
- coordinated parts of a sentence and clauses (in the treebank – all types of coordination constructions);
- phrase-style constructions aimed at including punctuation marks into the tree (in the treebank – all punctuation mark construction (PMC) types), for example:
  - PMCs of subordinate clauses,
  - PMCs of insertions and parentheses,
  - PMCs of address.

[23] Special thanks to Laura Rituma, Baiba Saulīte, Gunta Nešpore-Bērzkalne and Ilze Lokmane.

Transformational alternatives were created for these constructions to further study the impact of decisions on the design and usability of parsers.

2.3.2.3.1 Transformations of Complex Predicates

Each complex predicate contains one base element (*basElem* role in the corpus), which denotes the semantically dominant element, as well as one or more auxiliary verbs (*auxVerb* role in the corpus) and/or modifiers (*mod* role in the corpus). For example, in a phrase *ir skolotājs* ('[he/she] is a teacher'), the noun *skolotājs* ('teacher') is the semantically dominant word and therefore the base element, whereas *ir* ('is') is an auxiliary verb. Following the same logic, in the phrase *peldēt gribi* ('[you] want to swim') the verb *peldēt* ('swim') is the base element. Given that the construction allows these and only these components, it was decided to consider the following transformations (see Figure 5):

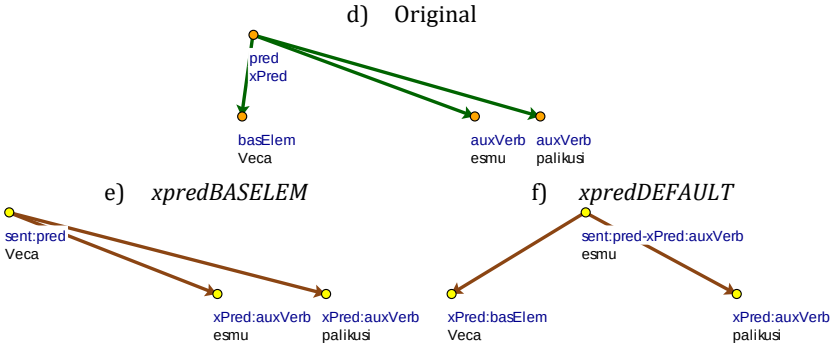


Figure 5: Examples of complex predicate transformations, legend: green edges – components of the x-word (phrase), brown edges – dependencies

- *xpredBASELEM*[24] – an element with the role *basElem* is selected as the root of the subtree corresponding to the phrase, whereas the other elements of the phrase (in the case of a correct tree – with the roles *auxVerb*, *mod*) are made dependents of the selected root element (see Figure 5b);
- *xpredDEFAULT* – the linearly first element (i.e. first relative to the order of tokens in the text) with the role *auxVerb* or *mod*, is selected

[24] Here and further, the names of the transformations are coordinated with those used in papers and in the *CorporaTools* repository.

as the root of the subtree corresponding to the phrase; the rest are made dependents of the selected root element (see Figure 5c).

The *xpredDEFAULT* transformation could be improved by refining the criteria for selecting the root element in the case of multiple *auxVerb* and/or *mod* roles, since the choice of the first linear element does not ensure that the chosen phrase element will represent the same relation. For example, in the phrase *esmu palikusi veca* ('[I] have become old') the verb *esmu* ('have') would be chosen as the first linear element, whereas in the phrase *veca palikusi esmu* (different word order), the verb *palikusi* ('become') would be chosen. However, these improvements were not possible to carry out in this phase of the study without manually supplementing the treebank with additional annotations that would help make such a choice.

2.3.2.3.2 Transformations of Coordination Constructions

Coordination constructions consist of two or more coordinated elements (with the corpus role *crdPart*) separated by punctuation marks (with the corpus role *punct*) and/or conjunctions (with the corpus role *conj*). For example, in the phrase *sēdēja, ēda un dzēra* ('sat, ate and drank') the coordinated elements are the verbs *sēdēja* ('sat'), *ēda* ('ate'), *dzēra* ('drank'), the last two coordinated elements are separated by a conjunction *un* ('and'), while the first two are separated by a punctuation mark (comma). The conjunction can also be located before the first coordinated element, as in the example *gan zēni, gan meitenes* ('both boys and girls').

The annotation of coordination constructions varies widely in different dependency corpora, differing both in general decisions and nuances. Popel et al. (2013) proposes to divide the types of annotation of coordination constructions into three families depending on the configuration of the coordinated elements:

- the Prague family – all coordinated elements are children of one of the separating conjunctions or punctuation marks;
- the Moscow family – the coordinated elements form a string of dependencies;
- the Stanford family – the other coordinated elements are dependents of the first or last coordinated element.

Based on this, transformations were created (see Figure 6) the results of which represent each of these families:

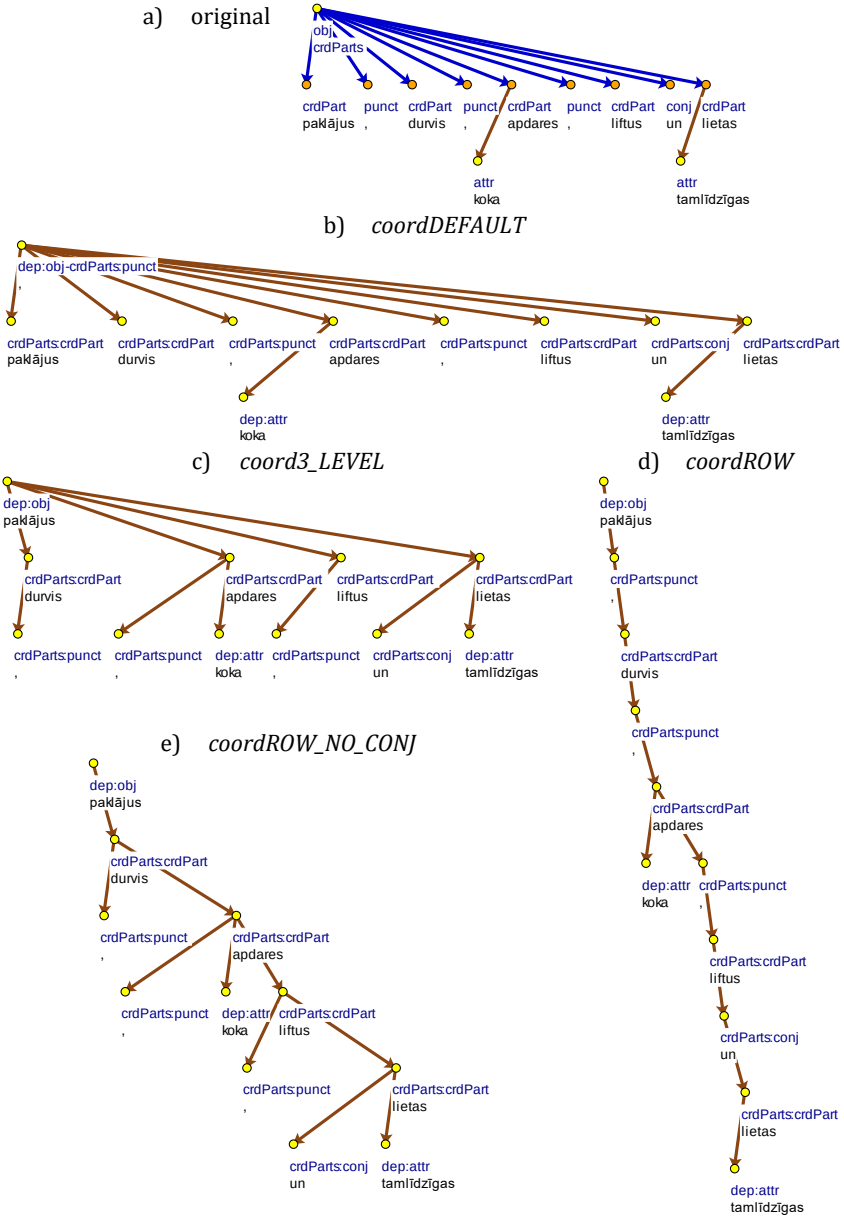


Figure 6: Examples of coordination construction transformations, legend: blue edges – components of coordination construction (phrases), brown edges – dependencies

- *coord3_LEVEL* – representative of the Stanford family: the linearly first element with the role *crdPart* is chosen as the root of the subtree corresponding to the phrase and other elements with the *crdPart* roles are linked as its children. The *conj* and *punct* elements are linked as children to the closest *crdPart* element, which is located after them in the sentence (see Figure 6c);
- *coordDEFAULT* – representative of the Prague family: a conjunction is chosen as the root of the subtree corresponding to the phrase, which is located between the linearly first element with the *crdPart* role and the second one. If there is no such conjunction, e.g. in the phrase *zēni, meitenes un suņi* ('boys, girls, and dogs'), the punctuation mark between the first two linear elements with the *crdPart* role is chosen. All other elements of the coordination construction are linked as children to the selected root of the subtree (see Figure 6b).
- *coordROW_NO_CONJ* – representative of the Moscow family: the linearly first element with the role *crdPart* is is chosen as the root of the subtree corresponding to the phrase. Each subsequent element with the *crdPart* role is linked as a child to the previous one. Elements with roles *conj* and *punct* are linked as children to the closest element with the *crdPart* role located after them in the sentence (see Figure 6e).
- *coordROW* – representative of the Moscow family: the linearly first element with the role *crdPart* is chosen as the root of the subtree corresponding to the phrase. Each subsequent element of the phrase is linked as a child to the previous one. If an element with a *conj* role is located before the first element with the *crdPart* role, it is linked to the first element with the *crdPart* role (see Figure 6d).

2.3.2.3.3 Transformations of Punctuation Mark Constructions

Each punctuation mark construction (PMC) contains the following: one base element (corpus role *basElemin*), which, together with the elements underneath it in the subtree, causes the use of the punctuation marks, and one or more punctuation marks (corpus role *punct*). This construction may also include conjunctions (corpus role *conj*), for example, in cases when a subordinate clause separated by punctuation marks starts with a conjunction or an arbitrary number of elements without a clearly defined syntactic role (corpus role *no*) e.g. addresses, insertions, particles. For example, in the sentence *Bet, Anna, aizver durvis!* ('But, Anna, close the door!') the PMC of the sentence consists of an exclamation mark, a conjunction *bet* ('but'), the base element *aizver* ('close'), and an element

with the *no* role, which further contains another PMC (address) with the basic element *Anna*.

Given that the most important elements of the construction are *basElem*, *punct*, and *conj* (if present), the following transformations were considered (see Figure 7):

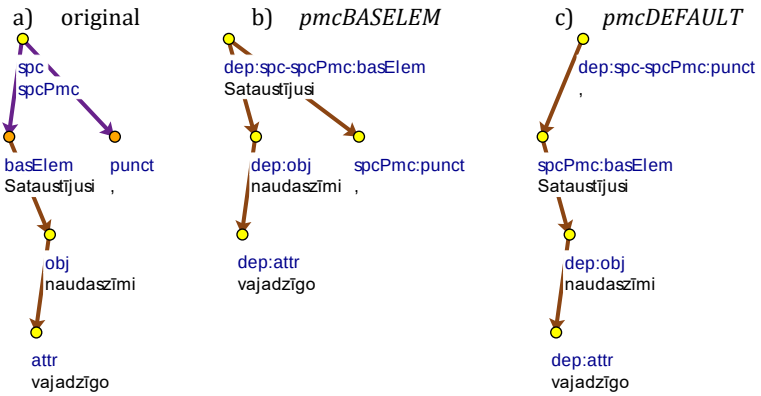


Figure 7: Examples of transformations of PMCs,
legend: purple edges – components of the PMC (phrases), brown edges – dependencies

- *pmcBASELEM* – an element with the role *basElem* is selected as the root of the subtree corresponding to the phrase, while other elements of the phrase are made dependents of the selected root element (see Figure 7b);
- *pmcDEFAULT* – the first linear element with the role *conj* (or, if there is none, the first element with the role *punct*) is selected as the root of the subtree corresponding to the phrase and the rest of the elements are made dependents of the selected root element (see Figure 7c).

2.3.2.3.4 Summary

Each phrase transformation is carried out independently from others, so it is possible to readily combine the considered transformation variants. Thus, it is possible to obtain 16 different transformations to convert the treebank data for the training of the dependency parser. By using these transformations, 16 treebank versions can be obtained, in which the same texts are annotated in different ways. However, in order to pinpoint the comparative advantages of the resulting treebank versions, practical experiments were necessary, since there were not enough

theoretical linguistic considerations that would give a clear answer regarding this.

2.3.2.4 Parser Training and Transformation Comparison

The first parser training experiments were carried out by manually comparing the parsing algorithms implemented in the *MaltParser* system (Pretkalniņa, Rituma, 2013). However, for further selection of the system parameters, it was chosen to use the *MaltOptimizer* system (Ballesteros, Nivre, 2012), which performs the selection of the most suitable parameters in a three-step process. In the first step, *MaltOptimizer* calculates the general parameters of the corpus, such as the number of tokens and the proportion of non-projective trees. In the second step, the parsing algorithm is selected. In the third step, the training features are selected according to the parsing algorithm – for certain tokens it is possible to add word form, lemma, morphological information and the (*FEATS)* column featured on the *CoNLL* tabular format, which allows to provide any additional information.

The experiments were conducted using the manually created morphological annotation already existing in the corpus, as well as the automatically tagged morphological annotation. In comparative experiments, emphasis has been placed on the use of automatically generated morphological annotation, as this option is better suited to the possible uses of the parser – manual morphological annotation takes time and human resources, so parsers are most often used for texts with automatically created morphological annotation. The use of automatically annotated morphology resulted in a slight drop in accuracy – around 1–2 percentage points (Pretkalniņa, Rituma, 2013).

For a better insight in the effect of the created transformations of syntactic annotations on parsing, two types of experiments were conducted. The extrinsic evaluation of parsers was based on the results generated by tools that employ parsers, whereas the intrinsic evaluation of parsers was performed by comparing parsing accuracy for fixed test datasets and individual structures within them.

2.3.2.4.1 Extrinsic Evaluation

Although a syntactic parser has the potential of vast applications in the data preparation and indexing for linguistic studies, more often they are integrated into various tool systems that use the results of syntactic analysis as input data for other tools. It is therefore essential not only to evaluate how syntactic representation affects the accuracy of parsing, but also examine its impact on the accuracy of the tools using the parsing-generated

results. For this aim, detailed experiments were conducted with the 16 transformations described and the three tools created for the Latvian language, which are applicable to various aspects of semantic analysis: the semantic role labeller, the coreference resolution tool and the named entity recognition tool.

Semantic role labelling (SRL) (e. g., Barzdins et al., 2014) is a semantic analysis task that works at the level of a sentence in the given Latvian language tool created at the time. This tool uses the *FrameNet*[25] (Ruppenhofer, 2010) method – it recognises a finite number of situations, i.e, semantic frames that consist of frame targets and frame elements. The employed Latvian language tool is designed for 26 different frames that are characteristic of news articles e.g., *birth*, *selling/buying*, *taking office*, *establishing a company*. The labelling is carried out in two stages: firstly, by locating the frame targets that identify the frame, and secondly, by pinpointing the instantiations of possible frame elements for the identified frames. As an example, the following sentence can be examined: "*Uzņēmums "Laima" iegādājies ražošanas tehniku 30 tūkst. eiro apjomā"* ('The company "Laima" has purchased production equipment worth 30 thousand euros'). If the semantic roles in the sentence should be labelled using the previously described SRL method, it can first be noted that the word *iegādājies* ('purchased') evokes the *selling/buying* frame, and that it can subsequently include the frame elements 'buyer' (the company "Laima") and 'purchase' (production equipment).

During the first stage of labelling, the SRL tool uses the dependency roles as features, but during the second stage both the roles and the structural features of the syntactic tree are used. Also, the structure of a dependency tree determines the boundaries of frame elements, if the element is instantiated by several words. In the previous example, the element *purchase* would correspond to *production equipment worth 30 thousand euros*; the labelling tool would mark *equipment* as a frame element, while the rest of the words, being the components of the frame element, would be identified as children or descendants of *equipment* in the dependency tree.

Coreference resolution (e. g., Znotins, Paikens, 2014) is a semantic analysis task that is carried out on the level of the entire text. The purpose of this task is to identify a type of sentence fragments that are called 'mentions', which refer to a single person or thing (referent). For example, in the text *Jānis ir centīgs jaunietis, viņš vienmēr izpilda mājasdarbus* ('Jānis is a diligent young man, he always does his homework') the process of coreference resolution would consist of identifying the fact that *Jānis* and

[25] *FrameNet* project website: https://framenet.icsi.berkeley.edu

viņš ('he') are different mentions of the same referent. In coreference resolution, the characteristics of the dependency structures are used as identifying features.

Named entity recognition (NER) (e. g., Paikens et al., 2012; Znotins, Paikens, 2014) is a semantic analysis task that is carried out on the levels of a sentence, as well as the entire text. The aim of this task is to identify place names, personal names, company names and other named entities within the scope of the system in question [26]. The process of named entity recognition for Latvian essentially uses templates and lists of names, but for the purposes of experiments the tool was supplemented by the use of syntactic features, namely, dependency roles and certain characteristics of dependency structures (for example, the ancestor closest to a certain node in the dependency tree, which is a noun).

The **results** of the experiments have been published in Pretkalniņa et al. (2014). Looking at the results of several semantic tools, it was concluded that the most suitable dependency representation may differ depending on the semantic task at hand.

- For NER, the use of syntactic features did not noticeably improve the accuracy of results, therefore, as far as tool pipelines are considered, it is possible to even use it before the parser.
- The best results in coreference resolution were achieved using parsers that represent coordinated parts of a sentence as *coordDEFAULT,* despite the fact that the syntactic parsing accuracy provided by these parsers themselves were noticeably worse.
- For the SRL, the use of syntactic information lead to a significant improvement in the second stage i.e., identifying frame elements – in this respect, the best results were achieved by a parser that represents coordinated parts of a sentence as *coordROW_NO_CONJ.*

Thus, in order to obtain optimal performance of semantic tools for the resources at the time, it was not enough to choose a parser with the highest accuracy of results, even if the accuracy rates differed by more than five percentage points (~10%). This confirms that the decision to use the LVTB hybrid annotation model in a treebank has been vastly successful; this allows for easy retrieval of different dependency representations, which in turn helps identifying the most appropriate dependency structure for each case of parser use.

At the same time, it should be noted that the results were obtained from a small corpus and there were large fluctuations in result accuracy

[26] The NER tool, which was developed for Latvian, does not consider dates and time identifiers as named entities, although sometimes such information is also included in the scope of the task.

during the training. Even though the results provided significant insights into the further development of the study at the time, it is not safe to draw conclusions from them regarding large corpora and other parsing methods.

2.3.2.4.2 Intrinsic Evaluation

The suitability of dependency representation for various tasks can be evaluated not only based on the accuracy of certain task performance, but also by how accurately the parser system learns one or other representation of a certain syntactic structure. Therefore, a second series of experiments was conducted in which the parser results were compared regarding structures with ambiguous representation, such as coordination constructions, complex predicates and PMCs. The results of these experiments were published (Pretkalniņa, Rituma, 2014).

A series of experiments were conducted to evaluate the parsers and the results were compared in three ways using LAS, UAS, and LA metrics:

1) the results of the parser as a whole;
2) the components of a certain phrase for the following groups of phrases (each group evaluated separately):
   - PMCs,
   - coordination constructions,
   - complex predicates;
3) phrase dependents for the following phrase groups (each group separately):
   - PMCs,
   - coordination constructions,
   - complex predicates.

To obtain the quantifiable tokens for the metrics of groups 2 and 3, the original (hybrid model) corpus annotation was used. The following components are considered as parts of a phrase:

a. a token that is marked as part of a phrase in the original annotation;
b. (if the phrase component itself is a phrase) a token which, after transformation into a dependency representation, becomes the root of the subtree representing the subphrase.

The following components are considered a phrase dependent:

a. A token that is marked as a phrase dependent in the original annotation;
b. (if the phrase dependent is a phrase) a token which, after transformation into a dependency representation, becomes the root of the subtree representing the phrase.

Summarising the accuracy of component recognition for the various dependency representations of phrase-style constructions, it was concluded

that the *xpredDEFAULT* and *pmcDEFAULT* parsers are superior, thus also confirming the tendencies observed in the overall results. When analysing the results of the recognition of coordination construction elements, it was observed that the results of *coordDEFAULT* parsers are clearly worse than others. This tendency was also reflected in the overall results. Component recognition and overall accuracy tests both show that the dependency structures of the Moscow family are superior – the best results were given by *coordROW* and *coordROW_NO_CONJ* parsers, but the results did not show that either one of both was definitely better than the other.

When analysing phrase-dependent recognition, the most difficult aspect was the small number of elements dependent on punctuation and coordination constructions within the corpus – 2.5% and 1% of tokens [27] respectively; therefore the accuracy was low for these dependents. Regarding the constructions of coordinated parts of a sentence, it was concluded that at the time the size of the corpus was not sufficient to fully learn to distinguish between the common dependents of coordination constructions and the dependents of one coordinated element, since LA had a very low (15–30%) recognition rate for all types of coordination constructions. Regarding the dependents of PMCs, one must keep in mind that in the case of *pmcDEFAULT,* the root of the corresponding dependency subtree is a conjunction or a punctuation mark, whereas in *pmcBASELEM* the fragment of the tree structure under question is similar to cases where the base element of the PMC is used with the same dependent, just outside of the PMC. It was therefore concluded that in the case of a sufficiently large corpus, the use of a *pmcDEFAULT* type of representation would in theory be more informative, whereas for a small corpus (such as the one available for the experiment) it is better to use *pmcBASELEM*, which is easier to learn for a parser.

On the other hand, when analysing complex predicate dependents with numerous examples in the corpus, recognition results are in favour of *xpredDEFAULT.* This, together with component recognition and general

[27] The LVTB data available at the time of the experiments were as follows: 15.3% of the tokens were components of coordination constructions, 31.9% of tokens were in PMCs and 7.1% were in complex predicates. The dependents of the mentioned phrases were relatively rare – there was 1% of coordination construction dependents in the whole corpus, as well as 7% of complex predicate dependents and 2.5% of the PMC dependents (in PMCs the dependent consists of elements that apply to the entire clause, and not to an individual part of the sentence; according to the theory of Latvian linguistics they are usually situants and determinants, as well as some types of subordinate clauses).

experiment results, clearly suggests that the use of *xpredDEFAULT* is the optimal solution.

Parser variants, the advantages of which are indicated by the analysis of individual linguistic phenomena, also count among the best in general tests. Compared to the general results of parsers obtained in the previous series of experiments (see Section 2.3.2.4.1 and (Pretkalniņa et al., 2014)), it can be observed that due to the small size of the corpus, the results have a rather high degree of instability. However, in general, these initial studies provided valuable insights into the construction of parsers and helped prepare for the subsequent studies described in the next chapter.

3 Latvian Universal Dependency Treebank

In 2014, using the *Universal Dependency Treebank* project (McDonald et al., 2013) and *the HamleDT* project (Rosa et al., 2014) as its basis, the Universal Dependencies, (UD) initiative was created. The project is coordinated by Joakim Nivre with the aim of offering a language-independent framework for creating treebanks[28]. The main goals of the UD initiative are as follows: high linguistic accuracy of annotation, cross-lingual consistency of annotations, quick process of annotation and parsing, easy access (intelligibility) for potential users of data outside the circle of linguistics, as well as usefulness for further use in natural language understanding (Nivre et. al., 2016). To achieve these goals, the UD initiative has developed guidelines for a dependency-based annotation model and provided a clear definition of the employed roles and morphological categories, as well as the possibilities of creating language-specific model extensions and guidelines. With the publication of a new version of the data every six months, the range of data created by the initiative is rapidly developing: the UD version 1.2 published in November 2015 includes 37 corpora for 33 languages, version 2.2 published in July 2018 includes 112 corpora for 71 languages, and version 2.11 published in November 2022 includes 243 corpora for 138 languages[29].

Participation in the UD initiative with a Latvian language corpus annotated according to the provided guidelines expands various possibilities for research and applications – the language-independent model allows comparative studies in which Latvian language data is used together with other language data, as well as the use of various tools developed by other researchers designed for this grammar model and data format. A treebank suitable for parser training is an essential goal of this work, which also aligns with the intention of UD to ensure a seamless parsing process. In addition, the striving of the UD towards both linguistic accuracy and usability in natural language understanding tasks aligns with conclusions from Section 2.3.2.4, which describes the necessity to select a syntactic representation that not only serves parser training purposes, but also ensures that the results generated by the parser would be informative enough for further use. Therefore, it was decided to explore the possibilities of creating custom transformation that would allow the data of the Latvian Treebank (LVTB) to be published within the UD framework.

[28] UD Initiative website: https://universaldependencies.org/
[29] UDv2.11 is available in the LINDAT repository
http://hdl.handle.net/11234/1-4923

3.1 The Creation of the Treebank

The first time that the Latvian UD Treebank (UDLV-LVTB) was published alongside other UD corpora was in version 1.3 in May 2016 (Pretkalniņa et al., 2016). The initial version of the corpus consisted of approximately a thousand sentences (the LVTB newswire portion), but in version 2.0 contained a transformed version of the entire LVTB material annotated at the time. In version 2.11 released in November 2022, the size of the UDLV-LVTB corpus has reached 285 thousand tokens and 16.9 thousand sentences.

UDLV-LVTB contains the following annotations (the names of the corresponding fields in the *CoNLL-U* tabular data format used by UD are given in parentheses):

- the text is divided into tokens and sentences;
- a lemma is specified for each token (*LEMMA* is a preferred but optional field according to UD guidelines);
- for each token, morphological information is indicated: a morphological tag from the original annotation (XPOS is a preferred but optional field), a part-of-speech identifier according to the UD specification (UPOS is a mandatory field), and the expansion of morphological features according to the UD specification (*FEATS* is a preferred but optional field);
- the structure of dependencies is indicated within the scope of a sentence, that is, each token has its parent in the dependency tree (*HEAD* is a mandatory field) and the dependency type i.e. syntactic role (*DEPREL* is a mandatory field) shown;
- starting with version 2.1, within the scope of a sentence, an expanded dependency structure containing additional links useful for further use in language understanding tasks (*DEPS* is preferred but optional field) is also included.

The annotation is obtained by processing LVTB data with a LVTB2UD transformation specifically created for UD purposes. This makes it easy to replenish the UD treebank when LVTB is updated with new data. The source code for the transformation is available online[30]. While moving from version 1.4 to 2.0, significant changes and clarifications are made to the UD guidelines, and the transformation is updated accordingly. Further on in this text, unless indicated otherwise, the operation of the transformer according to the guidelines of the latest version (2.11) is described.

[30] *GitHub* repository *CorporaTools*, LVTB2UD folder: https://github.com/LUMII-AILab/CorporaTools/tree/master/LVTB2UD

3.1.1 The Transformation of UDLV-LVTB Creation

UDLV-LVTB is a rule-based transformation grounded in the experience described in Section 2.3.2.3, but unlike the initial transformations, instead of being parametric, this one is designed as a complex set of rules aimed at describing with maximum accuracy how each of the LVTB constructions (phrases of different types, dependencies) is transformed into a UD construction, including both the relabelling the roles accordingly and structural changes.

LVTB2UD processes input data sentence by sentence, performing the processing steps for each sentence as described below. The processing of each sentence is algorithmically independent of the processing of previous and subsequent sentences.

3.1.1.1 Tokenization

The first step in the processing of each sentence is to deduce whether the LVTB tokenization complies with the UD guidelines, and if not, to create an altered tokenization. Such differences are rare and fall into two categories: corrections of editorial errors and 'words' with spaces in them.

In line with the practice of PDT, editorial errors, such as redundant diacritical marks (e.g. *gribās*) and words that are incorrectly spelled together or separately (e.g. *jā dara*, *kautkas*) in LVTB, are corrected at the same PML level (see Section 2.2.1) as morphological annotation (in the same file), while syntactic annotation is created at the next level (in another file) for the already corrected text. The UD methodology, on the other hand, requires using a text with all editorial errors and introduces the dependency role *goeswith* to connect unnecessarily separated words (e.g. *jā dara*). Since LVTB contains information about both the original text and the type of editorial correction, in this step the original text is reconstructed and, if necessary, *goeswith* links are also added. For errors such as omitted commas, for which the UD does not specify the manner of indication, codified comments are placed in the appropriate token's *MISC* field, which is intended to provide unspecified information. However, this representation may change in the future if the UD specifies a uniform way of indicating such information.

In the initial LVTB and UD versions, there was a large rate of incompatibility regarding the use of 'words' with spaces – while UD did not allow such cases at all, LVTB allowed them in numerous cases, such as the representation of figures written in numbers (*10 000*), coordination conjunctions and particles (*lai gan*) and abbreviations (*u. c.*). However, in version 2.0, the LVTB and UD approaches became much closer: The UD

supplemented the guidelines with the option of using 'words' with spaces with the prerequisite that they must listed in the language-specific documentation using regular expressions, while LVTB, in line with UD, deprecated annotating complex conjunctions and particles as unified tokens. As a result, both LVTB and UDLV-LVTB use only certain tokens containing spaces as enumerated list of abbreviations (*P. S.*, *N. B.* and such abbreviations as *u.c.* ('etc.'), *v.tml.* ('or similar') if they are written down with a space in them) and figures written down using numbers; in this aspect, specific conversion is no longer necessary.

3.1.1.2 Initial Morphological Information

The next step is to fill in the morphology fields *LEMMA*,, *XPOS, UPOS*, and FEATS. The fields *XPOS* and *LEMMA* are filled in with the morphology tags and lemmas given by LVTB. The *UPOS* and *FEATS* fields are filled in according to the rules developed by linguists that use lemmas and tags as input data. In some cases, this information is not enough to determine *UPOS*:

- UD needs a division of LVTB's pronouns into determiners (*DET*, a pronoun that replaces an adjective, *tā māja* ('that house')) and pronouns (*PRON,* a pronoun that replaces a noun, *tas ir viņš* ('it is him')), while LVTB does not provide such a division, since in Latvian it is often not expressed by morphological markers.
- UD requires a separation of adverbs that introduce a subordinate clause (*SCONJ*) from other adverbs (*ADV),* while LVTB does not require such a division as it is not expressed in Latvian by morphological markers.
- UD requires that for insertions from other languages, an appropriate part-of-speech should be indicated so it is known which part of speech the insertion replaces in the given sentence, whereas LVTB annotates such cases altogether as nonmorphological elements.

The contents of the *FEATS* column are essentially determined by the features included in the morphological tag, but in some cases features are also assigned according to the lemma enumeration, e.g. for most adjectives, the feature *Poss* (*possesive*) is not indicated, but expressed through such words as *manējais* ('mine'), *tavējais* ('yours'). The UD also offers a number of features and feature values that are not useful for the Latvian language because they do not appear as morphological categories, such as *Animacy* or *Case=Erg*. In order to better infer certain features, the annotation used by LVTB was also improved, e.g. participle annotations were supplemented with indicators of degree and negation. Thus, the participation in the UD initiative also influences the development of the hybrid model.

This step of morphological annotation, along with the tokenization described in the previous section, can also be performed if only morphological, but not syntactic, annotation is available for the text. Thus, these tools can also be used to make the results of the morphological tagger (Paikens et al., 2013) available for international projects.

3.1.1.3 Syntactic Structures of UD

The most important step of the transformation is the deducing of the syntactic structures of UD from the available LVTB annotations.

A comparison of the basic UD dependency tree and the LVTB annotation found that (1) the dependency relations used by LVTB correspond to the UD dependency links, (2) phrase-like constructions in LVTB mostly correspond to interconnected tree-like fragments in the UD tree and (3) the dependents of phrase-like constructions in the LVTB tree correspond to the root dependents of the relevant tree-like fragments in the UD tree. Therefore, the transformation can be constructed as a recursive algorithm that transforms each phrase-style construction or dependency by using the information available in its immediate neighbourhood. The developed transformation is based on tree traversal, by recursively processing first the children of each node (including both dependents and, if it is the node of a phrase, also the components of the phrase) and then the node itself (postorder traversal).

When processing each node, its dependents are assigned UD dependency roles, based on both the LVTB roles and the morphological characteristics of the dependent and the parent. When processing the nodes of phrase-like constructions, the corresponding fragment of the dependency tree is created and each component of the structure is assigned a UD dependency role, based on its LVTB role and morphological annotation, as well as the overall annotation of the phrase-like construction: type, role in the sentence and morphosyntactic tag of the phrase.

Empty nodes that denote word omission i.e. ellipsis nodes (or reduction in older papers), are processed as follows:

1. ellipsis nodes without dependent nodes are not converted to UD;
2. (for ellipsis nodes with dependents) based on their potential UD roles, one of the dependents is selected as a replacement for the ellipted element; this node is then subordinated to the parent of the ellipted element and the other dependents of the ellipted element are subordinated to it.

Over the course of this thesis, a detailed many-to-many mapping between the roles used by LVTB and UD has been developed[31], but it has also been concluded that in some cases the LVTB annotations do not allow to accurately determine the UD role (Pretkalniņa et al., 2016).

Version 2.0 of the UD guidelines describes optional supplementary annotations that would further facilitate the use of UD data in various software applications – the so-called enhanced dependencies. The guidelines propose five types of additional information for annotations, each of which is optional:

1. null nodes representing elided predicates;
2. edges that link each of the coordinated parts of a sentence to their common parent and common dependents (in basic dependency trees, only the first of a series of coordinated parts has such a direct link);
3. edges indicating the controlled/raised grammatical subjects of the sentence e.g. in the sentence *viņš grib ēst* ('he wants to eat') the subject *viņš* ('he') is connected not only to *grib* ('wants') (edge in the basic dependency tree), but also to *ēst* ('eat'); in the sentence *pavelkot virvi, viņa atsēja mezglu* ('by pulling the rope she untied the knot') the subject *viņa* ('she') is also connected to *pavelkot* ('pulling');
4. additional edges that link relative pronouns introducing a subordinate clause to a part of the main clause to which the pronoun refers to;
5. Additional information that can be included in the role:
    a) grammatical case or preposition can be added to the role of a non-core nominal part of a sentence, such as *nmod:loc* for a noun in locative;
    b) The adverb introducing a subordinate clause can be added to the role of a subordinate clause e.g., *advcl:kad* for an adverbial clause, which is introduced by the adverb *kad* ('when').

After analysing the guidelines and the LVTB annotations, it was concluded that it is impossible to comply with Point 4 without additional annotating of coreferences, since LVTB annotation does not distinguish whether pronouns and adverbs introduce a subordinate clause or not. For the same reason, the conditions of Point 5 can only be partially fulfilled – information is given only about grammatical cases and prepositions (a), but not about the adjectives introducing subordinate clauses (b).

[31] The latest version is published for each release on the LVTB website http://sintakse.korpuss.lv/. At the time of writing the work it was the following one: http://sintakse.korpuss.lv/docs/v2-11/LV2UD_mapping.pdf

Implementation of Point 1 of the enhanced dependencies guidelines is made possible by the ellipsis annotation strategy used by LVTB, which involves the formation of empty nodes in place of omitted elements, while the information required for Point 2 is provided by the fact that LVTB depicts coordinated parts of a sentence with a phrase-like construction, thus distinguishing the dependents of the individual coordinated part of a sentence from shared dependents.

The phrase-style construction *xPred* used for LVTB complex predicates allows to form a substantial portion of the links required for Point 3, as it groups together longer sequences of predicates, such as [*viņš*] *gribēja gulēt* ('[he] wanted to sleep'). However, in Latvian, participle constructions can also have a common subject with the predicate of the corresponding clause e.g., in sentences *māte sakās par dēlu neko nezinām* ('the mother says she doesn't know anything about the son') (Kalnača, Lokmane, 2018) and *žāvādamies viņš piecēlās* ('yawning, he stood up'). The available LVTB annotations do not allow to distinguish between such situations and sentences in which the participle has another, omitted, subject, as in the sentence *redzēju līstam* ('I saw [it] raining') and *aizķerot trolejbusa vadus, cirka ēka var sabrukt* ('if [something] gets caught on trolleybus wires, the circus building may collapse'). In ambiguous situations, the edges of the enhanced dependency graph are not added so as not to create erroneous annotations.

Initially, the transformation was designed only to create a basic dependency tree. After the publishing of the UD v2 specification, it was supplemented with a functionality that converts the necessary LVTB ellipsis nodes into UD enhanced dependency ellipsis nodes so that they, together with the basic dependency tree, form the 'backbone' of the enhanced dependency graph, and then supplement the it with the necessary additional edges (Pretkalniņa et al., 2018) using the information already present in the 'backbone'. Such a solution has proven to be suboptimal because it means that a less informative structure (a UD base tree) is created from a more informative data structure (the original LVTB tree), which must then be supplemented with information to obtain an enhanced dependency graph. Therefore, during the project "Full Stack of Language Resources for Natural Language Understanding and Generation in Latvian" (*FullStack-LV*) (see Section 3.2.1) prior to the UD version v2.4, the transformer was reworked (Gruzitis et al., 2018) in such a way that every step of the LVTB

tree traversal creates all the basic and enhanced dependency links to the nodes that have been processed so far[32].

After the improvements, the conversion **algorithm** works in the following steps.

1. Pre-processing:
   1.1. By traversing the tree bottom-up, a one-to-many mapping is constructed, in which node identifiers maps to lists of the coordinated parts of a sentence and coordinated clauses that the specific nodes consist of;
   1.2. By traversing a tree top-down, a one-to-many mapping is constructed in which the node identifiers of the grammatical subjects/subject clauses refer to lists of the nodes that the corresponding node relates to as the grammatical subject or subject clause. First, each of these lists are compiled of nodes according to the LVTB direct links, after which they are supplemented with parts of complex predicates, if the corresponding node is associated with a complex (i.e., multi-word) predicate. Nodes in each resulting list are sorted based their depth in the LVTB syntax tree, i.e., how long its shortest path to the root from the given node is.
2. Tree traversal and creation of a UD structure by processing each node:
   2.1. Before processing a node, all its dependents are recursively processed, and if it is a node of a phrase-like construction, its components are processed as well;
   2.2. It is determined which token represents a given node both in the basic dependency tree and the enhanced graph:
      2.2.1. if it is a ellipsis node, an "ellipsis transformation" is carried out – the dependents are analysed and the replacement for the parent in the basic dependency tree is defined, as well as an empty node for the enhanced dependency graph is created;
      2.2.2. if it is the node of a phrase-style construction, a "phrase transformation" is carried out – the components are analysed, the edges of the UD graph between them are created, and it is determined which of the components becomes the root of the UD tree fragment representing the

[32] *GitHub* repository *CorporaTools*, LVTB2UD folder, *NewSyntaxTranslator* class: https://github.com/LUMII-AILab/CorporaTools/blob/9e0786fe7713528d47e0aaeb7123e1f4fba2591a/LVTB2UD/src/lv/ailab/lvtb/universalizer/transformator/syntax/NewSyntaxTransformator.java

phrase; in the base tree, the phrase will be represented by a token corresponding to that component (determined recursively in previous steps), whereas, in the enhanced graph, it will be that same token or empty node if it has turned out to be the a ellipsis node;

2.2.3. if it is simply a node of a token, it is represented by the corresponding token both in the UD base tree and the enhanced graph.

2.3. Dependencies are created in both the base tree and the enhanced dependency graph between the dependent elements of the processed node (for the UD nodes representing them) and the UD nodes representing this node. When processing dependents for whom additional parents are specified in the mapping created in paragraph 1.2, the edges of the enhanced dependency graph are also drawn to them.

2.4. Simultaneously, each time an edge is created, the mapping of Point 1.1 is consulted and all the necessary links of the enhanced dependency graph are established between the coordinated elements of the parent and the child of the new dependency. The roles of the links are determined by the nodes' neighbourhood in the tree, the morphological characteristics of the parent and child, and the processing context (whether it is the processing of a phrase, an ellipsis, or a dependency).

3. At the end of the processing of the tree, a special role *root* is assigned to the root node of the created tree.

3.1.1.4 Post-Processing of Morphology

After UD syntactic structure has been obtained, it is possible to refine the morphological annotation – mainly, to reduce the number of incorrectly assigned PRON parts-of-speech by re-tagging pronouns that have fallen into the determinant role (*det)* in the UD syntactic structure as DET.

3.1.1.5 Comparison with the Transformations of Previous Experiments

Since the UD structure of basic dependencies is also obtained by transforming LVTB phrase-like structures into rooted subtrees, it is possible to compare the structural aspect of the transformation elaborated in this section with the transformations described in Section 2.3.2.3:

- PMCs are transformed in the same way as in the *pmcBASELEM* transformation (?);

- coordinated parts of a sentence are transformed according to *coord3_LEVEL* (2.3.2.3.2) strategy, adding a nuance in cases where a longer sequence of coordinated parts of a sentence is separated by a semicolon – the UD ensures that the sequences of coordinated clauses on both sides of the semicolon are processed as separate sequences, which are then subordinated to each other;
- complex predicates with the auxiliary verb *būt* ('to be') in copula constructions or perfect tense constructions, as well as with auxiliary verbs *tikt* ('get') and *tapt* ('become') in perfect tense constructions are transformed in the same way as in the *xpredBASELEM* strategy, while the rest are transformed according to the *xpredDEFAULT* strategy;
- phrase-like constructions consisting of one functional word and one full-meaning word, such as the prepositional construction *xPrep – uz galda* ('on the table'), *aiz stūra* ('around the corner') – are transformed conversely to the strategies described in Section 2.3.2.3, where the functional word becomes the root of the phrase, and in the UD it is the content word that becomes the root.

Compared to observations in Section 2.3.2.4, it can be noted that most of the elements of the UD transformation are easier for parsers to learn, so converting the data into UD format helps to achieve the goal of this doctoral thesis – to obtain a higher-precision parser for the Latvian language.

3.1.2 UDLV-LVTB Qualitative Assessment

In the article Pretkalniņa et al. (2018), a manual evaluation of the sample sentences has been carried out in order to assess the quality of the transformation. Due to human resource limitations, the evaluation is relatively small and therefore not considered as highly representative; however, in spite of this, it provides valuable insights into the quality of the transformation results. An assessment was carried out for 60 sentences (approximately 800 tokens, sentences are selected to proportionally represent the text genres in the corpus) in order to determine, whether the UD annotation errors found in the sentences have occurred due to faults in the original annotation, transformational imperfections, or due to the fact that LVTB does not contain the information to identify the correct links and their roles.

In the basic dependency trees, 19 incorrect links / link roles have been identified, i.e. the transformation LAS is 97.6%, which is considered a very good result. Of all the problems encountered, only one occurred due to a lack of necessary information in LVTB, while six are due to errors in the

original data. The other 12 can be considered as repairable transformation errors.

Within the same data, the quality of enhanced dependency graphs has also been assessed, however, due to the optional nature of enhanced dependencies, groups of enhanced dependency links that are not annotated at all, i.e., the enhanced dependencies specification groups described in Section 3.1.1.3, 4 and 5(b), are not considered to be errors. Three errors were identified due to mistakes in the original data, eight were due to enhanced dependency links with incorrect roles (all accompanied by the wrong grammatical case or preposition – enhanced dependencies specification group 5(a)) and 15 were due to missing links of coordinated parts of a sentence or subjects (enhanced dependencies specification groups 2 and 3 respectively). Although the test data did not show any cases when errors would have occurred due to a lack of necessary information in LVTB, it should not be assumed that there are no such cases at all, rather that they are relatively rare.

Even though the assessment was carried out for a small volume of data, it promotes a hopeful view of transformation accuracy. Furthermore, the conversion of the transformer described in Section 3.1.1.3 before UD v2.4 is directly aimed at improving major flaws – it significantly improves the mechanism for assigning enhanced dependency links associated with coordinated parts of a sentence and grammatical subjects, thus reducing the amount of potential errors.

3.2 The Importance and Impact of the Treebank

This section describes the further use of UDLV-LVTB both in research projects and for the construction of parsers for Latvian.

3.2.1 Treebank Data as a Basis for Future Research

In 2017–2019, in cooperation with the news agency LETA, IMCS UL carried out the project "Full Stack of Language Resources for Natural Language Understanding and Generation in Latvian"[33] (*FullStack-LV*), in which UDLV-LVTB plays an important role. During the project, a balanced text corpus of 10 thousand sentences was created, which has been annotated both syntactically and semantically on several levels using globally approved syntactic and semantic representations adapted to the Latvian language as well. In this corpus, the semantics of a sentence (see

[33] European Regional Development Fund (ERDF), industry-driven research 1.1.1.1/16/A/219

Figure 1) are represented using the *FrameNet* (Ruppenhofer, 2010) and *PropBank* (Bonial et al., 2014) models, whereas the semantics of text are depicted by Abstract Meaning Representation (AMR) (Banarescu et al., 2013) model. In addition, the corpus contains annotations of named entities and coreferences.

The *FrameNet* and *PropBank* levels are annotated in the treebank that has been annotated according to the UD approach. The UD treebank is automatically derived from LVTB, which is manually syntactically annotated according to the grammar model described in Section 2.1 (see Figure 8). Both treebank versions are published simultaneously. Thus, LVTB serves as the foundation for all subsequent levels of annotation in the *FullStack-LV* language resource. As part of the project, the LVTB2UD transformation has been significantly expanded and improved, and the size of the LVTB itself was increased to 13.6 thousand sentences (Gruzitis et al., 2018). One of the most significant achievements of the project is the integrated multi-purpose toolkit for various Latvian natural language understanding (NLU) tasks in *NLP-PIPE,*[34], which integrates into a single, configurable processing pipeline all necessary NLU components that have been trained using *FullStack-LV* data (Znotiņš, Cīrule, 2018; Gruzitis, Znotins, 2018; Paikens, 2017).

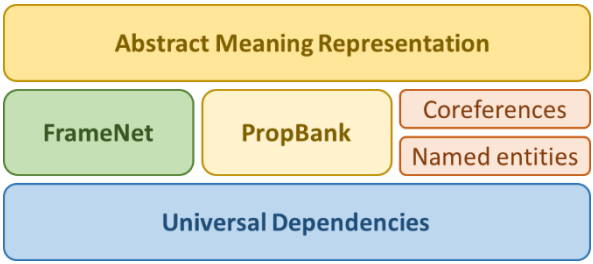


Figure 8: *FullStack-LV* multilayer text corpus: the representation layers at the bottom are used as the foundation for creating layers higher up (for more context, see Figure 1)

The improvement of LVTB and UDLV-LVTB corpora and their representations is being continued in the 2020–2024 State Research

[34] Source code is available in the *GitHub* repository https://github.com/LUMII-AILab/nlp-pipe, download is in the *Clarin.lv* repository http://hdl.handle.net/20.500.12574/4, demo available at https://nlp.ailab.lv

Programmes "Digital Resources for Humanities" [35] and "Letonika – Fostering a Latvian and European Society"[36] in accordance with the most recent UD guidelines and the latest LVTB-based research in Latvian grammar. The size of LVTB and UDLV-LVTB corpora has been increased to 17 thousand sentences, which also confirms the importance of the resource.

3.2.2 The Development of Parsers for Latvian

A historically significant point in the development of parsers for Latvian occurred in 2016, when Google published the *SyntaxNet* library and parser models for 40 languages, which were trained using UD v1.3. The library and modules were freely available[37], and the list of the 40 languages also included Latvian – this was the first time that an internationally recognised group of researchers from abroad had built a parser for the Latvian language, using data prepared in Latvia as the basis for it. The Latvian parsing module was trained using a corpus consisting of 3985 tokens[38] and yielded 58.92% UAS, 51.47% LAS.[39] For English, *SyntaxNet* results set a new record (Andor et al., *2016).*

Already in 2017, the parser building event *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (Zeman et al., 2017) took place; it used UD v2.0 as data for 45 languages, including Latvian.

---

[35] Project "Digital Resources for Humanities: Integration and Development" within the framework of the State Research Programme "Digital Resources for Humanities", project No: VPP-IZM-DH-2020/1-0001

[36] Project "Research on Modern Latvian Language and Development of Language Technologies" within the framework of the State Research Programme "Letonika – Fostering a Latvian and European Society",
project No: VPP-LETONIKA-2021/1-0006

[37] Trained models can be found in the *GitHub* repository of *tensorflow models* history: https://github.com/tensorflow/models/tree/a5d45f2ed20effaabc213a2eb9def291354af1ec/syntaxnet

[38] UD guidelines https://universaldependencies.org/release_checklist.html#data-split require for each treebank published to provide a canonical split into *train*/*dev*/*test* datasets and recommends to avoid moving sentences from one dataset to another between the releases. UDLV-LVTB follows this recommendation by proportionally increasing all three datasets, when a bigger version of the treebank is released.

[39] An overview of the results can be found in the *GitHub* repository of *tensorflow models* history:
https://github.com/tensorflow/models/blob/a5d45f2ed20effaabc213a2eb9def291354af1ec/syntaxnet/universal.md

The best results[40] for the Latvian language were given by the winning parser *Stanford* – it reached 79.26% UAS and 74.01% LAS (Dozat et al., 2017). Good results were also shown by the second best parser *C2L2*, which reached 77.43% UAS and 71.35% UAS (Shi et al., 2017). In the same year, the Latvian UD Treebank, along with other corpora, was used in a study (Nivre, Fang, 2017) that argues that LAS metrics mostly produce better results for parsers of analytical languages, such as English, and offers alternative metrics, thereby easing the decades-long dominance of the English language in embedded tool and metric assumptions in the field of research.

In 2018, the shared task was conducted again, using UD v2.2 data from 82 corpora for 57 languages. The size of the treebank available for the Latvian language at that time was already 81 thousand tokens (Zeman et al., 2018). The best results[41] for the Latvian language were given by parser *HIT-SCIR,*, reaching 87.76% UAS and 83.97% LAS (Che et al., 2018). The second-best result was given by the parser *Stanford* – 85.97% UAS and 81.85% LAS (Qi et al., 2018). In this shared task, the results were also compared using a new metric MLAS, which is similar to LAS, but also takes into account the accuracy of morphological annotation. Using this metric, the best results for the Latvian language were given by parser *Stanford* – 67.89%.

In both LAS and MLAS metrics, the results achieved for Latvian not only significantly exceeded the average results of the best parsers by language, i.e. 75.84% LAS and 61.25% MLAS, but also exceeded the average score in the so-called large corpus group, which was 84.37% LAS, 72.67% MLAS. For this task, the large corpus group consisted of 61 corpora; their individual volume had to be at least 25 thousand tokens. Of the larger UD treebanks mentioned in Section 2.2.2, the participants of the task included Czech *UD_Czech-PDT* and Russian *UD_Russian-SynTagRus* (at the time it was smaller – about 1 million tokens); the parsers trained with these corpora achieved some of the highest results in the task, exceeding 90% LAS and 85% MLAS. In general, the results of this Shared Task confirms that during it, thanks to a high-quality corpus, high-performance parsers have been created for the Latvian language.

In 2020, the *IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies* (Bouma et al., 2020) took place, and Latvian was among the 17 languages the data of which was offered to the participants.

[40] All results are available on the shared task website https://universaldependencies.org/conll17/results.html
[41] All results are available on the shared task website https://universaldependencies.org/conll18/results.html

For Latvian, the best parser achieved 85% ELAS accuracy (ELAS[42] – a metric derived from LAS and adapted specifically for evaluating enhanced dependency graphs). In 2021, the shared task was held again (Bouma et al., 2021), and the results reached up to 90.25% ELAS and 91.25% LAS accuracy[43] for Latvian, exceeding the average of 89.24% ELAS and 89.81% LAS. In these two tasks, parser accuracy was measured in enhanced dependency graphs, so even though the figures given here are not exactly comparable to the results of *CoNLL 2017-2018,* the graphs contain more information than basic dependency trees, therefore this analysis task was even more difficult. Thus, the high scores (for comparison, back in 2016, Andor et al. reports on *SyntaxNet* that the accuracy for English is above 90% for basic LAS dependencies, while Czech is just below this indicator) fully confirm the second hypothesis of the thesis, namely, that a medium-sized corpus (10-20 thousand sentences) is sufficient to create high quality, state-of-the-art parsers.

The corpus also simultaneously supports research on parsers and related technologies in Latvia. In 2016, a study was published that used the word embeddings approach, which was quite innovative and highly topical at the time. In the study, the parser trained with the UDLV-LVTB corpus reached 74.9% UAS (Znotiņš, 2016). In 2018, within the *FullStack-LV* project, the NLP-PIPE tool yielded the following results: 81.2% UAS and 76.8% LAS (Znotiņš, Cīrule, 2018*).* In 2020, the LVBERT tool achieved 89.9% LAS (Znotiņš, Bārzdiņš, 2020), and in 2022 (VPP, 2022) – 90.79% LAS. When analysing parser results depending on the size of training data (see Table 1), there is a significant increase in accuracy when comparing parsers using one or two-fifths of the data. When comparing parsers using two, three, or four-fifths of the corpus, the increase is about half a percentage point, while adding the last fifth of the corpus results in an insignificant improvement of accuracy.

The results lead to the conclusion that the given size of the treebank is enough to fully use the currently available possibilities of parser building technologies and to obtain high-quality parsers for the Latvian language. This means that the created corpus is optimal for performing the tasks defined in the thesis.

[42] An overview of the results of the task is available on the shared task website https://universaldependencies.org/iwpt20/Results.html

[43] An overview of the results of the task is available on the shared task website https://universaldependencies.org/iwpt21/results.html

Table 1: LVBERT parser accuracy increase depending on the size of training data (VPP,2022)

Part of training data used (*train*)	LAS % in the parameter calibration dataset (*dev*)	LAS % in assessment dataset (*test*)
20%	84.71	85.08
40%	88.51	89.28
60%	89.54	89.94
80%	90.13	90.56
100%	90.31	90.79

Conclusions

The goal of the doctoral thesis – to create and evaluate a large machine-readable treebank – has been achieved. The proposed hypotheses regarding the advantages of the hybrid annotation model and the suitability of a high-quality mid-sized corpus for high-precision (~90%) parser training have been confirmed in practice.

The thesis has provided the following conclusions:

- The choice to create and develop the hybrid grammar model described in the thesis has proven to be successful, as this model allows to represent the syntactic phenomena of the Latvian language by maintaining important nuances that are not always possible to accurately convey through models of pure dependency or phrase structure grammar.
    - In addition, corpus annotation in a rich hybrid format allows creating custom transformations for widely used but arguably simpler (in terms of information included) formats (e.g., UD), as well as conducting research on the most suitable dependency representations for various grammatical phenomena.
    - The dependency representation can affect both the accuracy of parsers and the text analysis tools that employ parsers. The observed effects are ambiguous – different tools prefer different dependency representations.
    - Unlike the simple and computationally efficient dependency models, the development of hybrid parser algorithms and machine learning models has not been a global research focus for long, it used to be more of a niche interest for researchers of morphologically rich languages. However, the issue of hybrid parsers has resurfaced lately more widely and recent experiments are showing promising results (Nivre et al., 2022).
- The UD grammar model has proven to be successful for further research and training of parsers, not only because of its international nature, but also because, based on the parametric analysis carried out in the thesis, in many instances it contains techniques for representing syntactic constructions, for which parser training systems also give better results.
    - The hybrid model used in the Latvian Treebank has enough elements in common with the UD model to make it possible to create a high-precision transformation in at least one direction. The UD model is arguably simpler, so a two-way transformation was not constructed.

- The hybrid model and the UD model have differing criteria for the selection and division of roles – the distinction between the syntactic roles used in the hybrid model is based more on the semantic structure of the sentence, and roles are, for example, an attribute (*attr*), adverbial modifier (*adv*). On the other hand, UD roles are selected mainly through morphosyntactic criteria, for example, in many cases the roles are grouped by part of speech – nominal modifier (*nmod*), adjectival modifier (*amod*), adverbial modifier (*advmod*), etc.
- In addition, a rule-based transformation into another (in this case, UD) annotation model facilitates detecting annotation errors and inconsistencies in the original corpus and thus, through feedback, improve the quality of the corpus.

- It has been concluded that the achieved size of the corpus (17 thousand sentences) is sufficient to create a parser of world-class accuracy – the best parser for the Latvian language in the *IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies* achieved the accuracy of 91.25 (LAS), which is significantly higher than the average result for all languages and is comparable to an average result in the large corpus group.
  - Due to the size, quality and compatibility of the corpus, various research teams have successfully used UDLV-LVTB data in four *CoNLL* and *IWPT* shared tasks in 2017, 2018, 2020 and 2021 for the development and evaluation of high-accuracy parsers. The corpus data has also been successfully used in parser studies in Latvia.
  - By analysing parser results depending on the amount of training data, it can be seen that the available size of the corpus is sufficient to fully utilise the currently available potential of parser building technologies and to obtain high-quality parsers for the Latvian language.
  - The convincing results of machine learning of the Latvian UD parsers confirm the quality and uniformity of the corpus annotation.
- Participation in the UD initiative has contributed to the international recognition of the Latvian language and other inflective language resources and promotes the creation of more suitable tools for inflective languages in the field of research, the historical origins of which are mainly found in the work with analytical languages.

Izmantotā literatūra

Andor, D., Alberti, Ch., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins., M. (2016). *Globally Normalized Transition-Based Neural Networks.* Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Vol 1.,Berlin, pp. 2442–2452.

Bagga, A., Baldwin, B. (1998). *Algorithms for scoring coreference chains.* Proceedings of LREC Workshop on Linguistic Coreference (LREC 1998), Granada, pp. 563–566.

Ballesteros, M., Nivre, J. (2012). *MaltOptimizer: An Optimization Tool for MaltParser.* Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, pp. 58–62.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N. (2013). *Abstract Meaning Representation for Sembanking*. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Sofia, pp. 178–186.

Barzdins, G., Gosko, D., Rituma, R., Paikens, P. (2014). *Using C5.0 and Exhaustive Search for Boosting Frame-Semantic Parsing Accuracy.* Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014), Reykjavik, pp. 4476–4482.

Bārzdiņš, G., Grūzītis, N., Nešpore, G., Saulīte, B. (2007). *Dependency-Based Hybrid Model of Syntactic Analysis for the Languages with a Rather Free Word Order.* Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007), Tartu, pp. 13–20.

Bielinskienė, A., Boizou, L., Kovalevskaitė, J., Rimkutė, E. (2016) *Lithuanian Dependency Treebank ALKSNIS*. Proceedings of the 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2016), Frontiers in Artificial Intelligence and Applications, Vol. 289, IOS Press, pp. 107–114.

Bohnet, B., Nivre, J. (2012) *A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing.* Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012), Jeju Island, pp. 1455–1465.

Bonial, C., Bonn, J., Conger, K., Hwang, J.,Palmer, M. (2014) *PropBank: Semantics of New Predicate Types.* Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, pp. 3013–3019.

Bouma, G., Seddah, D., Zeman, D. (2020) *Overview of the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies.* Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies, ACL, online, pp. 151–161.

Bouma, G., Seddah, D., Zeman, D. (2021) *From Raw Text to Enhanced Universal Dependencies: The Parsing Shared Task at IWPT 2021.* Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies, ACL, online, pp. 146–157.

Brants, S., Dipper, S., Hansen, S., Leziusy, W., Smith, G. (2002). *The Tiger Treebank.* Proceedings of Workshop on Treebanks and Linguistic Theories, Sozopol, pp. 24–41.

Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T. (2018). *Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation.* Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Brussels, pp. 55–64.

Chomsky, N. (1957). *Syntactic Structures*, Mouton & Co, The Hague.

Collins, M. (2002). *Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms.* Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Vol. 10 (EMNLP 2002), pp. 1–8.

Deksne, D., Skadiņš, R. (2011). *CFG Based Grammar Checker for Latvian.* Proceedings of the 18th Nordic Conference of Computational Linguistics, NEALT Proceedings Series, Vol. 11, Riga, pp. 275–278.

Dozat, T., Qi, P., Manning, C.D. (2017). *Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task.* Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Vancouver, pp. 20–30.

Džeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Žabokrtský, Z., Žele, A. (2006). *Towards a Slovene Dependency Treebank.* Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, pp. 1388–1391.

Grūzītis, N. (2011). *Ierobežotas latviešu valodas formālā gramatika un semantika.* Promocijas darba kopsavilkums, Latvijas Universitātes Datorikas fakultāte, Rīga.

Gruzitis, N., Pretkalnina, L., Saulite, B., Rituma, L., Nespore-Berzkalne, G., Znotins, A., Paikens, P. (2018). *Creation of a Balanced State-of-the-Art Multilayer Corpus for NLU*. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, pp. 4506–4513.

Gruzitis, N., Znotins, A. (2018). *Multilayer Corpus and Toolchain for Full-Stack NLU in Latvian.* Proceedings of the CLARIN Annual Conference 2018, pp. 61–65.

Hajič, J., Böhmová, A., Hajičová, E., Vidová Hladká, B. (2000). *The Prague Dependency Treebank: A Three-Level Annotation Scenario.* A. Abeillé (ed.): Treebanks: Building and Using Parsed Corpora, Kluwer, Amsterdam, pp. 103–127.

Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J., Beška, E. (2004). *Prague Arabic Dependency Treebank: Development in Data and Tools.* Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools, Cairo, pp. 110–117.

Hajič, J., Vidová Hladká, B., Pajas, P. (2001). *The Prague Dependency Treebank: Annotation Structure and Support.* Proceedings of the IRCS Workshop on Linguistic Databases, Philadelphia, pp. 105–114.

Lokmane, I., Saulīte, B. (2023a). *Infinitīva palīgteikumi un teikuma tipu robežgadījumi "Nacionālajā korpusu kolekcijā".* Abstracts of International Scientific Conference "Endzelīns. Language. Time" to commemorate the 150th anniversary of academician Jānis Endzelīns, Rīga, pp. 50–53.

Lokmane, I., Saulīte, B. (2023b). *Klauzāli subjekti "Latviešu valodas sintaktiski marķētajā korpusā".* Abstracts of 58th International Academic Conference in Honour of Prof. Arturs Ozols "Grammar and Word Formation", Rīga, pp. 49–47.

Kalnača, A., Lokmane, I. (2018). *Latvian Indeclinable Participle in -am(ies)/-ām(ies) and Raising vs. Control Constructions.* Verbs, Clauses and Constructions: Functional and Typological Approaches. Medina, Pilar Guerrero, Alonso, Roberto Torre and Raquel Vea Escarza (eds.). Cambridge Scholars Publishing, pp. 275–296.

Koo, T., Collins, M. (2010). *Efficient Third-order Dependency Parsers.* Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010), Uppsala, pp. 1–11.

Marcus, M. P., Kim, G., Marcinkiewicz, M. A., Macintyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B. (1996). *The Penn Treebank: Annotating Predicate Argument Structure.* Proceedings of ARPA Human Language Technology Workshop, San Francisco, pp. 110–115.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., Lee, J. (2013). *Universal Dependency Annotation for Multilingual Parsing.* Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL:HLT 2013), Vol. 2, Sofia, pp. 92–97.

McDonald, R., Pereira, F., Ribarov, K., Hajič, J. (2005). *Non-Projective Dependency Parsing using Spanning Tree Algorithms*. Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005), Vancouver, pp. 523–530.

Mel'čuk, I. (1988). Dependency Syntax: Theory and Practice. State University of New York Press, New York.

Mengel, A., Lezius, W., (2000). *An XML-based encoding format for syntactically annotated corpora.* Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC) Vol. 1, Athens, pp. 121–126.

Nešpore, G., Saulīte, B., Bārzdiņš, G., Grūzītis, N. (2010). *Comparison of the SemTi-Kamols and Tesnière's Dependency Grammars*. Proceedings of the 4th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2010), Frontiers in Artificial Intelligence and Applications, Vol. 219, IOS Press, pp. 233–240.

Nivre, J. (2003). *An Efficient Algorithm for Projective Dependency Parsing.* Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 2003), Nancy, pp. 149–160.

Nivre, J. (2009). *Non-Projective Dependency Parsing in Expected Linear Time.* Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009), Singapore, pp. 351–359.

Nivre, J., Basirat, A., Dürlich, J., Moss, A. (2022.) *Nucleus Composition in Transition-Based Dependency Parsing.* Computational Linguistics 1–38, https://doi.org/10.1162/coli_a_00450

Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D. (2016). *Universal Dependencies v1: A Multilingual Treebank Collection.* Proceedings of the 10th Language Resources and Evaluation Conference (LREC 2016), Portorož, pp. 1659–1666.

Nivre, J., Fang, C. (2017). *Universal Dependency Evolution.* Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), Gothenburg, ACL, pp. 86–95.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D. (2007a). *The CoNLL 2007 Shared Task on Dependency Parsing.* Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007), Prague, pp. 915–932.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi E. (2007b). *MaltParser: A language-independent system for data-*

driven dependency parsing. Natural Language Engineering, Vol. 13(2), pp. 95–135.

Nivre, J., Nilsson, J. (2005). *Pseudo-Projective Dependency Parsing.* Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor, Cambridge University Press, pp. 99–106.

Paikens, P. (2007). *Lexicon-Based Morphological Analysis of Latvian Language.* Proceedings of 3rd Baltic Conference on Human Language Technologies (HLT 2007), Kaunas, pp. 235–240.

Paikens, P. (2017). *Rīku kopa latviešu valodas semantikas analīzei.* Promocijas darba kopsavilkums, Latvijas Universitātes Datorikas fakultāte, Rīga.

Paikens, P., Auzina, I., Garkaje, G., Paegle, M. (2012). *Towards named entity annotation of Latvian National Library corpus.* Proceedings of the 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2012), Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 169–175.

Paikens, P., Grūzītis, N. (2012). *An implementation of a Latvian resource grammar in Grammatical Framework.* Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, pp. 1680–1685.

Paikens, P., Rituma, L., Pretkalniņa, L. (2013). *Morphological analysis with limited resources: Latvian example.* Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), NEALT Proceedings Series, Vol. 16, Oslo, pp. 267–277.

Pajas, P., Štěpánek, J. (2006). *XML-Based Represen-tation of Multi-Layered Annotation in the PDT 2.0.* Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006), Genoa, pp. 40–47.

Popel, M., Mareček, D., Štěpánek, J., Zeman, D., Žabokrtský, Z. (2013). *Coordination Structures in Dependency Treebanks.* Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT 2013), Vol. 1, Sofia, pp. 517–527.

Pretkalniņa, L., Levāne-Petrova, K. (2011). *Preparatory Work for Latvian Treebank.* Proceedings of International Conference CORPUS LINGUISTICS – 2011, St. Petersburg, pp. 53–58.

Pretkalniņa, L., Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011a). *Towards a Latvian Treebank.* M.Á. Mora, M. Carrió Pastor, (ed.): Actas del 3 Congreso Internacional de Lingüística de Corpus. Tecnologias de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus, Candel, Valence, pp. 119–127.

Pretkalniņa, L., Nešpore, G., Levāne-Petrova, K., Saulīte, B. (2011b). *A Prague Markup Language Profile for the SemTi-Kamols Grammar Model.* Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011), Riga, pp. 303–306.

Pretkalniņa, L., Rituma, L. (2012). *Syntactic Issues Identified Developing the Latvian Treebank.* Proceedings of the 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2012), Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 185–192.

Pretkalniņa, L., Rituma, L. (2013) *Statistical syntactic parsing for Latvian.* Proceedings of the 19th Nordic Conference of Computational Linguistics, NEALT Proceedings Series, Vol. 16, Oslo, pp. 279–289.

Pretkalniņa, L., Rituma, L. (2014). *Constructions in Latvian Treebank: the Impact of Annotation Decisions on the Dependency Parsing Performance.* Proceedings of the 6th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2014), Frontiers in Artificial Intelligence and Applications, Vol. 268, IOS Press, pp. 219–229.

Pretkalniņa, L., Rituma, L., Saulīte, B. (2016). *Universal Dependency Treebank forLatvian: a Pilot.* Proceedings of the 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2016), Frontiers in Artificial Intelligence and Applications, Vol. 289, IOS Press, pp. 136–143.

Pretkalniņa, L., Rituma, L., Saulīte, B. (2018). *Deriving enhanced Universal Dependencies from a hybrid dependency-constituency treebank.* Proceedings of the 21sh International Conference "Text, Speech, and Dialogue" (TSD), LNCS, Vol. 11107, Springer Link, pp. 95–105.

Pretkalniņa, L., Znotiņš, A., Rituma, L., Goško, D. (2014). *Dependency parsing representation effects on the accuracy of semantic applications – an example of an inflective language.* Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, pp. 4074–4081.

Qi, P., Dozat, T., Zhang, Y., Manning, C.D. (2018). *CoNLL 2018 Universal Dependency Parsing from Scratch.* Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Brussels, pp. 160–170.

Rituma, L., Nešpore-Bērzkalne, G., Saulīte, B., Pretkalniņa, L.. (2023). *Salīdzinājuma konstrukcijas "Latviešu valodas sintaktiski marķētajā korpusā".* Abstracts of 58th International Academic Conference in Honour of Prof. Arturs Ozols "Grammar and Word Formation", Rīga, pp. 62–63.

Rituma, L., Saulīte, B., Nešpore-Bērzkalne, G. (2019). *Latviešu valodas sintaktiski marķētā korpusa gramatikas modelis.* Valoda: nozīme un forma, 10. s., Latvijas Universitātes apgāds, 200–216. lpp.

Rosa, R., Mašek, J., Mareček, D., Popel, M., Zeman, D., Žabokrtský, Z. (2014). *HamleDT 2.0: Thirty Dependency Treebanks Stanfordized*. Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014), Reykjavik, pp. 2334–2341.

Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice.* Berkeley, International Computer Science Institute.

Schwartz, R., Abend, O., Rappoport, A. (2012). *Learnability-based syntactic annotation design.* Proceedings of the 24th International Conference on Computational Linguistics, (COLING 2012), Mumbai, pp. 2405–2422.

Shi, T., Wu, F.G., Chen, X., Cheng, Y. (2017). *Combining Global Models for Parsing Universal Dependencies.* Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Vancouver, pp. 31–39.

Štěpánek, J., Pajas, P. (2010). *Querying Diverse Treebanks in a Uniform Way.* Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), Valletta, pp. 1828–1835.

Straka, M. (2018): *UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task.* Proceedings of CoNLL 2018: The SIGNLL Conference on Computational Natural Language Learning, ACL, pp. 197-207,

Straka, M., Hajič, J., Straková, J. (2016) *UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing.* In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, pp. 4290–4297.

Tesnière, L. 1959. *Éléments de syntaxe structurale*. Klincksieck, Paris. (Tulkots angliski: Tesnière, L. Elements of Structural Syntax, John Benjamins Publishing Company, 2015.)

Vasiļjevs, A., Skadiņa, I. (2012). *Latvian Language Resources and Tools: Assessment, Description and Sharing.* Proceedings of the 5th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2012), Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 265–272.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L. (1995). *A Model-theoretic Coreference Scoring Scheme* Proceedings of the 6th Conference on Message Understanding (MUC 1995), Columbia, pp. 45–52.

VPP "Humanitāro zinātņu digitālie resursi" (VPP-IZM-DH-2020/1-0001) ziņojums "D2.7.2 – Source code and language models of the final version of the scalable NLP-PIPE", 2022.

Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinkova, S., Hajic, J., jr., Hlavacova, J., Kettnerová, V., Uresova, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C.D., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.C., Sanguinetti, M., Simi, M., Kanayama, H., dePaiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, Ç., Sulubacak, U., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Alcalde, H.F., Strnadová, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonca, G., Lando, T., Nitisaroj, R., Li, J. (2017). *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Vancouver, pp. 1–19.

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J., Petrov, S. (2018). *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.* Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, ACL, Brussels, pp. 1–21.

Znotiņš, A. (2016). *Word Embeddings for Latvian Natural Language Processing Tools.* Proceedings of the 7th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2016), Frontiers in Artificial Intelligence and Applications, Vol. 289, IOS Press, pp. 167–173.

Znotiņš, A., Bārzdiņš, G. (2020). *LVBERT: Transformer-Based Model for Latvian Language Understanding.* Proceedings of the 9th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2020), Frontiers in Artificial Intelligence and Applications, Vol. 328, IOS Press, pp. 111–115.

Znotiņš, A., Cīrule, E. (2018). *NLP-PIPE: Latvian NLP Tool Pipeline.* Proceedings of the 8th International Conference on Human Language Technologies – the Baltic Perspective (HLT 2018), Frontiers in Artificial Intelligence and Applications, Vol. 307, IOS Press, pp. 183–189.

Znotins, A., Paikens, P. (2014). *Coreference Resolution for Latvian.* Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014), Reykjavik, pp. 3209–3213.