TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Stiivo Siider

Süntaksianalüüsil põhinev teksti lihtsustaja

Bakalaureusetöö (9 EAP)

Juhendajad: Sven Aller
Heili Orav

Tartu 2019

Süntaksianalüüsil põhinev teksti lihtsustaja

Lühikokkuvõte:

Bakalureusetöö kirjeldab teksti lihtsustamist, keskendudes peamiselt süntaktilisele lihtsustamisele. Inglise keele puhul on süntaktilise lihtsustamise probleemi käsitletud arvukates teadustöödes. Neid tulemusi rakendatakse bakalaureusetöös eesti keelele. Töö eesmärgiks oli luua veebirakendusena teksti lihtsustaja, mille peamiseks lihtsustamismeetodiks oleks lihtlausestamine, s.t liitlausete jagamine lihtlauseteks. Lihtsustaja kasutab süntaksianalüüsiks loomuliku keele töötluse paketti EstNLTK.

Võtmesõnad:

süntaktiline lihtsustamine, sõltuvussüntaks, eesti keel, loomuliku keele töötlus

CERCS:

P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Text simplifier based on syntax analysis

Abstract:

This bachelor's thesis gives an overview of text simplification, focusing specifically on syntactic simplification to bring it's well-researched theory in English over into Estonian. The purpose of the thesis is to create a web-based text simplification application with it's main method of simplification being sentence splitting. For syntax analysis, the simplifier uses the Estonian natural language toolkit – EstNLTK.

Keywords:

syntactic simplification, dependency syntax, Estonian, natural language processing

CERCS:

P170 Computer science, numerical analysis, systems, control

Sisukord

Sissejuhatus

Keel on inimeste jaoks oluline eneseväljendusvahend, millega on võimalik anda edasi mõtteid, ideid, tahtmisi jms. Seda kasutatakse peamiselt kommunikatsioonis, nii suulises kui ka kirjalikus kõnes. Samuti on keel osa rahvast, kes seda kõneleb, ning paljude, eriti väiksemate riikide ja rahvuste jaoks on nende emakeel väga tähtsal kohal. Eestis on eesti keele jätkusuutlikkuse tagamisega ning arengu planeerimisega tegeletud alates 1998. aastast [1]. Keel pole siiski oluline ainult emakeele rääkijatele, vaid ka teistele rahvustele, kellel on huvi või vajadust sellest aru saada. Näiteks 2017. aasta seisuga on enda hinnangul 41 protsendil teistest rahvustest Eesti elanikest aktiivne eesti keele oskus ning 10% ei oska üldse eesti keelt [2]. Ülejäänud omavad passiivset keeleoskust: nad saavad keelest aru ning räägivad veidi või saavad veidi aru ning ei räägi üldse [2]. Seega on olemas inimesi, kes võiksid lihtsustatud eestikeelsetest tekstitest abi saada.

Töö eesmärk on toetada kirjalikust eestikeelsest kõnest arusaamist, muutes tekstis olevate lausete struktuuri lihtsamaks ning mõistetavamaks. Selleks on kavas luua veebirakendus, mis lihtsustab sissetulevaid lauseid, tuginedes EstNLTK [3] süntaksianalüsaatorile, mis leiab sõnadevahelisi sõltuvusi. Kasutajaliideseks on veebibrauseri laiendus, mis oleks võimeline lugema kasutaja poolt veebilehel markeeritud teksti ning, suheldes serveriga, kuvama selle lihtsustatud variandi.

Töö on jaotatud kaheks osaks. Esimene osa keskendub ingliskeelse tekstilihtsustamise teooriale ning võimalustele, rakendades neid eestikeelses lihtsustamises. Teine osa annab ülevaate kasutatavatest ressurssidest, tehnoloogilistest lahendustest, valminud serverist ning brauserilaiendusest ja analüüsib lihtsustaja kasutustulemusi.

1 Teksti lihtsustamine

Teksti lihtsustamine on protsess, mille käigus muudetakse sisend üheks või rohkemaks lihtsustatud lauseks. Selle saavutamiseks rakendatakse erinevaid süntaktilisi ning leksikaalseid operatsioone [4]. Need operatsioonid on kasutatavad iseseisvatena või kombineerituna erinevates lihtsustamisstrateegiates.

Esialgne eesmärk automaatse tekstilihtsustamise valdkonnas oli kiirendada loomuliku keele töötluse (ingl *natural language processing* e NLP) süsteeme (nt masintõlge ja sisukokkuvõte), kasutades teksti lihtsustamist sisendite eeltöötlusel [5]. Tänapäeval on eesmärgiks pigem pakkuda inimesi abistavaid tööriistu, et kõigil oleks kergem kirjalikku informatsiooni omandada [6]. Sellise lihtsustamisega muudetakse tekst kiiremini loetavamaks, parandatakse selle arusaadavust ning vähendatakse lugeja kognitiivset koormust [7, 8].

1.1 Teksti lihtsustamise vajadus

Teksti lihtsustamine on abiks kõigile, kelle keeleoskus on madal või piiratud. Nendeks võivad olla lapsed, võõrkeelena kõnelejad, kurdid või inimesed erinevate kõne- ja kognitiivsete häiretega, nagu näiteks düsleksia ja afaasia. Mason ja Kendall [9] uurisid, kuidas inimesed loetust aru saavad ning jõudsid järeldusele, et madalama keeleoskusega lugejad teevad vähem vigu, kui keerulised laused on tehtud lihtlauseteks. See tuleneb sellest, et lugedes lühemaid lauseid saavad madala keeleoskusega lugejad rakendada oma töömälu süntaksist arusaamise asemel semantilisele töötlusele.

Erinevad lihtsustamisoperatsioonid mõjuvad sõltuvalt inimesest erinevalt. Näiteks toob Siddharthan [10] välja peamiste operatsioonide kasulikkuse erinevate kõnehäiretega inimestele. Kurte ning afaasiaga inimesi abistab enim lause struktuuri lihtsustamine, sest neil on probleemid kognitiivsete ning keeleliste oskustega, mis on vajalikud pikematest lausetest arusaamisel. Düsleksikute jaoks on aga kasulikum leksikaalne lihtsustamine, kuna nende peamised probleemid on seotud üksikute sõnade ning sõnapaaride lugemisega. Siiski on lihtsam struktuur abiks, sest see vähendab pingutust loetava korrektsel tõlgendamisel.

Long ja Ross [11] ning Oh [12] leiavad teise keelena õppijate vaatest, et lihtsustamine pigem takistab keele omandamist, kuna lugeja jääb ilma keele autentsetest keelelistest konstruktsioonidest ning sõnavarast. Nende arvates on palju olulisem teksti täiendada ning selgitada.

1.2 Leksikaalne lihtsustamine

Leksikaalne lihtsustamine on teksti lihtsustamisviis, mille käigus asendatakse lauses olevaid keerulisi sõnu lihtsamatega, kandes edasi võimalikult hästi esialgse teksti mõtet ning säilitades sisendi süntaksit [13]. Tulemusena saadud lause mõistmine on kergem, kuna lihtsad sõnad on üldjuhul sagedasemad ja lugejale tuttavamad. Inglise keeles on teksti piisavaks arusaamiseks vaja teada 95% kasutatavast sõnavarast [14]. Sama idee võib kanda üle ka eesti keelele: lugeja peab teadma suurt osa teksti sõnavarast, et seda mõista. Seega vähendades leksikaalse lihtsustamisega vähetuntud sõnade arvu suureneb lugejale tuntud sõnade protsent tekstis.

Shardlow [15] esitleb traditsioonilist leksikaalset lihtsustamist neljasammulise protsessina, mida alustatakse tekstis olevate keeruliste sõnade identifitseerimisega. Seejärel leitakse iga keerulise sõna sünonüümide hulk, millest jäetakse alles ainult teksti konteksti sobivad. Lõpuks järjestatakse allesjäänud sünonüümid nende lihtsuse ja sobivuse järgi ning neist parim valitakse asenduseks. Sarnasele protsessijaotusele tugineb Peedoski [16] eestikeelne tekstilihtsustaja, mis asendab keerulised sõnad sünonüümide või ülemmõistetega.

Klassikalist leksikaalset lihtsustamist uurides leidsid Rello jt [6], et sõnade asendamine mõjub tolle aja tipplahendustes lugemiskogemusele ning tekstist sügavamalt arusaamisele pigem negatiivselt. Ühe võimaliku põhjusena tõid nad välja, et lihtsustatava teksti kontekst võib muutuda, kui asenduseks valitud sõna mõte ei ühti piisavalt esialgsega või kui moodustub mõni kummaline sõnade kombinatsioon, mis on teisiti tõlgendatav. Asenduseks pakutud vale sõna tekitab probleeme, kuna teksti mõistmine sõltub pigem sõnadevahelistest seostest kui üksikutest sõnadest, mistõttu kaob seos ümbritsevate sõnadega. Nende läbiviidud katsetel andsid parimaid tulemusi mitte-düsleksikute puhul esialgsed tekstid ning düsleksikute puhul soovi korral sünonüüme pakkuvad tekstid. Lisaks sellele on toodud välja süsteeme [11, 12] võõrkeelena õppijatele, mis kasutavad sõna asendamise asemel nende selgitamist. Sellised süsteemid pole andnud alati paremaid tulemusi, kuid need aitavad õppida loomulikumat keelt. Selgitatud laused on üldiselt pikad ja lingvistiliselt keerukad, mis võivad lugemise muuta keerulisemaks, kuid selle lahenduseks on pakutud kasutada süntaktilist lihtsustamist [11].

1.3 Süntaktiline lihtsustamine

Süntaktiline lihtsustamine on teksti lihtsustamise oluline osa, kuna see on ainus meetod, millega on võimalik lahti saada grammatilistest keerukustest ning on seetõttu leidnud laia kasutust olemasolevates lihtsustajates [15]. Grammatiliselt keerukas tekst on tihti selline, mida lugedes võib kaduda teksti mõte või seosed erinevate tekstiosade vahel, mistõttu on loetava korrektne tõlgendamine raskendatud. Seepärast asendatakse lihtsustamise protsessis kindlaid süntaktilisi konstruktsioone, et muuta teksti inimestele loetavamaks või programmidele töödeldavamaks [17].

Süntaktiline lihtsustamine jaguneb traditsiooniliselt kolmeks etapiks: analüüs, transformatsioon ning regeneratsioon [17, 15]. Järgnevalt antaksegi neist ülevaade.

1.3.1 Analüüs

Analüüsi etapis peamiselt uuritakse sisendlauseid, et oleks piisavalt teavet sisendi lihtsustamiseks. Lisaks hinnatakse selles etapis lause keerukust, et teada saada, kas sisend on lihtsustatav. Hindamine tugineb peamiselt süntaksianalüüsi poolt leitud sõltuvusstruktuurile [15]. Sageli kogutakse ka muud informatsiooni, näiteks infot sõnaliikide kohta, mille jaoks kasutatakse morfoloogilist analüüsi [17]. Morfoloogilise info kogumine on sõltuvusstruktuuri leidmisest kiirem, mistõttu saab enne süntaksianalüüsi hinnata lause sobivust lihtsustamiseks ning võimalusel saab ka töö lõpetada, vähendades lause peale kuluvat aega märgatavalt. Võttes näiteks lihtlausestamise võib ühe tegusõnaga laused vahele jätta, kuna tegemist on tõenäoliselt lihtlausetega.

Süntaksianalüüs on protsess, mille käigus uuritakse lause struktuuri ehk süntaksit. Selle tulemuseks on sõnade vahelisi seoseid kujutav sõltuvuspuu [18, 19]. Saadud sõltuvuspuu (joonis 1) koosneb tippudest, mis tähistavad üksikuid sõnesid lauses, ning kuna igal tipul võib olla ainult üks ülemus, kuid mitu alluvat, siis viitavad tipud enda ülemusele [20]. Üldjuhul kasutatakse viitamiseks paari, mis koosneb ülemuse indeksist ehk sõltuvussuhtest ning tippu iseloomustavast süntaktilise funktsiooni märgendist [20, 21, 22].

Joonis 1. Sõltuvuspuu näide [23]

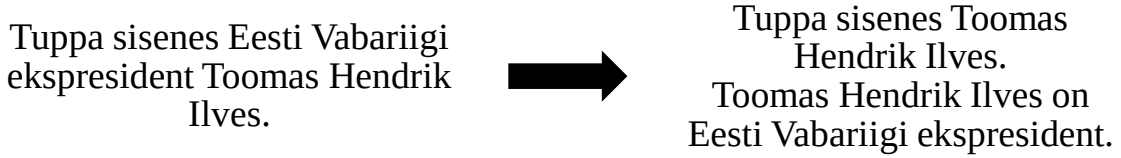
Eesti keele süntaksianalüüsiks on kasutatav EstNLTK loomuliku keele töötlemise tööriista sõltuvuspõhine analüüs [21], mis vaikimisi toetub eeltreenitud MaltParserile [24], mis on süsteem andmepõhiseks sõltuvusanalüüsiks. Lisaks on EstNLTK-s olemas teist tüüpi, reeglipõhine analüsaator VISLCG3 [25]. Süntaktiliste funktsioonide märgendamiseks on neil mõlema analüsaatori jaoks kasutusel märgendite hulk [26], mis sarnaneb Müürisepa esitletud ja Eesti keele kitsenduste grammatika jaoks kasutatud märgendite hulgal [19].

Analüüsi etapi osana võib ka sõnu ja fraase grupeerida, mille käigus luuakse nn ülemmärgendid, mis esindavad enda alluvuses olevat lauseosa ning võimaldavad käsitleda sisendlauset lihtsama struktuurina [15]. Näiteks võib lause osa *Eesti Vabariigi president Toomas Hendrik Ilves* grupeerida kui *Lisand(Eesti Vabariigi president) Pärisnimi(Toomas Hendrik Ilves)*, mida on võrreldes üksikute sõnadega lihtsam transformeerida.

1.3.2 Transformatsioon

Transformatsiooni etapis muudetakse lausestruktuuri vastavalt käsitsi kirjutatud või automaatselt genereeritud reeglitele, mille järgi teostatakse erinevaid lihtsustusi [15]. Inglise keeles on peamisteks lihtsustatavateks grammatilisteks konstruktsioonideks lisand, rinnastusseos, alistusseos, relatiivlause ja tegumood [10]. Eesti keelega võrreldes on antud konstruktsioonid teisiti käsitletavad ning tõlgendatavad, seega on käesolevas töös toodud välja iga nimetatud konstruktsiooni lühikirjeldus ja sobivus lihtsustamiseks.

Lisand [27] on nimisõna täiend, mis esitab teises mõttes seda sama nimisõna, millega kaasas ta on. Lisandi lihtsustamiseks (joonis 2) tuleb see eraldada nimisõnast ning tekitada sellega eraldi lause.



Joonis 2. Lisandi lihtsustamine.

Rinnastusseos [28] esineb süntaktiliselt võrdväärsete lauseosade vahel. Selle lihtsustamiseks (joonis 3) tuleb suuta eraldada seotud lauseosad ning moodustada nendega eraldi lausedd.

Joonis 3. Rinnastusseose lihtsustamine.

Alistusseos [28] on sarnaselt rinnastusseosele lauseosade vahelise seose tüüp. Alistusseos esineb, kui üks lauseosa ehk laiend, allub teisele lauseosale ehk põhjale. Näiteks on selliseks lauseks *Poiss kirjutab kirja*, kus *kirja* on laiend ja *kirjutab* on põhi. Liitlauseliseks näiteks on *Meie soov on see, et kõik rahule jääks*, kus liitlause esimene osalause on põhi ja teine laiend. Sellisel kujul seosed pole üldiselt lihtsustatavad, kuna osapoolte vaheline seos on tugev.

Alistusseose lihtsustatavaks juhuks on relatiivlause [28], mis on alistusseoses oleva liitlause alistuv osalause, mille sidend viitab põhilauses olevale nimisõnale, mille või kelle kohta see osalause käib. Relatiivlause lihtsustamiseks (joonis 4) tuleb see eraldada iseseisvaks lauseks ning asendada sidend nimisõnaga.

Joonis 4. Relatiivlause lihtsustamine

Tegumood [28] näitab lause subjekti ja tegevuse vahelist seost. Võimalikeks tegumoodideks on isikuline, kus lauses on olemas subjekt, ja umbisikuline, kus subjekt pole kirjas. Eesti keeles pole tegumood lihtsustatav, sest umbisikulises lauses ei ole võimalik selguse loomiseks tekitada korrektset subjekti, et muuta see isikuliseks.

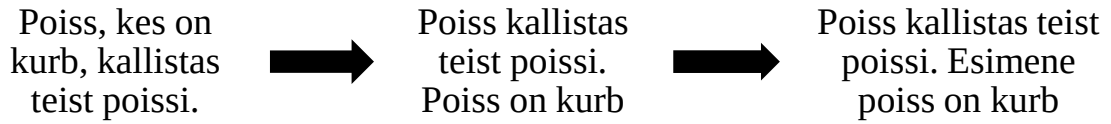
Grammatiliste konstruktsioonide lihtsustamiseks on kasutusel mitmed erinevad võtted. Aluísio jt [29] toovad välja viis üldist lihtsustamise operatsiooni:

- lihtlausestamine;
- diskursuse markerite ehk lauseid ja ideid ühendavate fraaside asendamine lihtsamate või sagedamastega;
- passiivse tegumoe muutmine aktiivseks;
- klauslite ehk praeguses kontekstis osalausete järjestuse muutmine;
- lausete viimine subjekt-verb-objekt kujule.

Eelnevalt kirjeldatud eesti keeles olevate konstruktsioonide lihtsustamiseks on vaja ainult lihtlausestamist. Teised meetodid, v.a tegumoe muutmine, on kasutatavad üldisemaks lausestruktuuri muutmiseks.

1.3.3 Regeneratsioon

Regeneratsioon on lihtsustamise valikuline etapp, mille käigus tehakse muudatusi lause süntaksipuus ja sõnades, et parandada loetavust, seotust ning asjakohasust [15]. Näiteks joonisel 5 tekib pärast transformatsiooni kaheti mõistetavus, kumb poistest on kurb. Selle parandamiseks saab regeneratsiooni etapp taastada asesõnalise seose genereerides juurde mingi viitava sõna, antud juhul *Esimene*.



Joonis 5. Sisendi transformatsioon, millele järgneb regeneratsioon.

Selle etapi tõi esimesena esile Siddharthan [17], kuna see võimaldab parandada transformatsioonil tekkivaid diskursuse tasemel probleeme, mis võivad vähendada teksti sidusust või muuta lausete mõtet. Teksti sidusust käsitleb Siddharthan kahel eraldi viisil: side- ning asesõnaline sidusus. Tema pakutud lahendus suurendab sidesõnalist sidusust parandades lausete järjekorda, genereerides siduvaid sõnu, mis väljendaksid eraldiseisvate lausete vahelisi seoseid, ning luues sobivaid tagasiviitavaid väljendeid, et pronoomenite asendamisel vältida nii mitmemõttelisust kui ka ebaloomulikust. Asesõnalise sidususe suurendamiseks uuritakse lihtsustatud teksti asesõnade seotust asendatava sõnaga ning ebamäärase seose korral asendatakse pronoomen varemloodud tagasiviitava väljendiga.

1.4 Masintõlge

Tänapäevastes tekstilihtsustamissüsteemides leiab aina enamat kasutust masintõlge [10, 15, 30]. Masintõlge on automatiseeritud kahe keele vaheline tõlge, mis on üks loomuliku keele töötluse kasutussüsteemidest.

Teksti lihtsustamiseks võeti kõigepealt kasutusele fraasipõhine statistiline masintõlge [10]. Statistiline masintõlge põhineb masinõppe meetoditel, rakendades õppimisalgoritmi varem tõlgitud paralleelkorpusele, et saada statistiline mudel, millega saab uusi lauseid tõlkida [31]. Fraasipõhine tõlkimine käsitleb lauses esinevaid fraase tervikuna, mis on oluline, kuna sõnapõhisel tõlkimisel ei arvestata ümbritseva kontekstiga, mis võib tõlget muuta [32].

Masintõlge sobib teksti lihtsustamiseks, kuna seda saab vaadelda kui esialgse keele tõlkimist lihtsustatud keeleks. Lisaks tugineb lihtsustamine tugevalt lause kontekstile, et teha sobivad muutusi sõnades või fraasides, mistõttu on töötamine fraasidega vajalik. Kirjeldatud ükskeelne tõlkimine on võimalik, sest üldjuhul on lihtsustatud variant keelest piisavalt erineva süntaksi ning sõnavaraga, et tõlkimise tulemuseks oleks uus tekst [15]. Selle edasiarenduseks on kujunenud närvivõrgupõhine masintõlge [33, 34], mis on traditsioonilises masintõlkes häid tulemusi näidanud [35, 36].

1.5 Sarnaste tööde ülevaade

Kuigi on loodud mitmeid tekstilihtsustajate süsteeme [5, 6, 4, 37, 38], siis veebirakendustena on olemas neist vähesed. Suurem osa veebirakendustest on leksikaalsed lihtsustajad, millest on ülevaate andnud Peedosk [16]. Süntaktilist lihtsustamist pakkuvaid rakendusi on vähem, neist kasutatavad on näiteks YATS [38] ja MUSST [39].

Lihtsustajate võrdlemiseks kasutatakse sama sisendteksti (joonis 6), mis sisaldab kõiki Siddharthani [10] väljatoodud lihtsustatavaid grammatilisi konstruktsioone.

A small and fluffy dog was carried by Jane Doe, my friend, to their new home. Because Jane wanted to talk to the previous owner of her house, she and her dog met with the old and weary homeowner, who was wearing a big trenchcoat and was smoking

Joonis 6. Sisendtekst koos markeeritud konstruktsioonidega

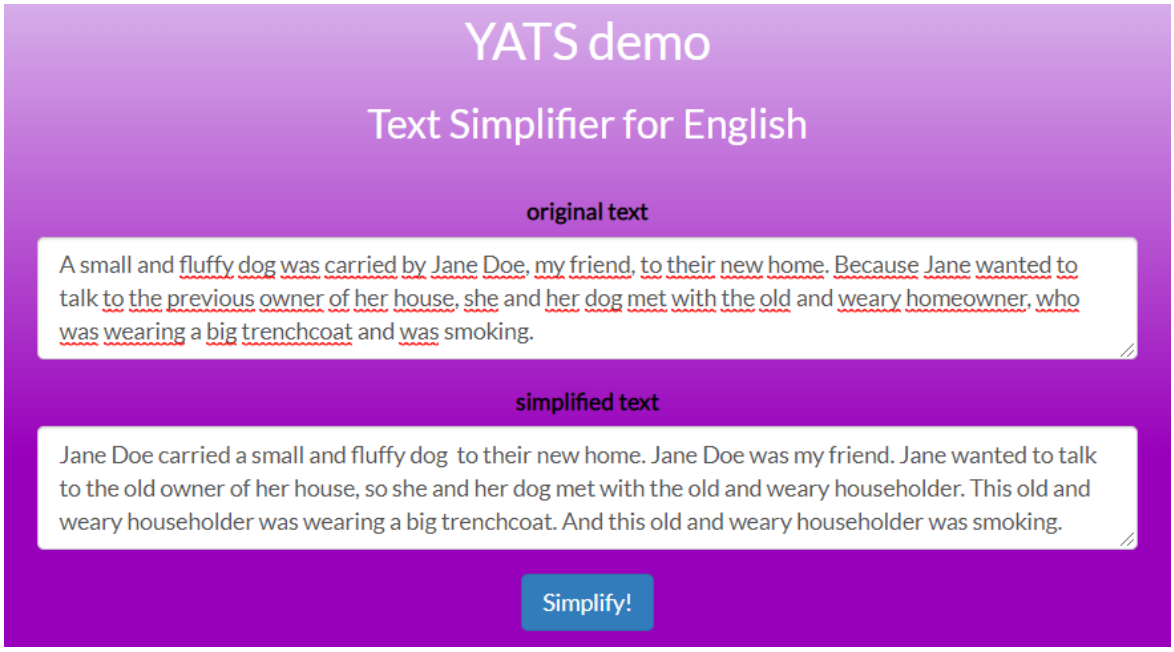
Joonisel 6 on punasega märgitud passiivne tegumood, rohelisega pärisnimi koos lisandiga, pruuniga alistusseose alistuv osalause, lillaga relatiivlause ning nimisõnafraas, millele see viitab, ja sinisega rinnastusseos.

1.5.1 YATS – Yet Another Text Simplifier

Ferrés' jt [38] loodud tekstilihtsustaja YATS [40] on võimeline teostama nii leksikaalset kui ka süntaktilist lihtsustamist. Nende leksikaalne lihtsustaja asendab vähem esinevaid sõnu sagedasematega, arvestades ka sõna kontekstiga, et asendus oleks adekvaatne. Süntaktiline lihtsustus on neil reeglipõhine protsess, mis on jagatud kahte etappi: dokumendi analüüs ja lause generatsioon.

Dokumendi analüüsi etapis analüüsitakse sisendit, millest saadakse kätte lauseosade vahelised sõltuvused ning leitakse ja märgendatakse erinevaid süntaktilisi nähtusi: lisand, relatiivlause, rinnastusseos, ühendsidendid, passiivne tegumood, määruslause ja alistusseos [38].

Lause genereerimise faasis [38] kasutatakse analüüsi tulemusi, et luua lihtsakoelisemad lausestruktuurid. Selleks rakendatakse tekstile reegleid, mis lihtsustavad eelnevalt väljatoodud süntaktilisi nähtusi kasutades lihtlausestamist, sõnajärjekorra muutmist, sõna asendust, tegusõnade ajavormide kohandamist, asesõnade asendamist ning sõnade suur- ja väiketähestamist.



Joonis 7. Veebirakenduse YATS kasutamine sisendtekstiga

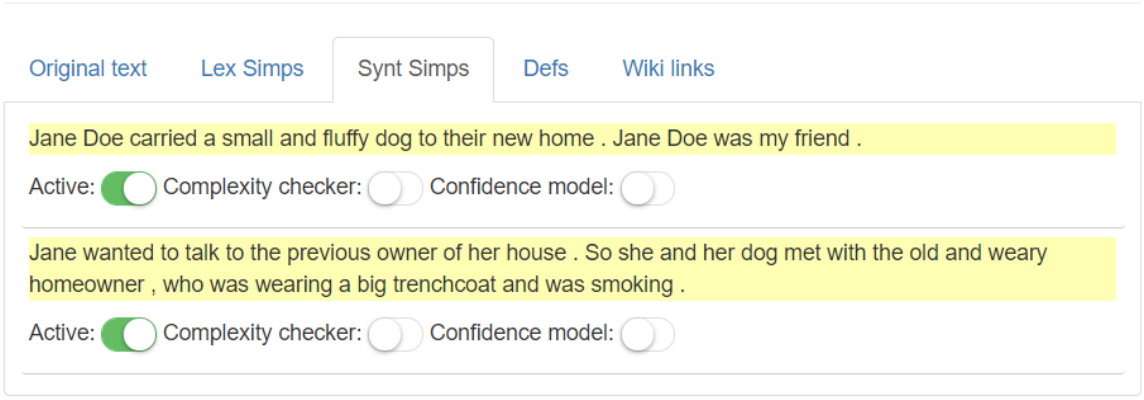
Joonisel 7 on näha sisendteksti lihtsustuse tulemus. YATS lihtsustas kõik viis esitletud grammatilist konstruktsiooni. Lihtsustaja silmnähtavaks probleemiks on nimisõnafraasi (*old and weary householder*) terviklik kordamine relatiivlause ning rinnastusseose lihtsustamisel, mis tekitab liigselt kordusi ning muudab teksti ebaloomulikuks.

1.5.2 MUSST - Multilingual Syntactic Simplification Tool

Scartoni jt [39] SIMPATICO projekti[1] raames loodud mitmekeelne tekstilihtsustaja MUTTS [41] on võimeline lihtsustama sisestatud tekstide sõnavara ning lausete struktuuri,

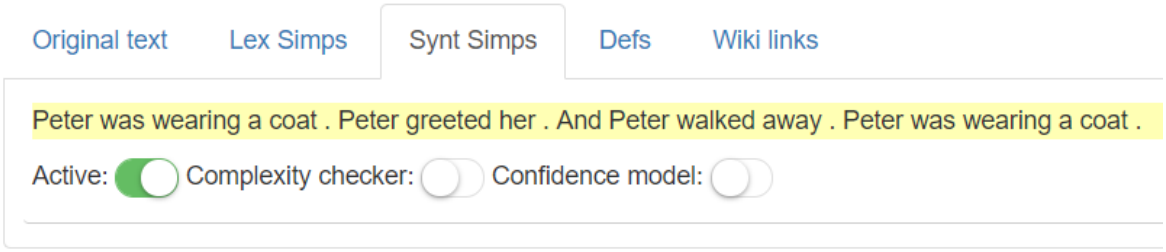
[1] https://www.simpatico-project.eu/

kuid peamine rõhk on viimasel. Samuti leiab nende süsteem originaaltekstis esinevatele sõnadele definitsioone või neile vastavaid Wikipedia artikleid. Loodud lihtsustaja on võimeline käsitlema itaalia, inglise, galeegi ning hispaania keelseid tekste ning seda on võimalik kergesti laiendada teistele keeltele. Süntaktilise lihtsustaja jagavad nad kolmeks osaks: analüüs, transformatsioon ning generatsioon. Lihtsustatavateks konstruktsioonideks on valitud ühendatud osalaused, relatiivlause, lisand ning passiivne tegumood. Lisafunktsionaalsusena on lisatud keerukuse kontrollija, mis otsustab, kas lauselihtsustamine on vajalik, ning usaldusmudel, mis hindab lihtsustatud lause sobivust.



Joonis 8. Veebirakenduse MUSST süntaktilise lihtsustamise tulemus sisendtekstil

Joonisel 8 kujutatakse süntaktilisi lihtsustusi, millel on näha ka lisafunktsionaalsuse valikud. Sisendtekstis esinevatest konstruktsioonidest lihtsustati alistusseos, lisand ning passiivne tegumood. Lihtsustamata jäid rinnastusseos ning relatiivlause, mida teiste sisendite puhul lihtsustati (joonis 9). Kasutades lisafunktsionaalsustest keerukuse kontrollijat jäi lihtsustamata esimene lause, kasutades usaldusmudelit teine lause.



Joonis 9. Lause *Peter, who was wearing a coat, greeted her and walked away.* lihtsustus

Võrreldes YATS lihtsustajaga on MUSST palju ettevaatlikum, eriti kui kasutada lisafunktsionaalsusi, mis vähendavad tehtavate lihtsustuste esinemist.

2 Programmi ülevaade

Koostatud programm koosneb tagarakendusest (ingl *backend*) ja eesrakendusest (ingl *front-end*). Tagarakenduse tööks on loodud sõnestaja, lausestaja ning rakendusliides (ingl *application programming interface* e API [42]), mis on lihtsustaja sidekanaliks nii loodud eesrakendusega kui ka teiste võimalike programmide ning süsteemidega. Lihtsustaja arendamiseks on kirjutatud ka testija, et näha, kuidas programm erinevate sisenditega hakkama saab. Terve programmi lähtekood on olemas GitHubis[2].

Teksti lihtsustaja rakendab kahte Aluísio jt [29] väljatoodud lihtsustamise operatsiooni: lihtlausestamist ja lausete subjekt-verb-objekt kujule viimist. Väiksemate muudatustena teeb lihtsustaja sarnaselt rakendusele YATS [38] asesõnade asendamist ning suur- ja väiketähestamist. Lihtsustatavatest konstruktsioonidest on lihtsustaja võimeline käsitlema relatiivlauseid ning liitlauselisi rindlauseid. Erinevalt esitletud üldisest teooriast ei grupeerita lihtsustajas sõnu fraasideks v.a jutumärkide või sulgude sees olevaid sõnu rekursiivse lihtsustamise eesmärgil. Üksikute sõnade tasemele jäämise põhjuseks on eesmärk tugineda EstNLTK süntaksianalüsaatorile, mis annab süntaktilist infot iga individuaalse sõne kohta.

Lihtsustajas on olemas varem esitatud [15] ülesehituse kohaselt analüüsi ja transformatsiooni etapp, kuid puudu on regeneratsioon, kuna see vajab keerulisemat tulemuste analüüsimis- ja hindamissüsteemi, mis teeks kindlaks transformatsiooni jooksul ning lõpus loodud lause vead ning oleks võimeline rakendama sobivaid parandusi.

2.1 Kasutatud tehnoloogilised lahendused

Lihtsustaja toetub oma töös tugevalt EstNLTK (v1.4.1) teegile, mis pakub morfoloogilist analüüsi, süntaksianalüüsi ning sünteesimist. EstNLTK kasutamiseks on lihtsustaja loodud programmeerimiskeeles Python. Lihtsustaja ehitamist alustati Python 2.7-s, mis hiljem asendati Python 3.4.2-ga, et kasutada uuemat, edasiarendatud versiooni. Asendamiseks oli vaja lisaks uurida ja muuta lihtsustaja ja API vahelist suhtlust, kuna seal oli mitmeid probleeme kodeeringu ühtlustamisel. Lihtsustaja rakendamiseks on vaja PHPd toetavat serverit

[2] https://github.com/StiivoSiider/TekstiLihtsustaja-Siider

(nt Apache 2.4.x[3]), millel on lubatud kasutada Pythonit (versioon 3.4 või 3.5, uuemad versioonid ei tarvitse EstNLTKga töötada) ning milles oleks installeeritud EstNLTK 1.4.1 koos kõigi vajalike sõltuvustega[4].

API on kirjutatud PHPs, kuna see võimaldab kergelt käsitleda veebisuhtlust ning käivitada serveris olevaid faile. Samuti on selles võimalik panna kokku HTMLis (ingl *Hypertext Markup Language*) veebilehti. See võimalus võeti kasutusse API lisavõimaluste loomisel.

Eesrakendus on veebibrauseri Chrome laiendus [43], mistõttu on selle loomiseks peamiselt kasutatud JavaScripti. Laienduse kasutajaliidese akna loomiseks on lisaks kasutatud HTMLi ja CSSi (ingl *Cascading Style Sheets*). Selle installeerimiseks (pildina juhend lisas 1) on vajalikeks eesrakenduse failide lahtipakitud kaust[5] ning arendajarežiimi lülitatud[6] Google Chrome (versioon 74.x või uuem). Arendaja režiim on vajalik Chrome'i veebipoeväliste laienduste kasutamiseks. Chrome'i pakkimata laienduste paigaldamise võimalust kasutades saab eesrakenduse lisada enda veebilehitsejasse.

2.2 Tagarakendus

Tagarakendus on programmi osa, kus toimub peamine programmi töö. Selle tuumaks on lihtsustaja, mis eeltöötleb sisendit, tokeniseerib selle ning saadab edasi analüüsimiseks ja transformeerimiseks. Lisaks on tagarakenduses olemas API, mis laseb teistel programmidel talle päringuid saata, ning testija.

2.2.1 Eeltöötlus

Tagarakendusse jõudvale sisendile rakendatakse esimesena eeltöötlust, mis kasutab Pythoni regulaaravaldiste mooduli asendusmeetodit. Selle sammu eesmärk on ühtlustada võimalikke jutumärke ning sulge ja eemaldada reavahetusi ning lihtsustatava teksti APIst tagarakendusse transportimisel tekkivaid sümboleid. Tänu ühtlusele ei pea programm edaspidi arvestama mitmete erinevate karakterite ning nende kombinatsioonidega.

[3] https://httpd.apache.org/

[4] https://estnltk.github.io/estnltk/1.4.1/tutorials/installation.html

[5] https://github.com/StiivoSiider/TekstiLihtsustaja-Siider/releases/tag/v1.0

[6] https://developers.chrome.com/extensions/faq#faq-dev-01

2.2.2 Tokeniseerijad

Tokeniseerimiseks on laiendatud EstNLTK pakutavat sõnestajat ja lausestajat.

Loodud sõnestaja (joonis 10) eesmärk on grupeerida kokku sulgudes ja jutumärkides olev tekst nii, et neid saaks paremini kasutada nii rekursiivsel lihtsustamisel kui ka lausete kokkupanekul. Jutumärkide puhul otsitakse järgmine jutumärk ning moodustatakse uus sõne, kuid sulgude puhul peab alati jõudma tagasi samale tasemele, et vältida sobimatuid sulgude paare. Äärmised sulud ja jutumärgid jäävad siiski alles iseseisvate sõnedena, et analüsaator teaks, et tegemist on jutumärkide või sulgudega ümbritsetud tekstiga.

```
['"', 'Sa', 'ei', 'kuulu', 'siia', '!"', 'hõikas', 'Jüri', '.']
['"', '"Sa ei kuulu siia!"', '!"', 'hõikas', 'Jüri', '.']
```

Joonis 10. Sõnestaja näide. Üleval on EstNLTK sõned, all muudetud sõned.

Loodud lausestaja (joonis 11) eesmärk on vältida olukordi, kus EstNLTK lausestaja lõhub jutumärkides oleva teksti mitmeks lauseks. Vältimiseks leiab lausestaja jutumärkidega algava, kuid mitte jutumärkidega lõppeva lause ning leides lõpu võtab see vahepealsed laused kokku üheks.

```
['"Ja nii on.', 'Nii on."', 'ütles Jüri.']
['"Ja nii on. Nii on."', 'ütles Jüri.']
```

Joonis 11. Lausestaja näide. Üleval on EstNLTK laused, all muudetud laused.

Neid tokeniseerijad kasutatakse analüüsi etapis, et sisend oleks edasiseks tööks sobivalt sõnestatud ja lausestatud.

2.2.3 Analüüs

Analüüsi etapi eesmärgiks on koguda transformatsiooniks vajalikku infot ning hinnata, kas sisend vajab muutmist. Alustuseks luuakse sisendist EstNLTK *Text* objekt, milles kasutatakse ülalmainitud tokeniseerijaid (joonis 12).

```
kwargs = {
    "word_tokenizer": CustomWordTokenizer(),
    "sentence_tokenizer": CustomSentenceTokenizer()
}

print(Text('Sisend lause (lühike)', **kwargs).word_texts)

> ['Sisend', 'lause', '(', '(lühike)', ')']
```

Joonis 12. *Text* objekti loomine kasutades teisi tokeniseerijaid.

Seejärel lihtsustatakse *Text* objekti lauseid ükshaaval, kuna lihtsustaja ei ole võimeline arvestama terviklike lausete vaheliste seostega, mistõttu on lihtsam lauseid individuaalselt töödelda. Siin leitakse ka lauses olevad sulgude ja jutumärkide vahelised tekstid ning lihtsustatakse neid rekursiivselt.

Esimeseks sisendi hindamiseks vaadatakse sisendi morfoloogilise analüüsi tulemusena saadud sõnaliike (joonis 13). Kui leitakse üks või null tegusõna, siis käesoleva lause töötlemine lõppeb, kuna tegemist pole liitlausega. Lisaks väiketähestatakse lause esimene sõna, v.a juhul, kus EstNLTK arvates on tegemist pärisnime või lühendiga.

```
sisend = Text('Poiss, kes istub pingil, on väike!')

print(sisend.postags)

> ['S', 'Z', 'P', 'V', 'S', 'Z', 'V', 'A', 'Z']
```

Joonis 13. *Text* objekti sõnaliikide vaatamine.

Järgmisena analüüsib lihtsustaja lause sõltuvussüntaksi kasutades EstNLTK analüsaatorit (joonis 14). Analüüsi tulemuseks on järjend, mis koosneb sõnadele vastavatest sõnastikest. Sõna sõltuvust tema ülemaga iseloomustab *parser_out* väärtus, mis näitab ülemuse indeksit ning seda, mis liiki nendevaheline seos on.

```
sisend = Text('Poiss on väike!')
sisend.tag_syntax()
print(sisend[LAYER_CONLL])

> [{'end': 5, 'parser_out': [['@SUBJ', 1]], 'sent_id': 0, 'start': 0},
   {'end': 8, 'parser_out': [['ROOT', -1]], 'sent_id': 0, 'start': 6},
   {'end': 14, 'parser_out': [['@PRD', 1]], 'sent_id': 0, 'start': 9},
   {'end': 15, 'parser_out': [['xxx', 2]], 'sent_id': 0, 'start': 14}]
```

Joonis 14. *Text* objektil süntaksianalüüsi teostamine ja tulemuse vaatamine.

Itereerides üle saadud informatsiooni, muudetakse kõigepealt sulgudes olevate sõnade sõltuvus- ning sõnaliiki. Kuna sõltuvustabelist (joonis 15) jäetakse välja kõik kirjavahemärgid, siis muudetakse ka sõna ülemust, mida otsitakse madalamatelt indeksitelt, sest sageli käib sulgudes olev tekst temale eelneva sõna kohta. Sellele lisaks koostab lihtsustaja varem mainitud sõltuvustabelit, milles sõna asukohale lauses vastab tema alluvate objektide järjend, ning lause järjendit, milles on loodud objektid järjestatud vastavalt nende asukohale lauses. Iga sõna kohta kogub lihtsustaja kokku tema indeksi, lemma, sõnavormi, sõnaliigi, ülema indeksi, sõna enda ning sõltuvusanalüüsi märgendi. Lisaks jagatakse informatsiooni kogumise ajal lause pea- ja tegusõnad järjenditesse.

```
> {-1: [{'form': 'b',
         'indeks': 1,
         'label': 'ROOT',
         'lemma': 'olema',
         'pos': 'V',
         'target': -1,
         'word': 'on'}],
    1: [{'form': 'sg n',
         'indeks': 0,
         'label': '@SUBJ',
         'lemma': 'poiss',
         'pos': 'S',
         'target': 1,
         'word': 'Poiss'},
        {'form': 'sg n',
         'indeks': 2,
         'label': '@PRD',
         'lemma': 'väike',
         'pos': 'A',
         'target': 1,
         'word': 'väike'}]}
```

Joonis 15. Sõltuvustabel lausele *Poiss on väike!*

Joonisel 15 on välja toodud osad märgendid, mis põhinevad EstNLTK poolt kasutataval märgendushulgal [26]. Seega saab lause *Poiss on väike!* sõltuvustabelist välja lugeda, et *Poiss* on lause subjekt (*@SUBJ*), *on* on lause juur (*ROOT*) ning *väike* on predikatiiv ehk öeldistäide (*@PRD*).

Programm kontrollib seejärel tingimust, et lauses oleks ainult üks juur. Juhul kui meil on mitu juurt, on lihtsustamine raskendatud, kuna analüüsija käsitleb ühte lauset kui mitut ehk süntaksipuu asemel on süntaksimets. Seejärel liigutakse edasi transformatsiooni etappi.

2.2.4 Transformatsioon

Transformatsiooni alustatakse sõltuvuspuus otseses ülem-alluv seoses olevate peasõnade eraldamisega teistest. Selle sammuga lihtsustatakse peamiselt rindlauseid, kuid ka relatiiv-lauseid, mille tegusõna viitab teisele tegusõnale. Siin vaadatakse iga tegusõna ning juursõna alluvat ning kui alluv on peasõna ning nende kahe vahel on asesõna *kes* või *mis* või sidend *ja* või *või*, siis üritatakse neid eraldada. Edukaks eraldamiseks peab alluval olema subjekt, mille ta leiab kas ülema tegusõna kaudu või enda alluvate hulgast. Lisaks peab juursõna puhul olema tegemist tegusõnaga, et temast saaks alluvat peasõna eraldada. Vastasel juhul on sageli tegemist subjektiga, mis on seekord enda kohta käiva tegusõna ülem ning nende eraldamine ei oleks mõistlik.

Seejärel kontrollitakse iseseisvad tegusõnu ehk neid, mille ülem ei ole teine tegusõna ega juurtipp. Kui lauses on alles vähem kui kaks iseseisevat tegusõna, siis lõpetatakse lausega töötamine, kuna sisendis oleks ainult üks või mitte ühtegi sõna, mis suudaks lauset kanda.

Transformatsiooni jätkatakse asesõnade asendamisega, et lihtsustada relatiivlauseid. Semantilise info puuduse tõttu eeldab lihtsustaja, et iga tegusõna kohta saab asendada ainult ühe asesõna, kuna asesõnade seos sellega, millele nad osutavad, tuleb läbi tegusõnade. Samuti piiratakse asenduse suurust ühele sõnale, et vältida liigset sõnakordust. Sobivaid tegusõnu leitakse kahe reegli abil:

1. Kui pronoomenile eelneb kirjavähemärk, millele eelneb nimisõna või pärisnimi, siis asendatakse asesõna selle nimisõna või pärisnimega. See tuleneb relatiivlause paiknemisest harilikult selle sõna järel, mille kohta ta käib.
2. Kui asesõna ülem on tegusõna, mille ülem on nimisõna või pärisnimi, siis saab asesõna asendada selle nimisõna või pärisnimega juhul, kui tegusõna alluvate hulgas pole nimisõnana subjekti. Kui tegusõna alluvate hulgas on nimisõnana subjekt, kustutakse asesõna ära.

Asesõna asendamise jaoks võetakse asenduse sõnavormist arv ja asesõna vormist kääne ning teostatakse süntees, et saada korrektne sõnavorm asendusele.

Järgmisena eraldatakse iseseisvad laused ja paigutatakse need subjekt-verb-objekt malli. Iga iseseisva lause loomist alustatakse ühest lause peasõnast, mis analüüsiga kindlaks määrati. Malli paigutamist teostatakse rekursiivselt sõna haaval, iga sõna otsestele alluvatele rakendatakse reegleid, mis määravad, kas alluv paikneb enne või pärast sõna. Kuna peasõnad võivad alluda teistele peasõnadele ja võivad seega kasutada sõltuvustabelis olevaid samu sõnu, siis peame need üksteisest eraldama. Seetõttu saab alluv olla seoses oma ülemusega vaid siis, kui vastab tõele üks reeglitest:

1. ülemuse liigiks on tegusõna;
2. ülemus on märgendatud lause peasõnaks;
3. alluva liigiks ei ole tegusõna;
4. kui alluva liigiks on tegusõna, siis peab ta olema ülemuse atribuut.

Need reeglid tagavad, et kui ülemus ei ole tegusõna ega peasõna, siis tema alluv ei ole tegusõna, välja arvatud juhul, kus alluv on ülemuse otsene atribuut.

Seejärel otsustab lihtsustaja, kummal pool sõna peaks alluv olema. Alluva asukoha otsustamiseks on mitmeid reegleid, mis jagunevad selle järgi, kummale poole ülemust alluv paigutatakse. Selleks, et alluv paigutataks ülemuse ette, peab paika pidama vähemalt üks eesreeglitest ning mitte ükski järelreeglitest, vastasel juhul pannakse alluv ülemuse järgi.

Eraldi juhtumina käsitletakse sidesõnu, mis pannakse eelnevate sõnade listi algusesse v.a juhul, kui sidesõna ülemus käib enda ülemuse ette ning sidesõna indeks on suurem kui tema ülemuse ülema indeks. See tagab näiteks lauseosa *mees ja naine* puhul, et sidend *ja* jääks oma kohale. Sidendi asukoht võib muutuda sest *ja* viitab järgnevale sõnale *naine*, mis omakorda viitab sõnale *mees* ning kui esineb olukord, kus *naine* peaks olema enne sõna *mees*, siis naiivse tulemuse *ja naine mees* asemel saame korrektse (kuid ümberpööratud) tulemuse *naine ja mees*.

Sõna eesreeglid on:

1. alluva silt määrab üheselt ära, et alluv on ülema ees;
2. alluv on subjekt, määrus, verbi negatiiv või sõna “olema” finiitses vormis.

Sõna järelreeglid on:

1. alluva silt määrab üheselt, et alluv on ülema järel;
2. sõna on verb ning alluv on määrus, objekt või predikatiiv;
3. sõna ja alluva silt on sama ning sõna silt on subjekt, predikatiiv, määrus või objekt.

Pärast reeglite järgi sõnade jaotust moodustatakse terviklik lause, mis lisatakse tagastamiseks mõeldud kõikide lausete sõnesse.

2.2.5 API

APIks on veebileht, mis on lihtsustajale ligipääsuks. Selle peamiseks funktsiooniks on ühendada eesrakendus tagarakendusega, mille jaoks võtab API URLi parameetri *l*, kustutab ära sisendist ülakomad lihtsustajale edasiandmise hõlpsustamiseks, käivitab lihtsustaja andes talle ette sisendi väärtuse ning tagastab saadud tulemuse. Ülakomade eemaldamine väldib üleliigset sõnestamist (joonis 16) lihtsustajas.

```
print(Text("Google'i uus logo.").word_texts)

> ['Google', "'", 'i', 'uus', 'logo', '.']
```

Joonis 16. Ülakoma esinemine sisendis tekitab liigset sõnestamist.

Lihtsustaja brauserilaienduseta kasutamiseks ja testimiseks on loodud lihtne API eesrakendus[7] (joonis 17), mis võimaldab kasutada lihtsustajat kirjutades sisendi veebilehel olevasse tekstikasti. Samuti on võimalik kasutajal näha lihtsustamise käigus kogutud lisainformatsiooni kui lubada *debug*.

Joonis 17. API eesrakenduse kasutamine.

Testimise ja hindamise eesmärgil on lisatud ka juhusliku lause lihtsustamine (joonis 18), mis võtab suvalise lause Tartu Ülikooli arvutilingvistika uurimisrühma Tasakaalus korpusest [44]. Selle kasutamiseks tuleb URL parameetri *random* väärtuseks anda *on*[8].

```
_________________________
Esialgne lause
 AO tõendab elementaarset arvuti kasutamise oskust ja on põhimõtteliselt eluaegne , nagu ka näiteks autojuhiluba .
-------------------------
Lihtsustatud lause
 Ao tõendab elementaarset arvuti kasutamise oskust. Nagu ka näiteks autojuhiluba on põhimõtteliselt eluaegne.
_________________________
16. lause, mida vaadati.
```

Joonis 18. Juhusliku korpusest võetud lause lihtsustamine.

Sobiv juhuslik lause kuvatakse esimesena, millele järgneb selle lihtsustatud versioon. Lisaks näidatakse, mitmendana see lause programmile ette anti. Juhusliku lause lihtsustamise on ajakulukas: kuna korpus on muutmata kujul, siis on võimalik, et lihtsustaja saab mitu lihtsustamatut sisendit.

2.2.6 Testija

Testija on tagarakenduse osa, mida kasutati lihtsustaja vahepealseks ning lõplikuks hindamiseks ning testimiseks. Selle jaoks kasutati samuti Tasakaalus korpust [44], millest võeti iga katse sisendiks umbes 20 tuhat lauset. Iga sisendlausega tehti päring APIle ning eraldati

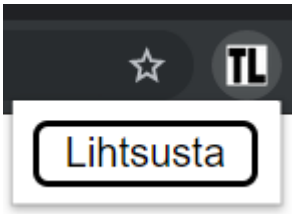
[7] http://prog.keeleressursid.ee/ss_syntax/

[8] http://prog.keeleressursid.ee/ss_syntax/?random=on

vastusest lihtsustatud lause ning informatsioon, mis statistika jaoks koguti kokku. Statistikana koguti edukate ning erinevatel põhjustel lõpetatud lihtsustamiste arve. Lisaks loendati ja koguti kokku saadud veateated, mis pika testimise käigus tekkisid.

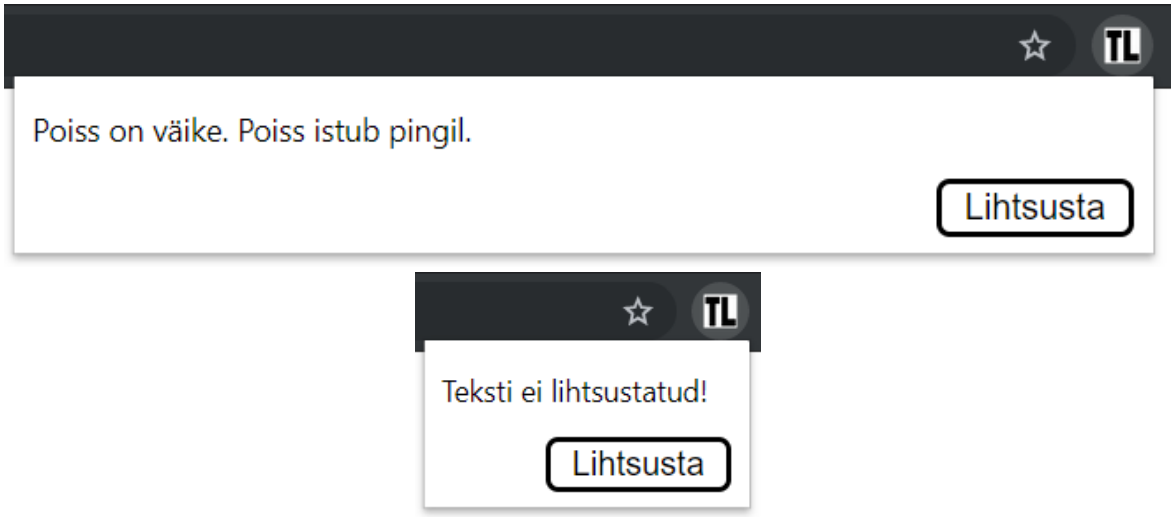
2.3 Eesrakendus

Programmi eesrakendus[9] on loodud veebibrauseri Google Chrome'i laiendusena, mille eesmärk on pakkuda teksti lihtsustajale hõlpsat ligipääsu. Laiendus on koostatud, tuginedes Google'i poolt koostatud õpetustele [43] ning veebilehe W3Schools juhenditele [45]. Valminud laienduse töö toimub peamiselt sellega kaasnevas hüpikaknas (joonis 19). Veebilehel markeeritud tekstile ligipääsemiseks on lisaks loodud skript, millega hüpikaken suhtleb Chrome'i sõnumiedastussüsteemi[10] kasutades.



Joonis 19. Laienduse hüpikaken

Kasutaja saab külastataval veebilehel markeerida teksti, vajutada laienduse nupule *Lihtsusta* ning mõne aja möödudes kuvatakse kasutajale lihtsustamise edukusest olenevalt kas teksti lihtsustatud versioon või teade, et teksti ei lihtsustatud (joonis 20).



Joonis 20. Üleval lihtsustatud teksti kuvamine, all teade teksti mittelihtsustamise kohta.

[9] https://github.com/StiivoSiider/TekstiLihtsustaja-Siider/tree/master/lihtsustaja-chrome-extension

[10] https://developer.chrome.com/apps/messaging

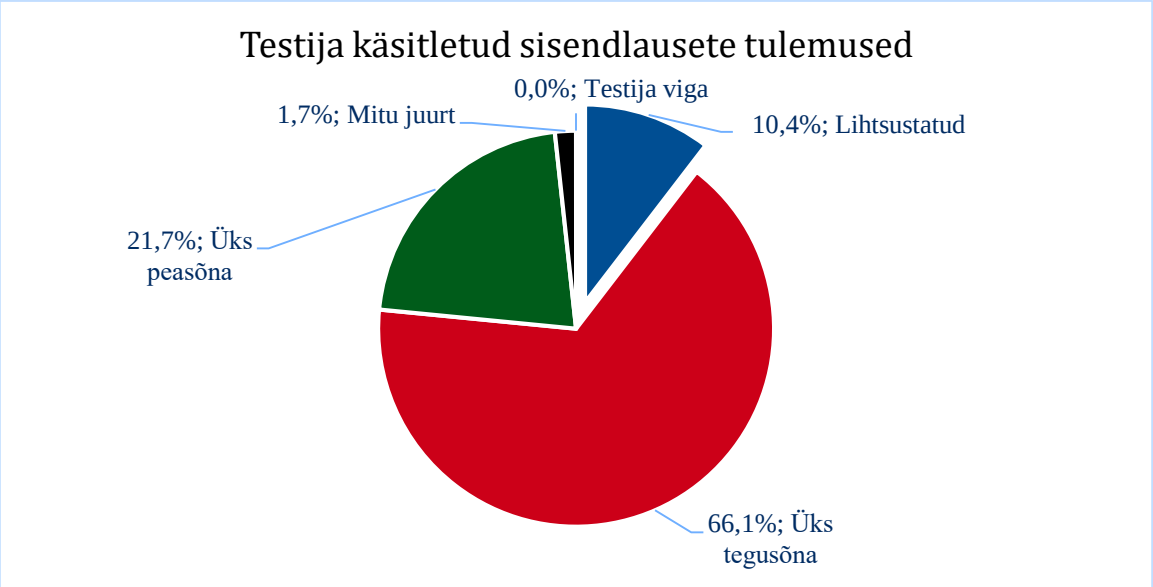
Lihtsustaja tööaega ning kasutusmugavust arvestades on lisatud CSSis loodud animatsioon[11], mis annab kasutajale märku, et lihtsustaja praegu töötab ning tulemuse saamiseks peab ootama.

2.4 Hindamine

Lihtsustaja hindamiseks kasutati küsimustikku ning varem kirjeldatud testijat. Testijaga hindamise eesmärgiks oli teha statistikat lihtsustamise edukusest ning lõpetamise põhjustest. Küsimustiku eesmärgiks oli hinnata lihtsustatud lauseid, kasutades kriteeriume, mis on välja pakutud muudetud tekstide hindamiseks [37, 38, 6, 46, 10]. Neist valiti küsimustikku eelistatus, loetavus, mõistetavus, grammatiline korrektsus ning mõtte säilivus.

2.4.1 Automaatne statistiline hindamine

Automaatseks hindamiseks kasutatava testija tööks võeti Tasakaalus korpusest [44] suvaliselt kümme korpusefaili nii, et sisendlauseid oleks umbes 20 tuhat. Testija tulemuste (vt lisa 2) statistika on esitatud joonisel 21. Statistikas erineb sisendlausete koguarv erinevate juhtude summast, kuna kogutud koguarv loeb korpusest saadud lauseid, kuid lihtsustatud lausel võib olla jutumärkide või sulgude vahel olevaid rekursiivseid lihtsustusi. Iga sisendlause kohta loetakse kokku maksimaalselt ühe rekursiivse lihtsustuse peatumise põhjus.



Joonis 21. Testija käsitletud sisendlausete tulemused

[11] https://www.w3schools.com/howto/howto_css_loader.asp

Statistika kohaselt lihtsustati 10,4% lausetest. Peamiseks lihtsustamise peatamise põhjuseks oli ühe tegusõnaga laused, mis moodustasid 66,1% sisendlausetest. Teisi põhjuseid, üks peasõna või mitu juursõna, esines vastavalt 21,7 ja 1,7 protsendil juhtudest. Esines ka üks testija viga, mille põhjustas ebaõnnestunud ühendumine serveriga.

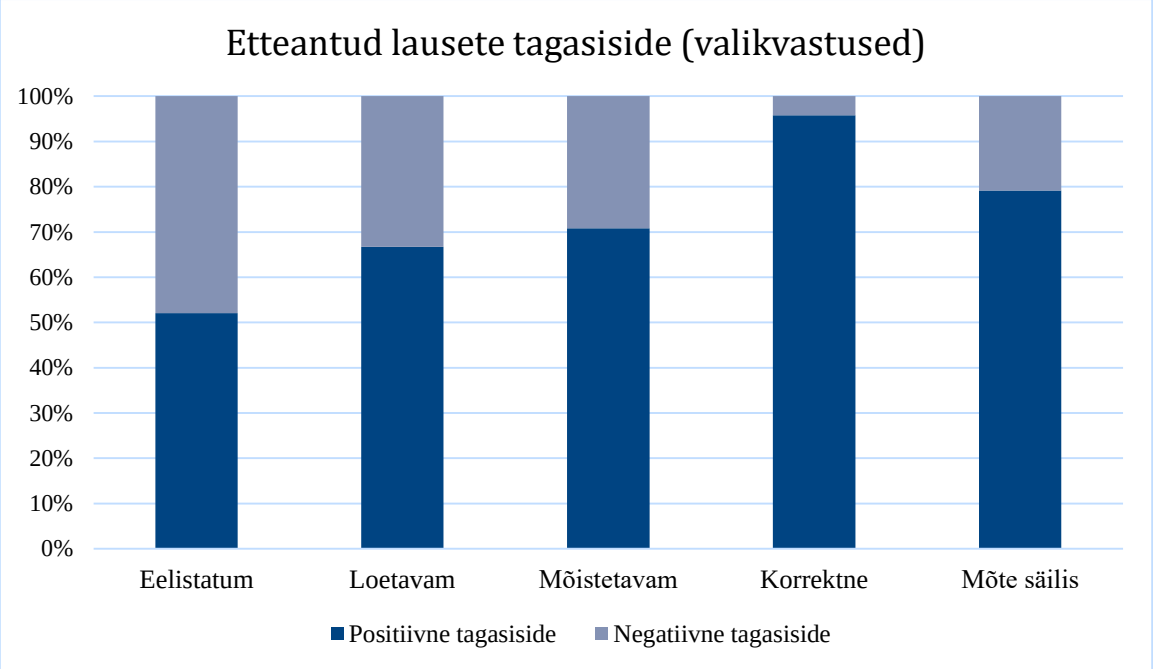
Nendest protsentidest saab järeldada seda, et lihtsustaja rakendatus on väga väike, kuna sagedaseid, ühe tegu- või peasõnaga lauseid ei lihtsustata. Lihtsustatud lausete osakaalu suurendamiseks saaks rakendada lihtsustamismeetodeid, mis ei nõua liitlauset, näiteks lisandi või lihtlauses esineva rinnastusseose lihtsustamine.

2.4.2 Arvamuspõhine hidamine

Arvamuspõhiseks hindamiseks kasutatav küsimustik (ülesehitus lisas 3) koosneb neljast osast. Esimesena küsitakse vajalikku üldinformatsiooni: tekstilihtsustamise kontekstis on olulised näiteks vastaja emakeel ning keelelisi oskusi mõjutavad häired nagu näiteks düsleksia. Seejärel antakse hindamiseks kolm lauset, mis demonstreerivad lihtsustaja võimalusi: relatiivlause, rinnastusseoses liitlause ning sõnajärjestuse lihtsustamine. Kolmandaks antakse (kasutades API lisavõimalusi) vastajale hindamiseks vähemalt viis juhuslikku lauset ning selle lihtsustust. Viimaseks on võimalus vastajatel ise lauseid sisestada ning saadud lihtsustusi hinnata.

Iga lihtsustuse hindamiseks (küsimused lisas 4) võrreldakse esialgset ning lihtsustatud lauset ning valitakse parem kolmes kategoorias: eelistatus, loetavus ning mõistetavus. Seejärel hinnatakse lihtsustatud lause loetavust ja mõistetavust viiepalliskaalal. Lisaks vaadeldakse binaarsete küsimustena grammatilist korrektsust ning mõtte säilivust. Iga individuaalse lihtsustuse ning kogu küsimustiku lõpus küsiti ka vabas vormis valikulist tagasisidet.

Koostatud küsimustikule oli 16 vastajat, kes kõik olid häireteta ning kõigi emakeeleks oli eesti keel. Hinnangud agregeeriti kahte gruppi: etteantud laused ja muud (juhuslikud ning vastaja poolt sisestatud) laused. Mõlema grupi korral käsitletakse eraldi valikvastustega ning viiepalliskaalal hindamise küsimusi. Kõigepealt tuuakse välja etteantud laused.

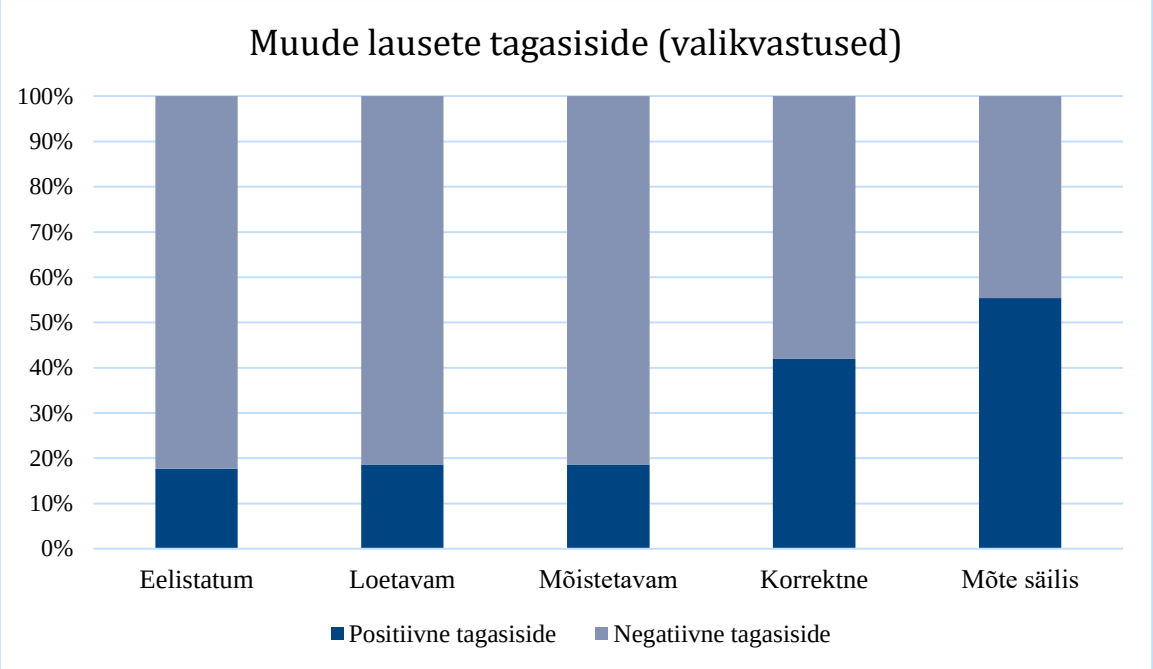


Joonis 22. Etteantud lausete valikvastuste tagasiside (valikvastused)

Joonisel 22 toodud tulemuste põhjal on etteantud lauseid hinnatud peamiselt positiivselt. Lihtsustatuid lauseid valiti eelistatuse, loetavuse ning mõistetavuse kategooriates rohkem kui etteantud lauseid. Loetavuse ja mõistetavause suurem protsent võrreldes eelistatusega tuleneb ilmselt vastajate kõrgetest keelelistest oskustest, mistõttu pole keerulisemad lausestruktuurid nende jaoks probleemiks, vaid on hoopis harjumuseks.

Loetavuse ja mõistetavuse viiepalliskaalal said etteantud laused samuti häid tulemusi – vastavalt 4,31 ja 4,75.

Etteantud lausete edukuseks võib peamiselt siiski pidada nende kunstlikkust. Tegemist on valitud lausetega, mis annavad mõistliku tulemuse, et demonstreerida toimivaid lihtsustamisvõimalusi. Seega ei saa nende tulemuste põhjal adekvaatselt hinnata lihtsustaja korrektsust. Seetõttu oli ka küsimustiku põhiosaks juhuslikud laused.



Joonis 23. Muude lausete tagasiside (valikvastused)

Joonisel 23 esitletud juhuslike ning vastaja sisestatud lausete tagasiside oli üldiselt negatiivne. Esialgseid lauseid valiti igas kategoorias üle 80% juhtudest, mis näitab, et lihtsustaja ei ole võimeline esialgseid sisendeid parandama. Samuti on madalalt hinnatud grammatilist korrektsust ning mõtte säilivust, mis viitavad vastavalt probleemidele lausemallis, sest sellega pannakse laused uuesti kokku, ning probleemidele transformatsioonis. Lihtsustatud lausete loetavust ja mõistetavust hinnati viiepalliskaalal vaevu üle keskmise, vastavalt 3,26 ja 3,22.

Juhuslike lausete lihtsustamisel loendati, mitmes lihtustajale antud sisend sai lihtsustatud, ning selle tulemus 11% on lähedane testija saadud osakaalule.

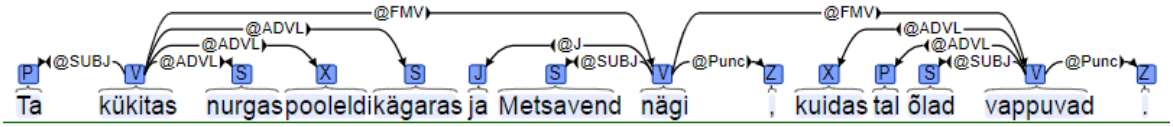
Küsitluse tulemusena ilmnesid mitmed probleemid lihtsustaja töös, mida järgnevalt käsitletakse.

2.5 Probleemid

Tuginedes jooksvalt rakendatud testija tulemustele ning küsimustikule, tuuakse siin välja suuremad probleemid nii lihtsustajas endas kui ka kasutatavas EstNLTK süntaksi analüsaatoris.

2.5.1 Sõnapõhise lausemalli rakendamine

Praegune lausemall, mida kasutatakse lihtsustatud lausete tagasi kokkupanekul ning subjekt-verb-objekti kujule viimiseks, moodustab väga tihti ebamääraseid ja vigaseid lauseid. Põhjuseks on peamiselt lauses esinevad määrused, mille kohta süntaksianalüsaator annab vähe infot. Samuti on mall ise lihtne: sõnu järjestatakse põhiliselt nende süntaktiliste funktsioonide järgi.



Joonis 24. Sõltuvusstruktuur määrustega [23]

Joonisel 24 on näha lausemalli jaoks keeruline lause, kuna siin esineb adverbiaal ehk määrus nii tegusõna ees kui ka järel, seega on hilisem otsustamine, kummal pool tegusõna need peaksid asuma, raskendatud.

Selle probleemi lahendamiseks pole sobivat võimalust leitud. Võimalik, et siin oleks abi sõnade esialgsest paiknemisest, mille põhjal määruseid järjestada, kuid samuti leidub ilmselt ka siis lauseid, mis sellisesse malli ei sobiks. Probleemi leevendamiseks võib siin kasu olla Shardlow'i [15] pakutud ülemmärgendamisest, sest sel juhul ei ole vaja iga üksiku sõna paiknevust käsitleda.

2.5.2 Asesõna viitab mitmele sõnale

Lauses, kus asesõna viitab lauses mitmele tegijale, nt paarile *Poiss ja tüdruk*, võib muutuda mõte. Selle põhjuseks on lihtsustaja piirang asendada asesõna ainult ühe sõnaga, et vältida omadussõnade kordusi. Näiteks on joonisel 25 näha, et asesõna *kes* asendati ainult sõnaga *tüdrukud*, mitte terve eelneva konstruktsiooniga, mis oleks lugejale loomulikum tõlgendusviis.

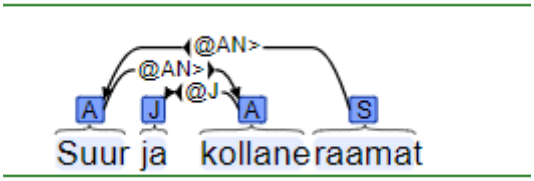
```
Poisid ja tüdrukud, kes jäid tundi hiljaks, said karistada.
```

```
Poisid ja tüdrukud said karistada. Tüdrukud jäid tundi hiljaks.
```

Joonis 25. Asesõna vigane asendus

Selle lahendamiseks oleks võimalik implementeerida Siddharthani [17] regeneratsiooni etapis tehtavat tagasiviitavate väljendite genereerimist, kus tekstis olevatele asesõnadele luuakse sobiv asendus. See täidaks varem seatud nõuet limiteerida asesõna asendamisel liigset teksti kordamist ning võimaldaks asendusel olla pikem kui üks sõna.

Teiseks võimaluseks oleks võtta arvesse sidendiga eraldatud sõnapaaride ülesehitust, mis esineb sõltuvuspuus kindla struktuurina (joonis 26). Keerulisemate sõnapaaride puhul võib selline lahendus eksida, mistõttu vajaks see enne rakendamist põhjalikku katsetamist.



Joonis 26. Sõltuvuspuu sidendiga eraldatud sõnapaarist [23]

Kolmandaks võimaluseks oleks ka siin kasutada ülemmärgendamist, millega saaks võtta *Poisid ja tüdrukud* kokku üheks märgendiks, mida saaks tervikuna asendamisel kasutada. See nõuaks siiski keerulisemat struktuuri, sest kui sõnal *Poisid* või *tüdrukud* oleks kaasas omadussõna, ei sobiks enam terviklik asendus.

2.5.3 Sidendiga eraldatud sõnapaari reformeerimine

Üheks märgatavaks probleemiks lause kokkupanemisel, mida kirjeldati ka programmi transformatsiooni etapi ülevaate juures, olid sidendiga eraldatud sõnapaarid. Võtame selle näiteks fraasi *mees ja naine*. Fraasi kokkupanekul võis tekkida olukordi, milles paari teine pool oli süntaktilise funktsiooni kohaselt esimese poole eelatribuut (nt joonis 26). Sellise olukorra tulemuseks oli vigane sõnapaar kujul *ja naine mees*. Probleemi lahendamiseks paigutati vaikimisi sõnast eespool paiknev sidend sellises olukorras sõna järgi, mis annaks tulemuseks *naine ja mees*. Saadud tulemus pole siiski korrektne, kuna sõnapaar on ümber pööratud.

Jäädes praeguse, sõnapõhiselt töötava lihtsustaja juurde, saab seda probleemi lahendada sõltuvuspuus kõrgemal, sõnapaari ülemust käsitledes. See on ajakulukas, kuna iga sobiva alluva korral peame rekursiivselt vaatlema tolle alluvaid. Leides alluva alluvate hulgast sobiva saame määrata korrektse sõnade järjestuse. Selle lahenduse positiivseks küljeks on see, et sidendeid saab järjekordselt paigutada alati sõna ette.

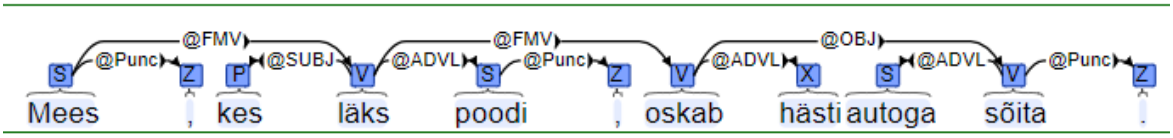
Teiseks võimaluseks oleks rakendada ülemmärgendust, mis käsitleks sellist fraasi ühtsena ning seega ei peaks muretsema nende järjestuse pärast.

2.5.4 Peasõnade eraldamine

Küsimustiku tulemusel selgus peasõnade eraldamise ülesehituses kolm probleemi: juurena esinevad subjektid moodustasid ühesõnalisi lauseid, komaga eraldatud peasõnu ei eraldatud ning asesõna kustutati ära juhul, kui see oli subjekt.

Juured, mis pole tegusõnad, olid probleemiks kuna pole kindel, mis süntaktilist funktsiooni need täpsemalt täidavad. Näiteks joonisel 27 olevas lauses on sõna *Mees* lause juur ning eraldades peasõnu lõigati ära selle ainus alluv, tegusõna *läks*, sest nende vahel on olemas asesõna *kes*, mis on ka subjektiks. Selle parandamiseks lisati peasõnade eraldamisel juurde kontroll, et peasõna ei eraldata tema ülemast juhul, kui ülem on juursõna ning pole tegusõna.

Teiseks probleemiks olid komaga eraldatud peasõnad, näiteks joonisel kujutatud tegusõnad *läks* ja *oskab*, mille vahelist sõltuvust ei lõhuta. Kuna nende vahel paiknev koma ei ole otseselt seotud kummagi tegusõnaga, on selliste olukorda identifitseerimine palju keerulisem kui asesõnade ja sidendite puhul. Selle lahendamiseks peaks lihtsustaja tegema kindlaks, et kahe tegusõna vahel olev kirjavahemärk on neid piisavalt eraldav. Praegune implementatsioon seda teha ei suuda ning näiteks toodud lause on seetõttu lihtsustaja jaoks lihtsustamatu.



Joonis 27. Lause, mille juureks on nimisõna [23]

Joonisel 27 olevas lauses ilmnes ka probleem asesõnadega, kus subjektina esinev asesõna *kes* lausest ära kustutati, jättes tegusõna *läks* subjektita. Selle parandamiseks lisati kontroll enne kaht peasõna eraldava sidendi või asesõna kustutamist, et kustutatav poleks tegusõna subjekt.

2.5.5 Süntaksianalüüsi tulemusena mitu juurt lauses

Mõningate lausete (nt. *Pingil istub poiss, kes on väike ja väsinud.*) puhul leidis kasutatav süntaksianalüsaator ühest lausest mitu juurt, mis näitab, et tegemist peaks olema mitme lausega. Nagu joonistel 28 ja 29 näha, siis tundub, et tegemist on lokaalse probleemiga, sest kasutades mõnda teist MaltParseriga süntaksianalüsaatorit [23] on tulemus erinev.

```
1       Pingil  pink    S       S       sg|ad   2       @ADVL   _       _
2       istub   istu    V       V       b       0       ROOT    _       _
3       poiss   poiss   S       S       sg|n    2       @SUBJ   _       _
4       ,       ,       Z       Z       _       3       xxx     _       _
5       kes     kes     P       P       pl|n    0       ROOT    _       _
6       on      ole     V       V       b       0       ROOT    _       _
7       väike   väike   A       A       sg|n    6       @PRD    _       _
8       ja      ja      J       J       _       9       @J      _       _
9       väsinud väsi    V       A_V     nud     6       @IMV    _       _
10      .       .       Z       Z       _       9       xxx     _       _
```

Joonis 28. Mitme juurega MaltParseri väljund

```
1  Pingil   pink     S S  sg|ad                2 @ADVL _ _
2  istub    istu     V V  indic|pres|ps3|sg   0 ROOT  _ _
3  poiss    poiss    S S  sg|nom               2 @SUBJ _ _
4  ,        ,        Z Z  Com                  3 @Punc _ _
5  kes      kes      P P  intrel|sg|nom        6 @SUBJ _ _
6  on       ole      V V  indic|pres|ps3|sg   3 @FMV  _ _
7  väike    väike    A A  sg|nom               6 @PRD  _ _
8  ja       ja       J Jc _                    9 @J    _ _
9  väsinud  väsinud  A A  sg|nom               7 @PRD  _ _
10 .        .        Z Z  Fst                  9 @Punc _ _
```

Joonis 29. Korrektne MaltParseri väljund [23]

Probleemi vältimiseks jätab lihtsustaja vahele kõik laused, milles on mitu juurt.

2.6 Edasiarendamisvõimalused

Programmis olevate probleemidele ning kehvadele hindamistulemustele tuginedes vajab lihtsustaja põhjalikku edasiarendust. Peamisteks puudusteks on ülemmärgendamine, regeneratsioon ning samuti väljundi kirjavahemärgistamine.

Ülemmärgendamine aitaks lihtsustada teksti transformeerimist ning lihtsustaks väga spetsiifiliselt kirjutatud sõnapõhiseid reegleid, kuna arvestama peaks suurema pildiga. See peaks olema eraldi sammuna pärast süntaksianalüüsi, sest EstNLTK analüsaator leiab sõltuvusi korrektselt vaid üksikute sõnade tasemel sõnestatud sisendiga. Seega oleks vajalik luua süsteem, mis suudab võtta saadud sõltuvuspuu, grupeerida selles esinevad sõnad ning moodustada saadud gruppidega uus sõltuvuspuu. Lisaks muudab see oluliselt lihtsamaks lausete kokkupaneku, sest kasutatavaid elemente on vähem.

Regeneratsioon, Siddharthani [17] välja toodud ning tugevalt rõhutatud süntaktilise lihtsustamise samm, on vajalik, kuna see muudab lihtsustatud teksti sidusamaks ning loetavamaks. Samuti parandaks loetavust korrektne väljundi kirjavahemärgistamine, mida praegune lihtsustaja ei tee. Taastades korrektsed kirjavähemärgid muutub tekst ka struktureeritumaks.

Lihtsustatavate lausete hulga suurendamiseks tuleks praegustele operatsioonidele lisaks implementeerida lisandi ning rinnastusseoses olevate lihtlausete lihtsustamist. Need täiendused liiguvad eemale praegusest mitmele peasõnale tuginevast süsteemist, mille asemel on vaja peasõnu hakata juurde looma. See vähendab ühe tegu- või peasõna pärast lõppenud lihtsustuste arvu, sest mõlemad konstruktsioonid võivad esineda lauses, kus on ainult üks tegu- või peasõna. Lisaks annaks see võimaluse viia rohkem lauseid subjekt-verb-objekt kujule, sest käesoleval juhul kasutatakse seda ainult pärast teisi lihtsustamisoperatsioone.

Süntaksianalüsaatori poole pealt on võimalik katsetada reeglipõhist VISLCG3 parserit[12], mis võib, kuid ei pruugi anda andmepõhise MaltParseriga võrreldes paremaid tulemusi.

Teksti lihtsustamise kontekstis on võimalik praegusele süsteemile lisada leksikaalne lihtsustaja või Longi ja Rossi [11] välja pakutud keeruliste sõnade selgitaja.

Väljapakutud edasiarendused viiksid süntaktilise lihtsustaja tänapäevaste võõrkeelsete lihtsustajate tasemele lähemale, andes eeldatavalt paremaid tulemusi.

[12] https://estnltk.github.io/estnltk/1.4.1/tutorials/dependency_syntax.html#vislcg3-based-syntactic-analysis

3 Kokkuvõte

Käesolevas töös uuriti erinevaid teksti lihtsustamise meetodeid, millest enim pandi rõhku süntaktilisele lihtsustamisele, ning toodi välja kaks süntaktilist lihtsustamist pakkuvat veebirakendust. Esitletud teksti lihtsustamise meetodid näitavad erinevaid võimalusi ning lahendusi, kuidas muuta tekst lugejale paremaks. Põhjalikumalt tutvustati inglise keele süntaktilise lihtsustamise teooriat, et rakendada seda eesti keele süntaksi lihtsustavale programmile.

Töö praktilises osas valmis teksti lihtsustaja, sellele ligipääsu andev API ning veebibrauseri laiendusena eesrakendus. Loodud lihtsustaja jagati kahte etappi: analüüs ja transformatsioon. Analüüsi etapis uuriti sisendi morfoloogiat ja süntaksi kasutades EstNLTK paketti, ning hinnati leitu põhjal sisendlause sobivust lihtsustamiseks. Transformatsiooni etapis lihtsustati relatiivlauset ning rinnastusseoses olevaid liitlauseid. Lisaks viidi lihtsustatud laused subjekt-verb-objekt lausekujule.

Loodud üksikute sõnade tasemel töötava lihtsustaja hindamise tulemus oli negatiivne. Juhuslikult etteantud lauseid pidasid vastajad paremaks vähem kui 20% juhtudest, kuna vähenes lause loetavus, mõistetavus, grammatiline korrektsus ning sageli kadus ka lause mõte. Lisaks oli lihtsustatud lausete osakaal madal – umbes 10%. Saadud tulemustest võib järeldada, et süntaktiline lihtsustamine ja lausete moodustamine on üksikute sõnade tasemel liialt keerukas ning raskesti teostatav ning keskenduma peaks üldistatud struktuuridele, näiteks fraasidele.

Võimalikeks edasiarendusteks on ülemmärgendite kasutamine, mis grupeerib üksikuid sõnu mugavamaks käsitlemiseks, ning regeneratsiooni etapi loomine, mis parandab teksti sidusust.

Viidatud kirjandus

[1] Haridus- ja Teadusministeerium. Eesti keel ja võõrkeeled. 2018. https://www.hm.ee/et/tegevused/eesti-keel-ja-voorkeeled (13.01.2019)

[2] Kivistik K. Keelteoskus, keelte kasutamine, kontaktid ja keeltega seotud hoiakud. Kultuuriministeerium. Eesti ühiskonna integratsiooni monitooring. 2017, lk. 53.

[3] EstNLTK - eesti keele töötluseks loodud teekide kogumik. https://estnltk.github.io/ (3.05.2019)

[4] Sulem E., Abend O., Rappoport A. Simple and Effective Text Simplification Using Semantic and Neural Methods. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, vol. 1, pp. 162–173.

[5] Chandrasekar R., Doran C., Srinivas B. Motivations and methods for text simplification. *Proceedings of the 16th conference on Computational linguistics*, 1996, vol. 2, pp. 1041–1044.

[6] Rello L., Baeza-Yates R., Bott S., Saggion H. Simplify or help?: text simplification strategies for people with dyslexia. *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, 2013, nr. 15.

[7] Crossley S. A., Allen D., McNamara D. S. Text simplification and comprehensible input: A case for an intuitive approach. *Language Teaching Research*, 2012, vol. 16, nr. 1, pp. 89–108.

[8] Štajner S., Saggion H. Data-Driven Text Simplification. *Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts*, 2018, pp. 19–23.

[9] Mason J., Kendall J. Facilitating reading comprehension through text structure manipulation. *Alberta Journal of Medical Psychology*, 1979, vol. 25, nr. 2, pp. 68–76.

[10] Siddharthan A. A survey of research on text simplification. *International Journal of Applied Linguistics*, 2014, vol. 165, nr. 2, pp. 259–298.

[11] Long M. H., Ross S. Modifications That Preserve Language and Content. *Simplification: Theory and Application*, 1993, pp. 29–52.

[12] Oh S.-Y. Two types of input modification and EFL reading comprehension: Simplification versus elaboration. *TESOL Quarterly*, 2001, vol. 35, nr. 1, pp. 69–96.

[13] Paetzold G. H., Specia L. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 2017, vol. 60, pp. 549–593.

[14] Laufer B. What percentage of text-lexis is essential for comprehension? *Special Language: From Humans Thinking To Thinking Machines*, 1989, pp. 316–323.

[15] Shardlow M. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*, 2014, pp. 58–70.

[16] Peedosk M. Eesti keele digitaalsete ressursside ja tehnoloogiate rakendamine teksti lihtsustamise programmis. TÜ arvutiteaduse instituudi bakalaureusetöö. 2017.

[17] Siddharthan A. Syntactic Simplification and Text Cohesion. *Research on Language and Computation*, 2006, vol. 4, nr. 1, pp. 77–109.

[18] Morphology and syntax. https://www.uni-due.de/SHE/REV_MorphologySyntax.htm (14.01.2019)

[19] Müürisep K. Eesti keele arvutigrammatika: süntaks. *Dissertationes Mathematicae Universitatis Tartuensis 22*, 2000.

[20] Sirts K. Syntactic Parsing. 2017. https://courses.cs.ut.ee/LTAT.01.001/2017_fall/uploads/Main/Lecture7.pdf (4.05.2019)

[21] EstNLTK - Dependency syntactic analysis. https://estnltk.github.io/estnltk/1.4.1/tutorials/dependency_syntax.html (4.05.2019)

[22] Muischnek K., Müürisep K. Eesti keele sõltuvuspuude pank ja selle keeleteoreetilised lähted. *Emakeele Seltsi aastaraamat*, 2016, lk. 122–145.

[23] Tartu Ülikool. Süntaksianalüsaator. 2015. https://korpused.keeleressursid.ee/syntaks (7.05.2019)

[24] Hall J., Nilsson J., Nivre J. MaltParser. http://www.maltparser.org/ (3.05.2019)

[25] Didriksen T., Bick E. VISL CG-3 Development Information. https://visl.sdu.dk/cg3.html

[26] Süntaksianalüsaatori väljundi selgitus. https://korpused.keeleressursid.ee/syntaks/dokumendid/syntaksiliides_ee.pdf (4.05.2019)

[27] Erelt M. Lisand. http://keeleabi.eki.ee/artiklid2/lisand.html (3.05.2019)

[28] Erelt M., Erelt T., Ross K. Eesti keele käsiraamat. Tallinn: Eesti Keele Sihtasutus. 2007.

[29] Aluísio S. M., Specia L., Pardo T. A., Maziero E. G., & Fortes R. P. Towards Brazilian Portuguese automatic text simplification systems. *Proceedings of the Eighth ACM Symposium on Document Engineering*, 2008, pp. 240–248.

[30] González J. A finite-state approach to phrase-based statistical machine translation. *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012.

[31] Lopez A. A Survey of Statistical Machine Translation. Technical Report 2006-47. University of Maryland Institute for Advanced Computer Studies. 2007.

[32] Zens R., Och F. J., Ney H. Phrase-Based Statistical Machine Translation. *KI 2002: Advances in Artificial Intelligence: 25th Annual German Conference on AI*, 2002, pp. 18–32.

[33] Zhang Y., Ye Z., Feng Y., Zhao D., Yan R. A Constrained Sequence-to-Sequence Neural Model for Sentence Simplification. 2017.

[34] Nisioi S., Štajner S., Ponzetto S. P., Dinu L. P. Exploring Neural Text Simplification Models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, vol. 2, pp. 85–91.

[35] Cho K., Merrienboer B.v., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[36] Sutskever I., Vinyals O., Le Q. V. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems 27*, 2014, pp. 3104–3112.

[37] Carroll J., Minnen G., Canning Y., Devlin S., Tait J. Practical simplification of English newspaper text to assist aphasic readers. *Proceedings of AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, 1998.

[38] Ferrés D., Marimon M., Saggion H., AbuRa'ed A. YATS: Yet Another Text Simplifier. *International Conference on Applications of Natural Language to Information Systems*, 2016.

[39] Scarton C., Aprosio A. P., Tonelli S., Wanton T. M., Specia L. MUSST: A Multilingual Syntactic Simplification Tool. *Proceedings of the IJCNLP 2017, System Demonstrations*, 2017.

[40] TALN-UPF. YATS Demo. 2016. http://able2include.taln.upf.edu/ (1.05.2019)

[41] Simpatico Authoring Tool. http://dh-server.fbk.eu:19003/simp-engines/tae/webdemo/index.html (4.05.2019)

[42] e-Teatmik: IT ja sidetehnika seletav sõnaraamat. http://www.vallaste.ee/

[43] Google Chrome. What are extensions? https://developer.chrome.com/extensions (3.05.2019)

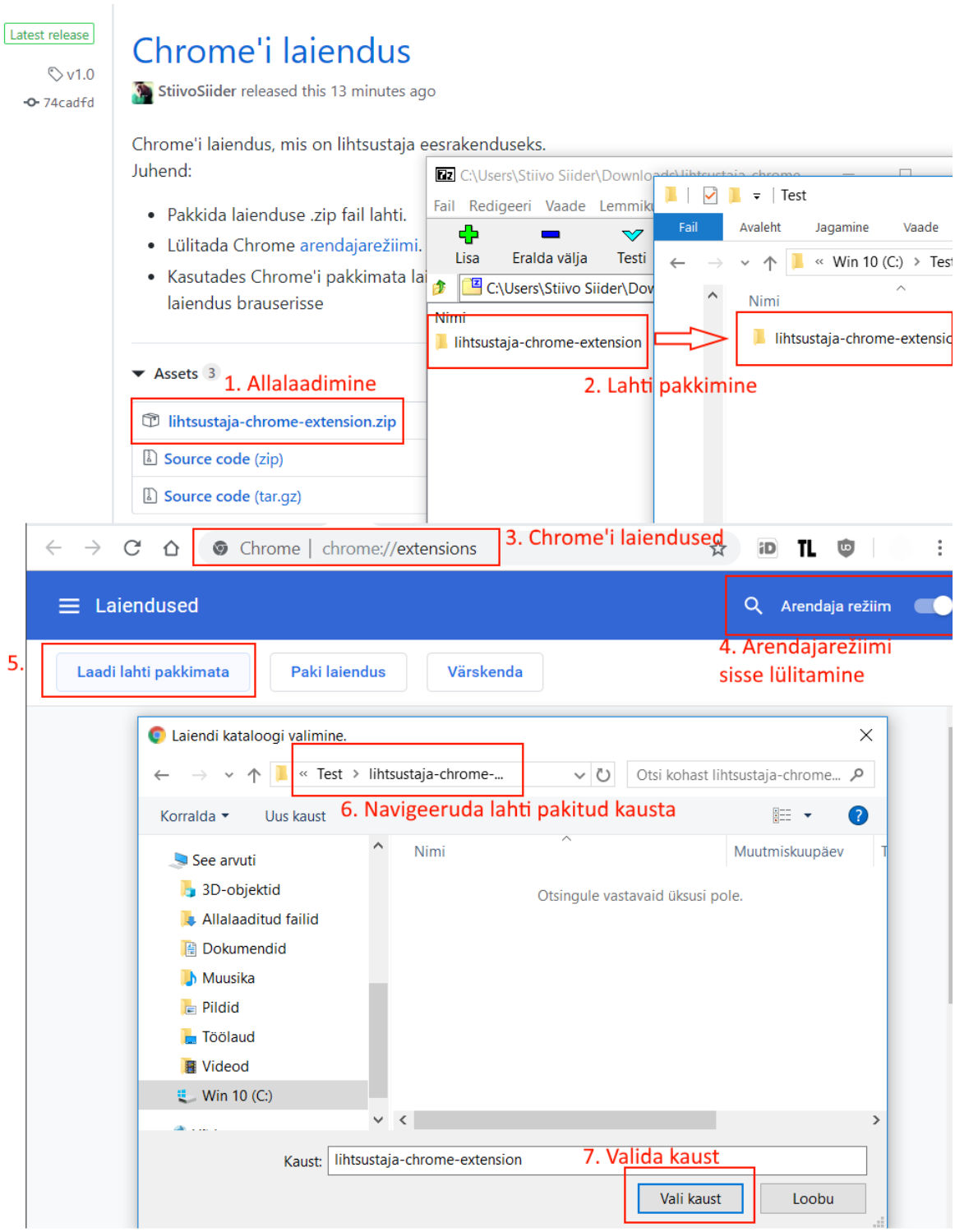
[44] Tartu Ülikooli arvutilingvistika uurimisrühm. Tasakaalus korpus. 2019. https://www.cl.ut.ee/korpused/grammatikakorpus/ (5.05.2019)

[45] W3Schools. https://www.w3schools.com/default.asp (6.05.2019)

[46] Siddharthan A., Nenkova A., McKeown K. Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Computational Linguistics*, 2011, pp. 811–842.

Lisad

I. Brauserilaienduse installeerimine arendajarežiimis



8. Laiendus on paigaldatud

II. Väljavõte testija tulemusest

```
Total 22866
Simplified 2437
Stopped 1 verb 15463
Stopped 1 main 5083
Stopped 2+ root 401
Errors 1

Ründajatel on mitmeid “ suursaavutusi ” , mis tõendab , et isegi hoolikalt kaitstud süsteemid pole ründekindlad .
Mitmeid "suursaavutusi" on Ründajatel et kaitstud isegi hoolikalt süsteemid pole ründekindlad. Suursaavutused tõendab.
------------------------------
Väärkasutuse tuvastussüsteemide põhiidee on lihtne : võrku siirdatakse agendid , mis jälgivad võrguliiklust ja võrdlevad seda teadaolevate ründekäekirjadega (intrusive signatures ) .
Väärkasutuse tuvastussüsteemide põhiidee on lihtne siirdatakse võrku agendid. Agendid jälgivad võrguliiklust. Mis võrdlevad seda teadaolevate ründekäekirjadega (intrusive signatures).
------------------------------
Anomaalia tuvastussüsteemid on kasutatavad võrkudes , mille “ normaalne ” ( baas- ) seisund on määratletav .
Anomaalia tuvastussüsteemid on kasutatavad võrkudes võrkude "normaalne" (baas-). Seisund on määratletav.
------------------------------
Kahjuks süveneb võrguadministraatorite defitsiit - infosüsteemide arv kasvab eksponentsiaalselt , administraatorite arv vaevalt lineaarselt .
Võrguadministraatorite defitsiit süveneb Kahjuks. Infosüsteemide arv administraatorite arv kasvab eksponentsiaalselt lineaarselt vaevalt.
------------------------------
Kui nii , siis vastav seos - tähistuseks-tähenduseks olemise seos - on samalaadselt elemendiks olemise seosega fundamentaalseos , mida ei defineerita .
Kui nii siis vastav seos tähistuseks-tähenduseks olemise seos samalaadselt elemendiks olemise seosega fundamentaalseos on. Ei defineerita fundamentaalseost.
------------------------------
Esitatava materjali aluseks on raamatu [ 1 ] kolmas peatükk , milles on omakorda olulisel määral edasi arendatud artiklites [ 2 ] ja [ 3 ] kirjapandud mõtteid .
Raamatu (1) kolmas peatükk on Esitatava materjali aluseks. Peatükk on omakorda olulisel määral arendatud edasi artiklites (2) ja (3) kirjapandud mõtteid.
------------------------------
Fundamentaalseosteks nimetame edaspidi selliseid seoseid ehk predikaate , mis ühes või teises teoorias ei kuulu defineerimisele , kuid samas on lähtealuseks vaadeldavate teooriate ülejäänud seoste määratlemisel .
Nimetame Fundamentaalseosteks edaspidi selliseid seoseid ehk predikaate. Predikaadid ei kuulu teises või ühes teoorias defineerimisele kuid on samas lähtealuseks vaadeldavate teooriate ülejäänud seoste määratlemisel.
------------------------------
Nii näiteks on hulgateoorias fundamentaalseoseks binaarne seos  , mis väljendab ühe hulga olemist teise hulga elemendiks .
Binaarne seos on Nii näiteks hulgateoorias fundamentaalseoseks. Seos väljendab ühe hulga olemist teise hulga elemendiks.
```

III. Küsimustiku üldine ülesehitus

Tekstilihtsustaja hindamine

Hea vastaja!

Küsimustik on koostatud tagasiside saamiseks bakalaureusetöö raames loodud tekstilihtsustajale, mis lihtsustab lausestruktuuri. Küsimustiku eesmärk on hinnata lihtsustaja korrektsust ja sobivust ning leida probleemkohti lihtsustamise protsessis.

Vastamine on anonüümne ning küsimustikuga kogutud materjali kasutatakse ainult teadusliku töö raames.

NB! Vastamiseks on soovitatav kasutada arvutit.

Stiivo Siider

* Required

Üldinformatsioon

Kas eesti keel on Teie emakeel? *

- ○ Jah
- ○ Ei

Kui Teil esineb mõni järgnevatest häiretest, siis märkige see.

- ☐ Afaasia
- ☐ Düsleksia
- ☐ Kurtus

Valitud laused

Siin osas antakse Teile ette valitud laused ning nende lihtsustused. Küsimused on 1. lausepaari korral selgitatud.

1. lausepaar

Relatiivlausete lihtsustamine

Esialgne:
Poisid ja tüdrukud, kes jäid tundi hiljaks, kuna ei kuulnud koolikella, mis helises vaikselt, said karistada.

Lihtsustatud:
Poisid ja tüdrukud said karistada. Tüdrukud jäid tundi hiljaks kuna ei kuulnud koolikella. Koolikell helises vaikselt.

2. lausepaar

Rindlause lõhkumine lihtlauseteks

Esialgne:
Päike paistab ja ilm on soe ning inimesed on isegi juba rannas.

Lihtsustatud:
Päike paistab. Ilm on soe. Inimesed on isegi juba rannas.

3. lausepaar

Relatiivlause lihtsustamine koos lause järjekorra ühtlustamisega

Esialgne:
Ilusa ilmaga on inimesed tulnud juba randa, mis Eedeni ja Tasku vahel asub.

Lihtsustatud:
Ilusa ilmaga on inimesed juba randa tulnud. Rand asub Tasku ja Eedeni vahel.

Juhuslikud laused

Siin osas palun Teil kasutada lehekülge:

http://prog.keeleressursid.ee/ss_syntax/?random=on

et saada tagasidet vähemalt viiele juhuslikule lihtsustamisele (võimalik anda kümnele).

Üleval väljatoodud lehekülg lihtsustab suvaliselt lauseid Tartu Ülikooli arvutilingvistika uurimisrühma Tasakaalus korpusest (https://www.cl.ut.ee/korpused/grammatikakorpus/), mis sisaldab lauseid aja-, ilu- ja teaduskirjandusest.

Lihtsustamine võib võtta kaua aega (30+ sek).

Lehekülje kasutamise järgselt palun Teil kopeerida kogu saadud tulemus "Tulemus" küsimuse lahtrisse. Tulemuse näide:

Esialgne lause
Positiivsete EIA reaktsioonide spetsiifilisust kontrolliti viiruse neutralisatsiooni testiga ning see kinnitas , et uuritud lambad omavad VVDV-d neutraliseerivaid antikehi .

Lihtsustatud lause
Kontrolliti Positiivsete EIA reaktsioonide spetsiifilisust viiruse neutralisatsiooni testiga. See kinnitas et uuritud lambad omavad VVDV-d neutraliseerivaid antikehi.
---
16. lause, mida vaadati.

4. lausepaar

Teie valitud laused

Siin on võimalik proovida lihtsustajat ise (http://prog.keeleressursid.ee/ss_syntax/) ning soovi korral anda tagasisidet kahele lausepaarile.

14. lausepaar

Tänan vastamast!

Soovi korral kui Teil on märkeid küsimustiku, lihtsustaja või muu sellega seonduva kohta, siis võite siin need kirja panna.

Tagasiside

Your answer

IV. Küsimused iga lausepaari kohta

Kumb lause on eelistatum? (Teie eelistus) *

- ◯ Esialgne
- ◯ Lihtsustatud

Kumb lause on loetavam (kui lihtne on lauset lugeda, kui suupärane või loomulik lause on)? *

- ◯ Esialgne
- ◯ Lihtsustatud

Kumb lause on mõistetavam (kui arusaadav on lause sisu)? *

- ◯ Esialgne
- ◯ Lihtsustatud

Lihtsustatud lause loetavus *

	1	2	3	4	5	
Väga halb	◯	◯	◯	◯	◯	Väga hea

Lihtsustatud lause mõistetavus *

	1	2	3	4	5	
Väga halb	◯	◯	◯	◯	◯	Väga hea

Kas lihtsustatud lause on grammatiliselt korrektne (eirates kirjavahemärke)? *

- ◯ Jah
- ◯ Ei

Kas lause lihtsustamisel lause mõte säilis? *

- ◯ Jah
- ◯ Ei

Muu tagasiside

Your answer

V. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina,

Stiivo Siider,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Süntaksianalüüsil põhinev teksti lihtsustaja**,

   mille juhendajateks on
   Sven Aller ja Heili Orav,

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

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

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

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

Stiivo Siider

10.05.2019