TARTU ÜLIKOOL
FILOSOOFIATEADUSKOND
EESTI JA ÜLDKEELETEADUSE INSTITUUT

Kristiina Toots

NETIKEELE METAGRAAFIA

Bakalaureusetöö

Juhendaja vanemteadur Heiki-Jaan Kaalep

Tartu 2014

Sisukord

Sissejuhatus

Käesoleva bakalaureusetöö teemaks on eesti netikeele metagraafia. Netikeel on üsna üldine mõiste, mis tähistab seda keelt, mida internetikeskkondades kasutatakse. Metagraafia hulka kuuluvad kõik märgid ja tähistusviisid peale tähestiku tähtede ja numbrite. Bakalaureusetöö kuulub uurimissuunda, mille üldine eesmärk on fikseerida netikeele metagraafia deskriptiivse uurimise lähtekohad ning püüda neid rakendada, kasutades arvutiprogrammide abi. Töö võiks pakkuda tulevikus lähtealust, millele toetuda internetikeele ja metagraafia uurimisel.

Kuigi internetikasutajate keelekasutust ei kontrolli mitte keegi – kasutajad võivad kirjutada täpselt nii, nagu nad tahavad –, siis millegipärast kehtivad teatud kirjutamata reeglid, tänu millele on tekstid küll inimestele loetavad, kuid tihtipeale käib selliste tekstide töötlemine arvutile üle jõu. Kui arvuti ei erista lausepiire ega tunne netikeelele omaseid erijooni, siis on ka võimatu teostada korrektselt automaatset märgendamist. Arvutilingvistika ja keeletehnoloogia huvituvad sellest, kuidas interpunktsioon netikeeles käitub: millised üldkasutatavad punktuatsioonireeglid on tekkinud või millised tavad on laialdaselt kasutuses (nt kuidas märgitakse lause algust või lõppu, mis märke kasutatakse jne). Hüpotees, et netikeele metagraafias on tekkinud mingid spontaansed reeglid, on küll bakalaureusetööga seotud, kuid see jääb käesoleva töö uurimisalast välja. Töös vaadeldakse pigem teatud liiki nähtust: sageli piisab ainult internetikeskkonnas kirjutatud tekstile peale vaatamisest, et näha, et see on midagi teistsugust kui tavaline kirjakeelne tekst. See on mõneti üllatav, sest isegi käsitsi kirjutatud erakirjades järgitakse üldiselt kirjakeele punktuatsiooni reegleid. Et internetis ei järgita neid reegleid, on viimase paarikümne aasta nähtus ja see väärib järelemõtlemist, kui mitte uurimist.

Siinse töö teoreetilise osa eesmärk on anda ülevaade metagraafia teooriast ning selle rakendusest vanades tekstides, netikeeles ja ka teistes valdkondades, tehnilistest võimalustest ning netikeelest ja emotikonidest. Töö empiirilise osa eesmärk on luua automaatne töövahend, mille abil on võimalik netikeele tekste uurida ning esitada mõningad näidisküsimused. Empiirilises osas analüüsitakse eestikeelsetest internetikeskkondadest pärit tekste. Materjal on saadud TÜ arvutilingvistika uurimisrühma arhiivist. Töös analüüsitakse jututubadest pärit vestlusi (92 vestlust 24 erineval teemal), netikommentaaride postitusi (77 teemal) ning foorumite vestlusi (64 teemal).

Välismaisetest autoritest on netikeele interpunktsiooni uurinud Bernard Jones, Geoffrey Nunberg jpt ning metagraafia kasutust Robert R. Provine, Robert J. Spencer, Darcy L. Mandell, Keith Houston jt. Interneti keskkondade metagraafiat pole väga palju uuritud. Eestis ei ole varem sellist analüüsi tehtud. Tartu Ülikoolis tegeleb netikeele töötlemisega arvutilingvistika töögrupp (Heiki-Jaan Kaalep, Kadri Muischnek, Siim Orasmaa) ning netikeelt ja vähesel määral ka emotikone ja interpunktsiooni netikeeles on oma töödes kirjeldanud näiteks Tiit Hennoste ja Anni Oja. Siinses töös toetutakse nii välismaistele kui ka kodumaistele netikeele uurijatele.

Bakalaureusetöö on jaotatud viieks peatükiks. Esimeses peatükis antakse ülevaade metagraafiast: metagraafia mõistest, ajaloost ja funktsioonist, kirjavahemärkidest üldisemalt ning metagraafia kasutusest keeletöötluses. Teises peatükis kõrvutatakse trükikunstieelne keel tänapäeva netikeelega ning võrreldakse nende metagraafiat. Kolmandas peatükis kirjutatakse netikeelest üldisemalt, selle kasutuskeskkondadest ja netikeelele omasest erinähust – emotikonide kasutusest. Neljandas peatükis antakse ülevaade uuritavate tekstide analüüsil kasutatud programmeerimiskeelest Python ja regulaaravaldistest, mis on tähtsal kohal tekstitöötluses. Tutvustatakse töö empiirilise osa materjali ning uurimiskäiku ja analüüsi jaoks koostatud algoritme. Viiendas peatükis kirjutatakse analüüsi tulemustest ja esitatakse töö võimalikud edasiarendused.

1. Metagraafia

1.1. Metagraafia mõiste

Metagraafia mõiste alla kuuluvad kõik märgid ja tähistusviisid peale tähestiku tähtede. „Metagraafia piirid ei ole selged. Kindlalt kuuluvad siia sõnasümbolid, tehnilised osundid, esiletõstud trükikirjadega, värvusesiletõstud, samuti kirjavahemärgid.“ (EKG II 1993: 387) Tähtis on ka lõigustus, mida on läbi aegade erinevalt märgitud: Vana-Kreekas kasutati allajoondust, keskajal eendrida, hiljem C-tähest arenenud lõigualgusmärki ¶, tänapäevast taandrida hakati kasutama alles 17. sajandil. (EKG II 1993: 387)

Netikeeles on metagraafial oma koht tekstitöötluses. Eristamaks lauseid teineteisest, on tihtipeale vaja teada, kuidas näiteks emotikone kasutatakse. Kuna neid kasutatakse tihti ka lauselõpumärkide asemel, siis peab tekstitöötlusel sellega arvestama.

2.3. Metagraafia ajalugu ja funktsioon

EKG-s kirjutatakse, et esimene märgistusega seotud nähtus pärineb Vana-Kreekast mõttelise lõigu esiletõstmisena allakriipsutamisega (*paragraphos*). Interpunktsiooni eelkäijaks Euroopas on peetud ka Aristophanest (257–180 eKr), kes lõi kolme pausipunkti süsteemi: retoorika hinge- ja lisahingepausid. Lühikese pausi nimetus oli *komma*, keskmine oli *kolon* ja pikk *periodos*, märkidena kasutati kõrget, keskkõrget ja madalat punkti. Püsima see süsteem siiski ei jäänud. Vahemärgistus tuli uuesti käiku alles keskajal. Alguses oli see harv nähtus, peamiselt munkade huviringis, kes pidid palju pühatekste ette lugema. 12. sajandil hakati taas vahemärkidega rohkem tegelema, sel ajal kujunes välja ka tänapäevane küsimärk, mis märkis küsilause intonatsiooni.

Hiliskeskajal tekkis märk nimega *virgula*, mis kandis tänapäeva komale sarnast funktsiooni ning millest areneski umbes aastaks 1700 koma. Renessanssiga elavnes ka vahemärgikultuur ning sellest ajast pärinevad ka Manuziode tööd, mis sõnastavad Euroopa vahemärgireeglid. (EKG II 1993: 386–388)

Tänapäevaks on paljud metagraafia märgid unustatud: *interrobang* (⁉ – hüüu- ja küsimärgi sümbioos, hämmastusmärk), *manicule* (☞ – 12.–18. sajanditel populaarne märk, mida lugejad kasutasid märkimaks huvipakkuvaid lõike tekstis) jpt (Houston 2013). Nii mõnigi märk on aga taasavastatud ja kasutusel arvutitrükis. Paljudes tekstiprogrammides saab kasutada lõigualgusmärki (¶ – *pilcrow*, Microsoft Wordis näitab ka tühikuid), sotsiaalvõrgustikes on laialdaselt kasutusele tulnud *hashtag* (# – trellid märkisid algselt, 14. sajandil, kaaluühikut nael, *pound*) (Houston 2013), meiliaadressi juurde kuuluv @-märk, mida kasutatakse ka netikeeles ingliskeelse prepositsioonina *at* (ühe teooria kohaselt (Allman 2012) oli märk algselt kasutusel raamatupidamises märkimaks fraasi *at the rate of*). Ka emotikoni saaks pidada metagraafia osaks. Oja (2009) sõnul on emotikon pisipildike, mis kirjeldab näoilmet, tunnet või tuju.

2.1. Kirjavahemärgid

Geoffrey Nunberg (1990: 11) ütleb, et interpunktsioon on tihtilugu kõnes esineva prosoodia ja pauside märkimiseks, kuid selle uurimine ei ole populaarne mitmel erineval põhjusel. Kui algselt huvitus kirjavahemärkidest pigem retoorika, siis tänapäeval põhinevad vahemärgireeglid siiski enamasti süntaktilistel kaalutlustel (EKG II 1993: 386), ning neist huvituvad lingvistid. Tavaliselt peetakse *interpunktsiooni* all silmas mitte-tähtnumbrilisi märke, mida kasutatakse näitamaks struktuurilisi suhteid lauseliikmete seas. Sinna hulka kuuluvad koma, semikoolon, koolon, punkt, ümarsulud, jutumärgid jms. Funktsiooni seisukohalt aga sarnanevad nad paljudele teistele

graafilistele märkidele (šriftivaheldumine, emotikonid, suurtähed, taandread ja tühikud). (Nunberg 1990:17)

Vahemärkide tehnilised funktsioonid on eraldamine (ühepoolne vahemärk) ja raamimine (kahepoolne vahemärk) ning sidumine, mis on seotud sõnamärkidega (nt sidekriips seob sõnapoolikud). Mõnel juhul ei muutu lause tähendus lause liigendusest, kuid enamasti küll. Lisaks tähenduse muutumisele muutub erineva lauseliigenduse või märgivalikuga tihti ka intonatsioon. Ka lause- ja sõnakatkestus kajastub interpunktsioonis (tavaliselt kolme järjestikuse punktiga). Kuigi kirjateksti lugedes on igati loomulik teha pause just kirjavahemärkide juures, siis vabas kõnes ei käi tihtipeale pausid ja kirjavahemärgid kokku. (EKG II 1993: 388)

1.4. Metagraafia kasutus keeletöötluses

Keeletöötluses (*Natural Language Processing*) on interpunktsiooni peaaegu täielikult ignoreeritud. Põhjuseks peetakse korraliku interpunktsiooniteooria puudumist, millele arvutilingvistiline lähenemine võiks toetuda. Selle tõttu eemaldatakse tihtipeale tekstitöötlusprogrammides kirjavahemärgid. (Jones 1996: 604) Netikeelt uurides aga võib selline tava viia mitteusaldusväärsete tulemuste ja valetõlgendusteni.

Internetikeskkondadest pärit tekstide töötlemisel on üks põhilisi probleeme see, et automaatne märgendamine ei toimi korrektselt. Programmid ei suuda toime tulla nt jututubade keelekasutusega. Märgendus nõuab lausete korrektset ülesehitust, jututubade keelekasutus aga on tihtipeale segane või on ebatavalise grammatika ja ortograafiaga. (King 2009: 304) Küberortograafia (ebatraditsiooniline ortograafia, mida kohatakse netikeeles) tõttu tekkinud probleemid korpusanalüüsi tarkvaras ei ole aga ületamatud. Ajaloolisi tekste uurivad korpuslingvistid on panustanud sellise automaatanalüsaatori arendamisse, mis suudaks märgendada ja töödelda 17.–18. sajandi ingliskeelseid tekste (Archer 2003; Rayson 2005, 2007; Pilz 2008, viidatud King 2009: 313 järgi).

Ebatraditsiooniliste tekstide märgendamine puudutab ka netikeele tekstide analüüsi. Punktuatsiooni seisukohast aga on selliste tekstide erinevuses märgata üht tähtsat erinevust – emotikonide kasutust netikeeles.

2. Trükikunstieelse keele *vs* netikeele metagraafia.

Netikeskkondades nähtav keelekasutus ei ole uudne – selliseid kõne ja kirja hübriide võib leida juba keskaegsetes tekstides. Ühelt poolt näevad lingvistid vaeva, et lausestada selliseid ebaharilikke tekste tänapäeva grammatikareeglite järgi, teisalt arvatakse, et võib-olla ei tohiksi üritada vanadele traditsioonidele toetuvaid tekste tänapäeva raamidesse suruda. Trükikunstieelsete tekstide lausetel ei pruugi küll olla tänapäeva mõttes loogilist struktuuri, kuid nad ei ole ka valesti kirjutatud – nii nagu tekst kirja pandud on, nii ju ka räägitakse.

G. Nunberg (1990) kirjutab oma töös, et ortograafiliste lausete (ortograafiline lause EKK järgi võib olla ka liitlause osalause või mitme liht- ja liitlause ühend, kuid siiski algab suurtähega ja lõpeb lauselõpumärgiga (EKK 2000: 345–346).) tänapäevane vorm ja mõiste tekkisid alles peale trükikunsti leiutamist. Levinson (1986, viidatud Nunberg 1990: 129 järgi) arvab, et trükitud tekstis saab lauseid hakata määratlema 17. sajandist, kuigi kõik kirjanikud ei järginud samu reegleid ja paljud tekstid ei vasta sugugi tänapäeva nõuetele. Näiteks on toodud tekst 1617. aastast:

> *"But then I reasoned againe, Christ was both God and man , therefore hee coulde withstand the terrors of death :but I am a fleshly man, and perhance I cannot vndergoone the cruell pangs of death:but my conscience solved all this doubt,in that the martyrs were fleshly man,and siners , yet by the grace of God were strengthened to die, therefore by the same grace shall be sustained. And in this cogitation I was much comfoorted and prevailed in spirit,&wholly gave my selfe over to suffer death:and they lead me streight waies to the place of execution, and bound me hand, and foot in maner of a corpse upon the earth, as apeareth by this figure."* (Nunberg 1990: 129)

Eelmises tekstinäites tuleks tähelepanu pöörata eelkõige punktuatsioonile: see ei järgi tänapäeva kirjakeeletavasid, kuid mõte on sellegipoolest arusaadav. Tekst on kirja pandud nii nagu seda kõneldaks, meenutades kõne- ja kirjakeele vahepealset ala. Komasid on kasutatud ebaühtlaselt, mõne märgi ees on tühik, mõne märgi järel ei ole

tühikut, ühes lauses on kaks koolonit ning kasutatud on metagraafilist vahendit &. Siinkohal saaks paralleele tuua tänapäeva netikeelega – ka seda on peetud kõne- ja kirjakeele vahepealseks alaks (vt Crystal 2001, Ferrara jt 1991, Murray 1990 diskussioone netikeele mõiste teemal). Võrdlemaks 1617. aastast pärit teksti tänapäeva netikeelega, võib vaadelda järgmisi kolme näidet, mis on pärit selle bakalaureusetöö analüüsitavast materjalist.

Postitus kommentaariumist:

> ma lausa vihkan seda euro asja , jätke meie maa rahule, anastajad maalt välja ja reeturid mõista põhiseaduse paragrahv 54 (kodaniku kohus on olla ustav põhiseaduslikule korrale ja kaitsta Eesti iseseisvust)ja karistusseadustiku paragrahv 235 (Eesti iseseisvuse vastu suunatud tegevus)alusel süüdi.
> mina jään oma esimeseŽJAh i juurde mille ma ütlesin 12 aastat tagasi iseseisvuse poolt

Postitus foorumist:

> Suured tänud Sul Andres toetava ja igati julgustava vastuse eest, vähemalt keegi ikka hoolib minusugusest abitust algajast! :))
>
> Kui ma juba siin kirjutan, et siis arendaks teemat edasi, nimelt kalipso, vest ja laud koos poomi ja purje jne. –Palju maksab? Mingit luksust ei vaja ning ka secondhand tooted on täiesti aksepteeritavad!
>
> Kas oleks arukas minna Havai Surfis harjutama või oleks arukam mõnele spetsile külje alla pugeda? Ise kahjuk ei tunne kedagit proffi,–oskajast rääkimata.
>
> Mõtteis surfates,
>
> Andrus

Postitus jututoast:

> tere lambad kas reede õhtul tõesti muud teha pole kui siin mõttetus jutukas passida!!!! tropid olete!!!!

Need näited eksivad paljuski tänapäeva kirjavahemärgireeglite vastu, ometigi on kõik tekstijupid mõistetavad. D. Crystal kirjutab, et punktuatsioonil on suur tähtsus netikeeles, kuna märkide abil saab kirjatekstile anda kõne omadusi (prosoodia, intonatsioon jm). Kirjavahemärkide kasutamine sõltub inimesest: mõni ei kasuta üldse märke, mõni kasutab siis kui vaja vältida segadust, mõni järgib kirjakeele reegleid igas situatsioonis jne. Kasutusele on tulnud ka palju uusi märke, mis ei ole traditsioonilises punktuatsioonisüsteemis kasutusel olnud, nagu näiteks trellid (#). Netis võib esineda ebatavalisi märgikombinatsioone, näiteks märgivad pausi suvaline arv punkte (...), korduvad sidekriipsud (---) või komad (,,,,,). Millegi rõhutamiseks või oma suhtumise näitamiseks saab liialdada juhuslike märkidega, nt !!!!!! või £$£$%!. (Crystal 2001: 89)

1617. aasta tekstile on iseloomulik näiteks komade ja tühikute ebaühtlane kasutusviis (nagu ka kommentaariumi ja foorumi näidetes), ka kooloneid on ebaharilikult kasutatud (nagu ka foorumi näites mõttekriipsu), raske on mõista, kus üks lause algab ja teine lõppeb (nagu ka kommentaariumi ja jututoa näites) ning sõna „ja“ asemel on kasutatud märki „&“.

G. Nunberg nendib, et mil iganes laused ka saavutasid ühise grammatilise vormi, pole nad võrreldavad keskajast ja renessansist pärit tekstide süntaksiga ja selle tõttu ei tohiks omistada sellistele tekstijuppidele „lausete“ tähendust. Kuigi lausetele omaseid vahendeid on küll kasutatud, siis see, mis kirja on pandud, ei ole vastavuses tänapäeva lauseehitusega (ei funktsionaalselt ega süntaktiliselt). Seega ei tohiks selliseid tekste arvustada nende lohaka ja ebaselge keelekasutuse poolest. G. Nunberg juhib tähelepanu sellele, et nendel kirjanikel polnud ühtki konventsionaliseerunud süsteemi, millele oma tekstide loomisel toetuda. See on ka põhjuseks miks nende „laused“ ei sisalda sellist informatsiooni struktuuri ja konteksti kohta, mida tänapäeval peetakse vajalikuks. G. Nunberg väidab, et seetõttu pole võimalik ka „moderniseerida“ vanemate tekstide punktuatsiooni. Selline tekstide kaasajastamine tuleneb valest eeldusest tänapäeva punktuatsiooni kohta: arvatakse, et kirjavahemärgid näitavad informatsiooni ja suhteid

neutraalsel viisil, seega peaksid need märgid olema kasutatavad kõikide tekstide puhul. (Nunberg 1990: 130–131)

Võib-olla peaks ka tänapäeva netikeele uurimisel lähtuma sellest, et netikeel on midagi hoopis muud kui kirjakeel? Ehk sarnaneb netikeel (või vähemalt selle metagraafia) isegi rohkem trükikunstieelsetele kui ilukirjanduslikele tekstidele ning neid ei tohikski püüda mingitesse raamidesse suruda?

3. Netikeel

Netikeel ehk uue meedia keel on internetisuhtluses kasutatav kirjaliku keeleteostuse variant. Tegelikult on netikeele mõiste väga lai: näiteks ei ole e-kirjades kasutatav keel sama, mis jututubades, ning kommentaariumides kasutatav keel pole sama, mis blogides (Herring 2007). Siinses töös käsitletakse netikeelt üldistatult ning kirjeldatakse ainult jututubade, kommentaariumide ja foorumite keelekasutust.

Sünkroonses infovahetuses on oluline kiirus, selle tõttu võib keelelise eripärana näha lühidust ja optimaalsust. Asünkroonilises infovahetuses on aga tekst tihti läbimõeldum, sest kirjutajal on rohkem aega. Optimaalsus tähendab sõnade ja fraaside lühendamist (*OMG – Oh My God*), suurtähtede puudumine lause alguses, kirjavahemärkide puudumine, näpuvigu ei parandata ning Anni Oja (2006: 260) sõnul: „välja jäetakse kõik, mida välja jätta saab“. Kuna netikeele kasutajale on optimaalsus väga tähtis, siis on ka kirjavahemärkide kasutamisel tekkinud teatud tavad. Sellised spontaanselt tekkinud ning kasutajate seas laialdaselt levinud kirjutamata reeglid on arvutilingvistika üks põhiprobleeme: kuidas arvuti mõistaks, millal on punktuatsiooniga peetud silmas lause lõppu ja millal mitte?

3.1. Netikeele kasutuskeskkonnad

Internetis on võimalik asuda suhtlusse, kus osalejaid on mitu, rääkida-kirjutada saab samaaegselt, saades kirjutatule vastuse paari sekundi jooksul või vastust võib oodata nädalaid. Suhtluskeskkonnad on näiteks uudisgrupid, meililistid, diskussioonilistid, e-konverentsid, kommentaariumid, foorumid, jututoad jpt. (Crystal 2001: 129) Selles töös keskendutakse kolmele viimasele. Anni Oja (2006: 259) on öelnud, et keeleuurija jaoks on internetisuhtluses tähtsad aspektid aeg, vestlusparterite arv ja suhe. Aja järgi

jaotatakse suhtlus sünkroonseks (nt MSN) ja asünkroonseks (nt e-kiri) infovahetuseks. Vestluspartnerite arvu järgi saab jaotada internetisuhtluse monoloogiks, dialoogiks või multiloogiks. Monoloogis on publik enamasti anonüümne ning tagasisidet ei oodata. Dialoogis on kaks osapoolt ning suhtlusvõimalusi on mitmeid. Multiloogis osaleb rohkem kui kaks inimest ja suhtlus võib toimuda näiteks jututubades või kommentaariumides. Anonüümsus torkab eriti esile internetikommentaarides. Kui identifitseerimine ei ole nõutud, siis on kommentaarid tihtipeale teravamad ja solvavamad. Eristada saab ka avalikku ja privaatset suhtlust. (Oja 2006: 260–262)

Jututuba on sünkroonne suhtluskeskkond, kus inimesed saavad suhelda nii avalikult kui ka privaatselt. Keelekorpuse materjali kogumiseks on lubatud kasutada programme, juhul kui on saadud luba jututoa administraatorilt ja kaasvestlejailt. (Oja 2006: 262) Sellises keskkonnas astub kasutaja käimasolevasse vestlusesse, mis toimub reaalajas. Kasutaja kirjutatu saadetakse ühele keskarvuti aadressile ning see tekst sisestatakse omakorda ekraanile, mis uueneb pidevalt teiste kasutajate kirjutatuga. Jututoa liikmed näevad oma sisestatud teksti kohe peale selle saatmist ning sellele oodatakse ka kiiret vastust. (Crystal 2001: 130)

Kommentaarium on asünkroonne suhtluskeskkond, mis on osa mingist teemaveebist, kus teksti juurde on võimalik jätta oma arvamus. Inimesed avaldavad oma mõtteid vabamalt ning konfliktid on tavalised. Kuigi kommentaarid on tihtipeale anonüümselt postitatud, on võimalik „käekirja“ põhjal eristada autoriteete, kelle arvamusi teised kasutajad tihti toetavad. (Oja 2006: 265)

Foorum on samuti asünkroonne suhtluskeskkond. Foorumites on üldjuhul ette antud kindel üldteema, mis on omakorda jaotatud alateemadeks, tuues kokku huvitundjad kindla valdkonna vastu. Tekst on tavaliselt kirjakeelsem kui sünkroonsetes vestlustes ning suhtlus asjalik. Samas võib tekkida ka konflikte. Nagu jututubadeski, hoiavad vetlustel silma peal moderaatorid. (Oja 2006: 264)

Sellistes asünkroonsetes keskkondades saadetakse sama moodi nagu sünkroonses keskkonnas, kirjutatu keskarvuti aadressile, kuid sellised tekstid jäädvustatakse teatud formaadis, nii et gruppide kasutajad näevad kirjutatut isegi siis, kui möödunud on palju aega. Kasutajatele ei ole tähtis oma postitust koheselt näha ja kiiret vastust ei eeldata. Sellised grupid on loodud tänu mingile teatud ühisele huvile (ühine teema või küsimus), asjaosalised võivad olla nii amatöörid kui ka spetsialistid. (Crystal 2001: 130–132)

3.2. Netikeele iseloom

1980ndatel ja 90ndate alguses hakati proovima klassifitseerida netikeele kasutust. Olles harjunud kahe erineva keelenähuga – kõne ja kirjaga –, hakkasid lingvistid arutlema selle üle, et kas netikeel on eriline kirjaviis, sest see on kirja pandud kasutades klaviatuuri ja loetud arvutiekraanilt. Või on see "kirjutatud kõne" (Maynor 1994), sest tal on kõne tunnused, sh kiire vooruvahetus, mitteformaalsus, ning tal esineb prosoodiale omaseid nähtusi? Või on see hoopis kõne ja kirja vahepealne ala? (Ferrara jt 1991, Murray 1990)

Sellised algsed püüded seletada netikeele mõistet olid siiski üsna üldised, justkui netikeel oleks ühene, homogeenne žanr või suhtlemise tüüp. Üks võimalusi netikeele kirjeldamiseks võiks olla David Crystali (2001) pakutud variant: netikeel on mitme variandiga veebis kasutatav keel, mida iseloomustavad lühendid, emotikonid ja mittestandardne õigekiri. Netikeele variandid on tundlikud erinevate tehniliste võimaluste ja olukordade suhtes, mistõttu on netikeel kompleksne ja muutuv (Baym 1995, Cherny 1999). Siiski on täheldatud, et teatud kasutuses ei muutu netikeel sugugi kiirelt. Näiteks nendib Anni Oja peale oma 2009. aastal korraldatud uurimust Delfi kommentaaride keelest, et kirjutajate keelekasutuse muutumist ei olnud selles keskkonnas 10 aasta jooksul näha. Kõnekeele osakaalu kasvu oli küll aimata, kuid A. Oja ei pea seda sugugi halvaks märgiks. (Oja 2009)

Internetis on kujunenud teatud etikett. Kuigi „netiketti“ ei nõuta üheltki internetikasutajalt, siis järgivad hea tava austajad reegleid sellegipoolest. 20 aasta tagusest „netiketist“ on kirjutanud Tiit Rummo artiklis „Käitumine Internetis“ (1995). Mõned tänapäeva interneti käitumisreeglid puudutavad ka netikeele ortograafiat. Näiteks ei soovitata kirjutada oma sõnumeid läbivalt trükitähtedega – netikeskkondades tähendab see karjumist. D. Crystal (2001: 15) kirjutab, et pole ühtki universaalset käitumisreeglit selle kohta, kuidas erinevates keskkondades kirjutada, ometi teatakse, kuidas peaks meilides kirjutama, kuidas jututubades ning kuidas kommentaariumides ja foorumites. Isegi ilma reeglistikuta mõistavad kasutajad teineteist väga hästi. Arvutid kahjuks veel mitte.

Kuna jututubades toimub suhtlemine sünkroonselt, siis on kasutajatele tihtipeale tähtis kiire vooruvahetus ning selleks kasutatakse kirjaviisi, millele vastab ingliskeelne termin *Instant Messaging*. Tiit Hennoste (2013) on seda nimetanud eesti keeles terminiga „spontaanne kirjalik netivestlus reaalajas“. Need on näiteks SMSid ning MSNi ja jututubade vestlused. Siinses töös jäädakse kasutama rahvusvahelist lühendit IM.

Üldiselt peetakse IM-i mitteametlikuks kirjaviisiks, millel on kõnekeele stiil, kirjavahemärgid võivad puududa, kirjutatakse hooletult ja kasutatakse liialt lühendeid (tihti jäetakse sõnades täishäälikud ära) ning iseloomulik on ka ebastandardne õigekiri. IM-is kasutatavat interpunktsiooni pole küll palju uuritud, kuid järjest enam pööratakse tähelepanu selle tähtsusele. (Zhou, Zhang 2005) Kuna teadlastele polnud veel päris selge, kuidas inimesed tajuvad keelereegleid, eriti kirjavahemärke IM-is, siis viidi läbi katse (Zhou, Zhang 2005: 391–393). Leiti, et 99,14% juhtudest jäeti lauselõpumärk ära, pikemad laused olid jagatud lühemateks osadeks ja iga osa saadeti eraldiseisva reana, kirjavahemärgid jäeti enamasti ära kirjeldavais lauseis (mitte küsilauseis vms) ning õigekirjale ei pööratud tähelepanu: näiteks kasutati lause alguses suurt tähte väga harva. (Zhou, Zhang 2005: 397–398)

Tänapäeva IM-i (näiteks sõnumikasutust) on võrreldud 19. sajandi telegrammidega – mõlemad on lühikesed kiirsõnumid. Kui telegraafid tulid laialdasemalt kasutusele, leidus peagi ka skeptikuid, kes olid kindlad, et uus suhtlusvahend laastab kirjanduse valdkonna ja inimeste keeleoskuse (Nunberg 2008). Arvati, et tulevikus on kirjanikud napisõnalised, ei kasuta piisavalt täiendsõnu ega sünonüüme ning tekstid meenutavad lihtsalt raamatute kokkuvõtteid (Swackhamer 1848: 409–413). Öeldakse, et trükikunstil, telegraafil, telefonil ja ka televisioonil on olnud samasugune mõju ühiskonnale nagu seda on tänapäeval internetil. Nende mõjudega on kaasnenud ka sarnased hirmud: kas selliste meediumite abil võib tekkida kaos (nii lingvistiline kaos kui ka ühiskondlik)? (Crystal 2001: 2)

3.3. Emotikonid

Internetikeeles on mitteverbaalsete vahenditena kasutusel emotikonid. Emotikonid on abiks öeldu rõhutamisel või pehmendamisel ning emotsioonide või tundevarjundite edasiandmisel. Kasutusel on nii paarist kirjamärgist koosnevad pildid kui ka mitmerealised sümbolpildid. (Oja 2006: 260) Emotikonide teemal on viimastel aastatel üsna palju kõneldud ja kirjutatud (vt Dresner, Herring 2010, Provine jt 2007). Siinses bakalaureusetöös pole ehk emotikonide kirjeldamine ja nende selgitamine nii tähtis – võib arvata, et suur osa arvutikasutajatest teab, millega tegu on. Tähelepanu tuleks juhtida hoopis sellele, kuidas emotikone kasutatakse lauseis ning kuidas nad asendavad kirjavahemärke. Lingvistide jaoks on huvitav ka see, et miks on emotikone just arvutikirjas hakatud kasutama – võimalus nende kasutamiseks on olemas olnud kirjutamise kasutuseletuleku algusest saadik. D. Crystali (2001: 38) üks seletus on, et netikeel on teistest kirjutatud vormidest sarnasem kõnekeelele ja selle tõttu vajab emotsioonide ja žestide väljendamiseks lisaabivahendeid. Näiteks raamatutes ja ajakirjades kasutatavat keelt on võimalik vormida nii, et lisamärke pole vaja – autoril on piisavalt aega oma tekst arusaadavaks teha.

Emotsioonide väljendamine vestluses on loomulik osa inimese spontaanses kõnes. On märgatud, et näost näkku kõneledes esinevad naerupahvakad ja itstitused enne või peale lauset või fraasipiiril, ning peaaegu mitte kunagi fraasi sees (*Sa lähed kuhu? – haha vs Sa lähed – ha-ha – kuhu?*). Kõnes ei esine naerupahvakud juhuslikult, vaid on seotud pausidega fraasipiiridel. Internetikeskkondades, nii asünkroonsetes (nt e-kirjades) kui ka sünkroonsetes (nt jututubades), kasutatakse emotikone sarnaselt suulise kõne emotsiooni väljendamisega: emotikonid ei lõhu üldjuhul fraase ning nende kasutus netikeeles on sarnane kirjavahemärkide kasutamisele. (Provine jt 2007: 299–301)

2007. aastal uuriti emotsioonide väljendamist internetis ning tulemustest kirjutati artiklis „Emotional Expressions Online“ (Provine jt 2007). Analüüsitavaks materjaliks olid küll ingliskeelsed foorumid, kuid paralleele emotikonide kasutamisega saab tuua ka eestikeelsete netikeele kasutajatega. Tulemustes selgus, et emotikonid ei olnud tekstis suvaliselt, vaid nad esinesid vägagi aimatavates ja lingvistiliselt paljutähendavates kohtades: 836 lausest asetses emotikon enne või peale lauset või fraasipiiril 829 korral (99%) ning keset fraasi esines emotikone vaid 7 korral (1%). Emotikonide harv paiknemine keset fraasi on eriti märkimisväärne, sest statistiliselt on sellise paiknemise jaoks palju rohkem võimalusi kui fraasipiiridel. Näiteks lauses „*Before you know it, you'll be wishing you still had all that free time! :)*“ (Provine jt 2007: 303) on vaid kolm erinevat võimalust paigutada emotikone fraasipiiridel: alguses, lõpus ja komakohal ning 12 võimalikku kohta, mis ei ole fraasipiiril. (Provine jt 2007: 301–303)

Artikli autorid arvavad, et emotikonide kasutus on ilmselge püüe kompenseerida kuuldelist ja visuaalselt informatsiooni, mis on kättesaadav näost näkku kõneldes (Provine jt 2007: 305).

4. Materjal ja meetod

Analüüsitav materjal on pärit TÜ arvutilingvistika uurimisrühma arhiivist. Töödeldakse selliseid tekste, mis on juba ka korpustes, kuid nende korpuste algset metagraafiat on teisendatud. TÜ arvutilingvistika uurimisrühma kogutud nn uue meedia korpus sisaldab uudisgruppide, foorumite, kommentaariumide ja jututubade tekste. Selles bakalaureusetöös on töötlemiseks võetud kolme viimase korpuse algsemad, originaalilähedasemad variandid. Praeguseni puudub sobiv tekstitöötlusprogramm, mis suudaks segasemaid, ebastandardseid jututubade, foorumite ja kommentaariumide tekste töödelda. Põhjuseks on selliste keskkondade netikeele väga vaba kasutus, kus näiteks lause algust ja lõppu ei suuda arvuti eristada.

Töös analüüsitakse jututubadest pärit 92 vestlust 23 erineval teemal, kommentaariumidest vestlusi 77 teemal ning foorumitest vestlusi 64 erineval teemal. Jututubade vestlused on salvestatud aastast 2003 ning ühe vestluste keskmiseks pikkuseks on u 18 tundi. Kommentaariumid pärinevad aastast 2004 ja üks kommenteeritav teema sisaldab keskmiselt 2800 kommentaari. Foorumid on pärit aastatest 2002–2004 ning iga teema sisaldab umbes 2500 postitust. Materjali analüüsimiseks kasutati programmeerimiskeelt Python.

4.1. Miks programmeerida

Michael Hammond (2003) kirjutab sellest, et miks lingvistid, psühholingvistid, kirjandusteadlased jt peaksid üldse midagi tahtma programmeerimisest teada. Teoses käsitletakse küll programmeerimiskeelt Perl, kuid programmeerimise ja lingvistikaga seotud küsimustele saab sellegipoolest vastuseid. Tänapäeval on peaaegu võimatu keelematerjale läbi töötada ilma arvuti abita. Materjale töödeldakse, analüüsitakse,

liigitatakse, jaotatakse jpm arvutite abil. Kuigi erinevaid tarkvarapakette on küll loodud keeleuurijatele, siis selleks, et tõeliselt omandada uuritav valdkond, on hädavajalik mingil määral vallata programmeerimist. (Hammond 2003: 1)

M. Hammond kirjutab, et kui keeleuurija soovib teostada suuremat sorti analüüsi ning kui programmeerida ei osata, siis on vaid mõned võimalused töö läbiviimiseks. Näiteks võib seda käsitsi teha (kuigi suure tekstihulga puhul võib see meetod olla veidi riskantne), kellegi palgata programmeerima või kasutada mõnd olemasolevat tarkvarapaketti. Viimane variant on küll vahetevahel piisav ja sobilik, kuid mitte alati. Paketi võimalused võivad olla piiratud: uurija vajadused ei pruugi minna kokku sellega, milleks pakett disainitud tegema oli. Ühtlasi ei pruugi sellised tarkvarapaketid olla kasutatavad intuitiivse mõtlemise abil, vaid vajavad käskude tundmist või on vajalik teatud kontrollkeele (*control language*) selgeks õppimine. Ning lisaks ei pruugi kõik paketid kõikide operatsioonisüsteemidega töötada või on nad kulukad. (Hammond 2003: 1–2)

4.1.1. Programmeerimiskeel Python

Python loodi 1990ndatel Hollandis. Selle põhiliseks autoriks on Guido van Rossum Stichting Mathematisch Centrumist, kuigi arengule aitasid kaasa veel paljud teisedki (HL). Praeguseks on Python kujunenud üheks populaarseimaks programmeerimiskeeleks. Tema erilisus väljendub tema lihtsuses. Võrreldes teiste programmeerimiskeeltega, on Pythonit kergem ära õppida, see on võimas ning kõigile kättesaadav ja kasutatav kõikide operatsioonisüsteemiga arvutitel. Pythoni kogukond on loonud abimaterjalid nii algajatele kui ka edasijõudnud programmeerijatele. (Welcome to Python) Tänu selle keele populaarsusele on abimaterjalideks loodud erinevaid veebipõhiseid kursuseid, mis aitavad soovijail Pythoni õppimisega algust teha (nt Codecadamy ja LearnPython.org).

Nagu enamikes programmeerimiskeeltes, on ka Pythonis valmis kirjutatud hulk funktsioone (mooduleid). Nii mõnigi neist käivitub automaatselt, kuid paljud moodulid tuleb sisse tuua (importida). Tekstitöötluses on tähtsal kohal regulaaravaldiste moodul.

4.1.2. Regulaaravaldised

Üks põhilisi abivahendeid tekstitöötluses on Pythoni juurde kuuluv regulaaravaldiste moodul (kasutatakse ka teistes programmeerimiskeeltes). Regulaaravaldiste (ka regexp, regex või re) abil leitakse mustreid tekstis ehk nende abil on võimalik iseloomustada tekstijada (nt veebis otsingumootorites mõne teatud sõna leidmiseks). Regulaaravaldised on tähtsad tööriistad nii informaatikas kui ka lingvistikas: neil on tähtis osa näiteks tekstitöötluses ja korpusanalüüsis. Regulaaravaldis on selline valem, mis kirjeldab sõnesid. Sõne (*string*) on jada, mis koosneb ükskõik, millistest tähtnumbrilistest (tähed, numbrid, tühikud, kirjavahemärgid jm) osadest. Regulaaravaldiste abil tehtud otsing nõuab mustrit, mida otsitakse ning tekstikorpust, kus otsingut teostada. (Jurafsky, Martin 2000: 21–23)

Pythonis kasutatakse langjoont („\“), võimaldamaks kasutada erimärke literaalsete sümbolitena (nt „.“ võib tähendada ükskõik, mis märki, aga „\.“ tähendab punkti). Mõningaid erimärke (The Python Standard Library) leiab tabelist 1.

Tabel 1. **Erimärgid.**

Erimärk	Tähendus
`.`	ükskõik, mis märk, v.a reavahetus;
`^`	sõne algus;
`$`	sõne lõpp;
`*`	kordab eelnevat re-d 0 või rohkem korda;
`+`	kordab eelnevat re-d 1 või rohkem korda;
`?`	kordab eelnevat re-d 0 või 1 kord;
`{m}`	kordab eelnevat re-d täpselt *m* korda;
`{m, n}`	kordab eelnevat re-d *m* kuni *n* korda.

4.2. Uurimisprotsess

Materjali analüüsimiseks kirjutas selle bakalaureusetöö autor Pythonis programmi, millel on kolm erinevat varianti (Lisa 1, Lisa 2 ja Lisa 3) erinedes üksteisest vaid mõne sisendi või käsu poolest. Need kolm peaprogrammi suudavad töödelda vastavalt kommentaariumide, foorumite ja jututubade tekste. Kõigis kolmes programmis on imporditud funktsioonid neljandast, eraldiseisvast programmist, mille on samuti loonud töö autor (Lisa 4).

Kuigi kõiki kolme programmi kasutati, siis tekkis sellegipoolest analüüsi käigus mõningad komplikatsioonid millele tuleb tähelepanu pöörta. Nimelt olid foorumi postitused raskesti töödeldavad, kuna nende märgendused olid väga eripalgelised ning programm ei suutnud postitusesiseseid tekstijuppe alati üheks postituseks lugeda. Näiteks kui mõni kasutaja oli mõnd eelmist postitust tsiteerinud või vastanud mõnele postitusele kasutades funktsiooni *reply,* ning selle tsiteeritava postituse üks rida oli liiga pikk, siis lisati see automaatselt uuele reale ning selle tõttu kaotas uuelt realt tsitaadimärgi, mille järgi programm oleks saanud orienteeruda tekstis. Selle tulemusena sai uuel real olev lause uueks postituseks (kuigi ta seda ei oleks tohtinud olla). Näide ühest tsiteeritud postitusest:

```
> Tere!
>
> Kivisöe probleem on, et teda on raske koduses katlamajas kasutada, sest
tema
> automatiseerimise seadmed on kallid. Brikett (puidu, kui ka turba) on
ikkagi
> eelkõige käsitsi põletamisele mõeldud.
>
```

Postituses näitab märk > kordust. Sõnadel „tema“ ja „ikkagi“ ei ole aga seda märki ees ning selle tõttu ei suuda programm neid sõnu siduda ülejäänud tekstiga. Tulevikus on ehk selle analüüsi programmi võimalik täiustada nii, et suudetaks töödelda ka praeguseid tekste, kuid selles bakalaureusetöös jäädi algse variandi juurde. Kuigi foorumi kohta käivaid andmeid ei saa täielikult usaldada, siis kasutati andmeid

sellegipoolest, sest selliseid erijuhte ei ole väga palju ning üldpildi kuvamiseks ja võrdluseks võib neid andmeid kasutada küll.

Üheks takistuseks olid ka täpitähed. Kuigi tavaliselt on võimalik `utf-8` piirangud kõrvaldada, siis teadmata põhjustel polnud seda Sublime Text 2 tekstiredaktoris (programmide kirjutamiseks mõeldud tekstiredaktor) võimalik teha. Seega otsides näiteks suurtähti, siis ei leita tähti Õ, Ä, Ö ja Ü. Kuna täpitähti on tähestikus vaid neli, ning selliseid sõnu pole palju, mis algaksid täpitähtedega, siis ei arvatud, et täpitähtede kohta informatsiooni puudumine takistaks tekstide töötlemist ja tulemuste analüüsimist. Vaadates lisasid 1, 2, 3 ja 4, on märgata, et programmi on kommenteeritud inglise keeles. Inglise keelt on kasutatud just selle tõttu, et täpitähti ei saadud kasutada ka väljakommenteeritud osades.

4.2.1. Programmi algoritmi üldine kirjeldus

Peaprogrammides imporditakse esiteks Pythonis sissekirjutatud moodul `os`, mis sisaldab operatsioonisüsteemi spetsiifilist funktsionaalsust, nagu näiteks kataloogidest failide kuvamist ja töötlemist. Teiseks imporditakse autori kirjutatud moodul, mis sisaldab funktsioone, mille abil tekstitöötlust teostatakse. Näiteks sisaldab see moodul funktsiooni, mis eemaldab tekstidest arvutiloetavad märgendid (*tag*id).

Kõikides programmides antakse erinev indeks, mille abil leitakse kataloog, milles on töödeldavad failid. Iga programm leiab kas kommentaare, foorumite postitusi või jututubade tekste hõlmavad kataloogid.

Järgmiseks luuakse tühilist (`super_array`), kuhu hiljem lisatakse üheks elemendiks uus list, mis sisaldab ühe faili postitusi. Näiteks saab foorumite puhul listi `super_array` esimeseks elemendiks uus list, kus iga element on teema „arvutid.ajaviide“ üks postitus, teiseks elemendiks on uus list, kus iga element on teema

„arvutid.cad“ üks postitus, kolmandaks elemendiks on uus list, kus iga element on teema „arvutid.disain“ üks postitus jne.

Lisamaks analüüsitavate failide tekstid listi `super_array`, kasutatakse `for` tsüklit, mis käib üle failides esinevate tekstide ning kutsub välja funktsiooni, mis töötleb seda faili nii, et tagastatakse list, kus elementideks on failis esinenud postitused ilma märgistuste, kommentaaride, korduvate tekstide ja teiste arvutiloetavate märgendusteta. Sellest, kuidas täpsemalt need märgistused eemaldati, kirjutatakse peatükis 6.2.1 Algoritmid. Käsu `append` abil lisatakse funktsioonist tagasisaadud list omakorda listi `super_array`. Kui kommentaariumi ja foorumi programmis kasutatakse identset `for` tsüklit, siis jututoa programmis kasutatakse veel üht `for` tsüklit teise tsükli sees, sest failid olid paigutatud veel eraldiseisvatesse kataloogidesse ja mõnes kataloogis võis olla veel teisigi katalooge.

Edasi luuakse uus tühilist `data`, mida kasutatakse salvestamaks tekstide kohta käiv informatsioon. Näiteks lisatakse `data` esimeseks elemendiks üks postitus, kommentaar või jututoa üks voor, teiseks elemendiks number, mis väljendab suurtähti tekstijupi alguses, kolmandaks elemendiks number, mis näitab mitmel korral on tühik jäetud kirjavahemärgi ette jne. Need arvutused kutsutakse esile kasutades funktsioone moodulist `functions`. Selleks, et funktsioone esile kutsuda, kasutatakse `for` tsüklit, mis käib üle listi `super_array` ning selle sees luuakse omakorda uus `for` tsükkel, mis käib üle elementide, mis asuvad `super_array` elementides. See tähendab, et iga uuritav element on üks postitus failis. Teise `for` tsükli sees luuakse uus tühilist (`vahelist`), kuhu salvestatakse ühe faili kõik andmed ning peale andmete kogumist lisatakse `vahelist`i salvestatud tulemused listi `data` funktsiooni `append` abil. Kuna tegemist on `for` tsükliga, siis kirjutatakse `vahelist` üle järgmiste failide andmetega, mis andmete kogumise lõpus lisatakse uueks elemendiks listi `data`. Kui `for` tsükkel on käinud üle kõikide failide, siis on koostatud list `data`, mis hõlmab endas omakorda liste, mis kannavad informatsiooni iga foorumite, kommentaaride või jututubade

postituste kohta. Listi `data` on kokku võimalik salvestada 13elemendilisi liste (vt tabel 2).

Tabel 2. **Elemendis kantav informatsioon.**

`data[n]`	elemendis kantav informatsioon
`n = 0`	töödeldava postituse tekst;
`n = 1`	mitu korda leitakse mustrit, mis vastab sellele, et...
`n = 2`	...kasutatakse suutähti;
`n = 3`	...punktuatsioonile eelneb tühik;
`n = 4`	...punktuatsioonile järgneb tühik;
`n = 5`	...punktuatsioonile järgneb tühik, millele järgneb suurtäht;
`n = 6`	...peale uut rida (*newline*) järgneb väike täht;
`n = 7`	...peale uut rida (*newline)* järgneb suurtäht;
`n = 8`	...tekstile järgneb tühik ja emotikon, millele järgneb suurtäht;
`n = 9`	...tekstile järgneb tühik ja emotikon, millele järgneb väike täht;
`n = 10`	...postituses on üksik emotikon;
`n = 11`	mitu tühikut on postituses;
`n = 12`	mitu tähtnumbrit on postituses.

Peale viimast `for` tsüklit on võimalik erinevate käskude abil leida vastuseid erinevatele küsimustele. Näiteks võib foorumipostitusi analüüsivas programmis käsu `print len(data)` abil leida mitu postitust oli kõikides foorumites kokku. Või kasutades `for` tsüklit võib arvutada kokku näiteks kõik `data[10]` arvud ning selle kaudu leida mitu üksikut emotikoni leidus kõikides tekstides kokku. Või `for` tsükli ja `if` tingimuse abil võib leida, mitmes erinevas postituses võib leida mingeid teatud mustreid (mitte mitu neid mustreid kokku on, vaid mitmes postituses neid esineb).

Programmid väljastavad kirjeldavat statistikat. Kasutades neid andmeid, saab võrrelda nii ühe netikeele kasutuskeskkonna arvulisi näitajaid, kui ka kasutuskeskkondade vahelisi näitajaid. Näiteks saab öelda, et kommentaare oli kokku 36 645, neist 32 131 korral kasutati suurtähti. Samas saab ka öelda, et üksikuid emotikone esines kommentaariumides vaid 0,08% kordadest, foorumites 0,16% ja jututubades 3,68% kasutajate sisestatud tekstidest.

4.2.2. Algoritmid

Moodulis `functions` imporditakse moodul `re`, mis annab võimaluse kasutada regulaaravaldisi. Funktsioone on kokku 13 (vt Lisa 4). Kolm esimest funktsiooni leiavad regulaaravaldiste abil märgendused, mis näitavad, kus kasutajate postitus algab. Näide kommentaariumist:

```
<kiri>
<teema> Priidik, 18.09.2003 11:37</teema>
 Egas reamees saa omapäi tegutseda ja arhiivi minna.
 Vähemalt Teise Taseme Boss peab pungale käsu andma, et uusi
 uskumatuid lugusi juhtuma saaks hakata.
</kiri>
```

Regulaaravaldis `"</teema>(.*?)</kiri>"` leiab kasutaja sisestatud teksti „Egas reamees saa omapäi tegutseda ja arhiivi minna. Vähemalt Teise Taseme Boss peab pungale käsu andma, et uusi uskumatuid lugusi juhtuma saaks hakata.“

Näide foorumist (kasutaja perekonnanimi ja meiliaadress on kustutatud):

```
<kiri>
<aeg> Sun, 05 Jan 2003 22:49:02 +0200</aeg>
<autor> Vahur _______ <_______@_____________.ee></autor>
<teema> Re: mk-etapp =?ISO-8859-1?Q?otep=E4=E4l?=</teema>
<jutt>

Taavi wrote:

> Millised kohad oleksid kõige paremad raja ääres sõidu vaatamiseks??
> Hea koht
> on siis selline, kus näeb suuskajaid kaua ja näiteks mitmes suunas kah.


Üldiselt on ju rajakaardid netis saadaval. (www.sportnet.ee/mk) Vaata ja
tudeeri ise.

Vahur

</jutt>
</kiri>
```

Võrreldes kommentaariumide tekstidega, on foorumite postitused keerulisemalt märgendatud. Analüüsimisest peab välja jääma märgistused aja, autori ja teema kohta, jutu ja kirja algust ja lõppu märkivad märgendused ning ühtlasi peab välja jääma ka see tekst, millele kasutaja on vastanud. Selle tõttu on kasutatud ka rohkem erinevaid regulaaravaldisi vajalike postituste leidmiseks. Regulaaravaldised on välja toodud tabelis 3.

Tabel 3. **Foorumi märgendusi otsivad regulaaravaldised.**

Re	Välja jäetakse tekstiosad...
`"^\s*[<>-]"`	...mis algavad tühiku või märkidega <, > või -;
`"^[\s>]"`	...mis koosnevad vaid tühikust;
`"^.*<.*>.*$"`	...kus keset rida on märgid < või >;
`".*wrote:*$"`	...mis sisaldavad sõne “wrote:”;
`".*@.*"`	...mis sisaldab meiliaadressi;
`"^0.0.$"`	...kus esineb märgistus “0.0.”;
`"^[\s]*[\w\d\?\.,!]*[\s]*$`	...kus on ühesõnalised read, millel on rohkem kui üks tühik alguses ja lõpus.

Näide jututoast:

```
[01:24:16] * Power666_eemal is now known as Power666_zzZZzz
[01:24:25] <Urgo> mis toimub
```

Re `"\[[0-9]{2}:[0-9]{2}:?[0-9]{0,2}\]\s<[^>]+>\s(.*)\r\n"` leiab mustrid, mis näitavad aega ja kasutajanime, postituse ja rea lõppu. Kusjuures kellaaeg võib olla märgitud kahel erineval viisil: kas minuti või sekundi täpsusega. Selles näites peab funktsioon kätte saama vaid kasutaja Urgo sisestatud teksti „mis toimub“.

Järgmised kuus funktsiooni leiavad postitustest teatud mustreid. Mustrite vasted lisatakse listi ning programmile tagastatakse nende listide pikkused. Tabelis 4 on näidatud, milliseid mustreid otsitakse.

Tabel 4. **Funktsioonide regulaaravaldisi.**

Re	Re funktsioon
`"[A-Z]"`	kasutatakse suurtähti;
`"[A-Za-z0-9]+\s+[\.\?!,]"`	tekstile järgneb vähemalt üks tühik ja tühikule punkt, küsi- või hüüumärk, koma;
`"[A-Za-z0-9]+[\.\?!,]+\s+"`	tekstile järgneb vähemalt üks punkt, küsi- või hüüumärk, koma ning märkidele järgneb vähemalt üks tühik;
`"[A-Za-z0-9][\.\?!]\s[A-Z]"`	tekstile järgneb punkt, küsi- või hüüumärk, millele järgneb tühik ja suurtäht;
`"^\r?\n?\s?[a-z]"`	rea alguses võib esineda kas šrifti- või reavahetus või tühik ning järgneb väike täht;
`"^\r?\n?\s?[A-Z]"`	rea alguses võib esineda kas šrifti- või reavahetus või tühik ning järgneb suurtäht.

Funktsioon, mis otsis emotikonidega seotud mustreid kutsus omakorda välja teise funktsiooni, mis otsis 121 erinevat emotikoni. Kusjuures iga emotikoni viimase märgi kordust loetakse samaks emotikoni originaalvariandiga. Näiteks on järgnevad kaks emotikoni loetud samaks:

```
:)          :)))))))))
```

Regulaaravaldis, mis otsib emotikone, on järgnev:

```
"(:-\)+|:\)+|:o\)+|:\]+|:3+|:c\)+|:>+|=\]+|8\)+|=\)+|:\}
+|:\^\)|:-D+|:D+|8-D+|8D+|x-D+|xD+|X-D+|XD+|=-D+|=D+|=-3
+|=3+|B\^D+|>:\[+|:-\(+|:\(+|:-c+|:c+|:-<+|:<+|:-\[+|:\
[+|:\{+|;\(+|:-\|\|+|:@+|>:\(+|:'-\(+|:'\(+|:'-\)+|:'\
+|D:<+|D:+|D8+|D;+|D=+|DX+|v\.v+|D-':+|>:O+|:-O+|:O+|
:-o+|:o+|8-0+|O_O+|o-o+|O_o+|o_O+|o_o+|O-O+|:\*|:\^\*+|
\(\s'\}{'\s\)|;-\)+|;\)+|\*-\)+|\*\)+|;-\]+|;\]+|;D+|
;\^\)+|:-,+|>:P+|:-P+|:P+|X-P+|x-p+|xp+|XP+|:-p+|:p+|
=p+|:-b+|:b+|d:+|>:\\+|>:/+|:-/+|:-\.+|:/+|:\\+|=/+|
=\\+|:L+|=L+|:S+|>\.<+|:\|+|:-\|+|:\$+|:-X+|:X+|:-#+|
:#+|O:-\)+|0:-3+|0:3+|0:-\)+|0:\)+|>:\)+|>;\)+|>:-\)+|
\|;-\)+|\|-O+|%-\)  +|%\)+|<3+|</3+)+"
```

Lisaks kirjutati kaks funktsiooni, mida küll selles analüüsis ei kasutata, kuid tulevikus on võimalik nende abil uurida näiteks ühepikkuselisi postitusi või uurida märke ühe postituse lõikes jms. Esimene funktsioon arvutab kokku, mitu tühikut on ühes postituses. Selle järgi saab leida, mitu sõna ühes postituses on. Teine funktsioon arvutab kokku, mitu tähte ja numbrit on ühes postituses. Välja jäetakse märgid ja tühikud.

5. Netikeele metagraafia analüüs

Analüüsiti 77 erineva kommentaariumi tekste, 64 foorumi postitusi ja 94 erineva jututoa vestlusi. Igat postitust uuriti eraldi. Kirjeldavat statistikat koguti kõikide internetikeskkondade postitustest ning vastused saadi automaatselt 13 esitatud küsimusele. 11 esimest küsimust koos tulemustega on välja toodud tabelis 5.

Tabel 5. **Statistika.**

		Kommen-taariumid	Foorumid	Jututoad
1.	Mitu postitust oli kokku?	36 645	382 146	498 897
		100%	100%	100%
2.	Mitmes postituses kasutati suuri tähti?	32 131	300 040	82 766
		87,68%	78,51%	16,59%
3.	Mitmes postituses leidus kirjavahemärkide ees tühikuid?	3668	15 980	11 026
		10,01%	4,18%	2,21%
4.	Mitmes postituses leidus kirjavahemärkide taga tühikuid?	34 031	219 971	41 829
		92,87%	57,56%	8,38%
5.	Mitmes postituses kasutati korrektsele lausele omast mustrit – tekstile järgneb lauselõpumärk ning sellele järgneb peale tühikut suurtäht?	17 449	113 445	690
		47,62%	29,69%	0,14%
6.	Mitmes postituses oli mõni uus rida, mis algas väikse tähega?	8626	105 161	472 821
		23,54%	27,52%	94,77%
7.	Mitmes postituses oli mõni uus rida, mis algas suurtähega?	20 949	234 386	35 385
		57,17%	61,33%	7,09%
8.	Mitmes postituses oli peale teksti mõni emotikon, millele järgnes suurtäht?	88	3939	34
		0,24%	1,03%	<0,01%
9.	Mitmes postituses oli peale teksti mõni emotikon, millele järgnes väike täht?	120	2342	1006
		0,33%	0,61%	0,20%
10.	Mimes postituses esinesid emotikonid üksikult?	31	597	18 384
		0,08%	0,16%	3,68%
11.	Mitu postitust leidus, kus emotikon oli tekstijupi või postituse lõpus?	39	13 921	25 822
		0,11%	3,64%	5,18%

Kaks viimast küsimust, mitu tühikut oli postituses ja mitu tähtnumbrit oli postituses, jäid selles analüüsis kõrvale. Tulevikus saaks neid küsimusi kasutada võrdlemaks märkide kasutust ühepikkuste postituste vahel. Näiteks võib uurida, mitmes *n*-sõnalises postituses oli *m* sellist mustrit, kus emotikonile järgnes suurtäht jms.

Kokku oli 36 645 kommentaariumi, 382 146 foorumi postitust ja 498 897 jututoa postitust. Kommentaariumides oli 4514 postitust ehk u 12% kõikidest kommentaaridest, kus ei kasutatud suuri tähti, foorumites oli selliseid 82 106 ehk u 22% foorumi postitustest ning jututubades 416 131 ehk u 83% kasutajate sisestatud tekstidest. Et jututubades pole u 83%-l postitustest kasutatud suurtähti näitab, et sünkroonses internetikeskkonnas tõepoolest pööratakse vähem tähelepanu sellele, kuidas ja millal kasutada suurtähti.

Emotikone kasutatakse tihtipeale lauselõpumärkidena (vt Provine jt 2007). Selles analüüsis uuriti, kas potentsiaalselt võiks leiduda selliseid lauseid, kus ühe lause lõpetab emotikon ja järgmine lause algab suurtähega. Seega otsiti selliseid mustreid, kus tekstile järgnes emotikon ning emotikonile järgnes suurtäht. Kommentaariumides leidus 88 (0,24%) postituses selliseid kombinatsioone, kokku oli selliseid kombinatsioone 92. Foorumites oli 3939 (1,03%) sellist postitust, kokku 4135 kombinatsiooni ning jututubades 34 (<0,01%) postituses, kus igas postituses esines sellist mustrit üks kord. Võrreldes protsentjaotusi, siis võib öelda, et kõige enam oli foorumites selliseid postitusi, kus esines vähemalt üks kord selline muster, mis võib näidata, et emotikoniga lõpetatakse lauset. Kõikidest postitustest esines selliseid vaid 1,03%-l, kuid arvuline näitaja, 3939, on siiski küllaltki suur. Et jututubades esines selliseid mustreid vaid alla 0,01%-i ei ole imekspandav, sest tavaliselt kasutatakse jututubades lühikesi lauseid või fraase, mis on kiirsõnumitele (*Instant Messaging*) omased (vt Zhou, Zhang 2005, Oja 2006), seega on igati loomulik, et jututubades ei esine väga palju mitmelauselisi postitusi.

Netikeele kasutajad aga ei tarvita suurtähti alati seal, kus tavaliselt ortograafiareeglid nõuavad nende kasutamist. Selle tõttu uuriti ka selle bakalaureusetöö uurimuses võrdluseks, mitmes postituses esines mustreid, mis erinevad eelmisest selle poolest, et peale emotikoni alustatakse teksti kirjutamist väikese tähega. Varasematest uurimustest on tehtud järeldusi, et emotikon ei lõhu pea iial fraase (vt Provine jt 2007), seega võib oletada, et suurel osal juhtudest asuvad emotikonid kas lause alguses, lõpus või fraasipiiril. Kommentaariumides oli 120 (0,33%) postituses selliseid mustreid (kokku leidus selliseid mustreid 134), foorumites 2342 (0,61%) postituses (kokku 2489) ning jututubades 18 384 (3,69%) postitust (kokku 18 384).

Emotikonid võivad küll lõpetada netikeeles lauseid, kuid kirjakeeles peetakse lauselõpumärkideks siiski hüüu-, küsimärki või punkti. Ka selles analüüsis uuriti, kas selliseid kombinatsioone leidus postitustes, mis vastavad klassikalisele lausele. Otsiti kombinatsioone, kus tekstile järgneb lauselõpumärk, sellele järgneb tühik ja suurtäht. Kommentaariumides esines 17 449 (47,62%) postituses selliseid mustreid (kokku 57 011), foorumites 113 445 (29,69%) postituses (kokku 227 994), jututubades 690 (0,14%) postituses (kokku 848).

5.1. Spontaansed reeglid

Analüüsi käigus selgus, et pigem kasutatakse suurtähti kommentaarides (u 88%) ja foorumite postitustes (u 79%) kui jututubades (u 17%). Sellised tulemused on kooskõlas teooriaga: on leitud (vt ptk 3), et jututoas on netikeel üldiselt lohakam, kuna on tarvis kiirelt oma sõnum edasi anda. Kiiruse mõttes on peale suurtähtede ärajätu tihtipeale jäetud kustutamata ka liigsed märgid, nagu näiteks tühikud:

> Mis Int-Lambotisse **puutub ,siis** 13 milj. eest ersatsi oli ilmselt vaja T.H.ilvese vanemate kodu taastamiseks, kurtis ju viimane, et mis riik see selline **on ,kus** isegi ministri palgast ei piisa isakodu kordategemiseks. RP liikmele B.Vaherile k tahaks mainida, et silmakirjalikumat erakonda kui RP on siin raske leida, just nemad

```
käisid kirikus valimiseelset vannet andmas, selles samas kirikus , mis
aastasadade jooksul põlishiisi hävitas. Endal on küll kavas tammik
rajada ning H.Kingole ütleks C.R.Jakobosoni sõnadega "... kõik
muistsed tuuled veel pole kadund säält"
```

Siinses analüüsis leidus aga vaid u 10%-s kommentaaridest, u 4%-s foorumi postitustest ja u 2%-s jututubade postitustest selliseid olukordi, kus punkti, küsi- ja hüüumärgi või koma ette oli jäänud tühik. Kuigi võiks arvata, et just jututubades on jäetud sisse üleliigsed märgid, siis statistika seda ei näita. Põhjuseks võib olla see, et üleüldiselt kasutatakse jututubades vähem kirjavahemärke. Seevastu oli tekstijuppe, mis sarnanesid ortograafiliselt õigetele lausetele selle poolest, et peale teksti oli punkt, küsi- või hüüumärk (kusjuures tühikut ei olnud ette jäänud) ning sellele järgnes tühik ning suur algustäht. Selliseid mustreid leidus u 48%-s kommentaaridest, u 30%-s foorumite postitustest ning vaid u 0,14%-s jututubade tekstidest.

Vaadates analüüsitud postitusi käsitsi, selgus konteksti kaudu, et tihtipeale märgib lause lõppu reavahetusklahvi *enter* vajutamine, kusjuures *enter*it võib olla vajutatud mitu korda. Sellistel juhtudel ei pruugi kasutajad lauselõpumärki kasutadagi, kuna järgmine lause algab uuelt realt. Selle tõttu ei pruugi ka uus lause suure tähega alata. Eriti on sellised mustrid nähtavad jutututbades, nt:

```
näen et meil tekkis veel yks vestluskaaslane
```

Kuid ka foorumitest võib leida selliseid näiteid:

```
See on jahh see offspring jms.
ja artikkel the viitab jahh ainulaadsusele

ja kui iga kuradi persekukkund sysadmin ei kögiseks iga asja peale siis
saaks ehk sellest asjast ka kõik aru.
```

Programmi tuleks tulevikus täiendada, et saaks täpsustada reavahetuse funktsiooni. Selles analüüsis leiti küll mustreid, kus otsiti nii väikse kui ka suure tähega algavaid uusi ridu, kuid pole võimalik öelda, kas nendele ridadele eelnenud laused olid lõpetatud lauselõpumärkidega või mitte. Mustreid, kus uus rida algas väikse tähega, leidus 23,54%-s kommentaaridest, 27,52%-s foorumite postitustest ning 83,97%-s jututubade

tekstidest. Mustreid, kus uus rida algas suurtähega, leidus 57,17%-s kommentaaridest, 61,33%-s foorumite postitustest ning vaid 7,09%-s jututubade tekstidest.

Huvitav oli ka asjaolu, et emotikone ei leidunud väga teksti keskel. Uuriti mitmes postituses leidus mustreid, kus tekstile järgneb emotikon ning emotikonile suurtäht:

> Olen nõus tasuta omandama :) Kunagi jõin nii tee- kui piimaseene poolt töödeldud vedelikke, piimaseenega oli rohkem tüli ning töödeldud tee maitses ka tunduvalt paremini. Mälestused, võiks teeseene küll "üles soojendada".

Ning mitmes postituses leidus mustreid, kus tekstile järgneb emotikon ning emotikonile väike täht:

> Vaata,väikeriik saab olla sõber vaid kaugel oleva suurriigiga,:)) või suudad tuua vastupidiseid näiteid?

Kommentaariumides oli selliseid postitusi vastavalt 0,24% ja 0,33%, foorumites 1,03% ja 0,61% ning jututubades 0,01% ja 0,20%. Emotikone esines üksikult, ilma tekstita, samuti vähestel juhtudel: kommentaariumides 0,08%, foorumites 0,16% ja jututubades 3,68% kõikidest kordadest. Ka ei olnud väga palju selliseid postitusi, nagu järgnev näide, kus emotikon oleks lõpetanud tekstijupi või postituse:

> Kuu aega tagasi alles oli ja juba jälle :(
>
> Kas mingi ravi kiirendab selle möödumist? Kergendada on ilmselt võimalik. Aga kiirendada?

Selliseid kombinatsioone esines 0,11%-s kommentaaridest, 3,64%-s foorumite postitustest ja 5,18%-s jututubadest.

5.2. Ettepanekuid netikeele analüüsiks

Võttes arvesse selle bakalaureusetöö empiirilise osa analüüsi tulemusi, tehakse ettepanekuid tulevastele netikeele tekstitöötlejatele. Eelnevat üldistades tuleks arvestada sellega, et jututubades kasutatakse suurtähti harva, seevastu kommentaariumides ja foorumites on üle 3/4 postitustest kasutusel ka suurtähed. Kirjavahemärkide kasutuses

ei ole võimalik praeguse analüüsi tulemuse järgselt kindlaks teha, millised spontaansed reeglid on tekkinud. Küll aga võib ära märkida, et kommentaariumides ja foorumites oli kirjavahemärkidel suurem tähtsus kui jututubades. Ka emotikonid käitusid kirjavahemärkidena esinedes tekstisiseselt just kommentaariumides ja foorumites tihedamini kui jututubades. Seevastu jututubades esines rohkem üksikuid emotikone ning rohkem postitusi lõppes emotikoniga kui kommentaariumides või foorumites.

Tulevikus võiks programme veelgi täiendada ning täpsustada mõningaid otsinguid. Näiteks võiks leida mooduse, kuidas vahet teha uutel ridadel, mis on automaatselt sisestatud ja neil, mis on kasutajate sisestatud. Samade programmide abil saab ka võrrelda samu mustreid postitustega, mis erinevad üksteisest pikkuse poolest. Näiteks saab otsida ühesugust informatsiooni kõikide 6–10sõnaliste postituste hulgast, kõikide 11–15sõnaliste postituste hulgast jne.

Edaspidi võiks koguda ka uusi tekste, sest kümne aasta jooksul võib olla toimunud muutusi nii keelekasutuses kui ka tarkvaras. Ühtlasi võiks tekste töödelda otsast peale ilma kõrvalise eeltöötluseta. See tagaks tekstitöötluse ühtlase stiili ja vähendaks veaohtlikke olukordi.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli anda ülevaade metagraafiast ja netikeelest ning nende rakendusest erinevates valdkondades. Teiseks eesmärgiks oli luua automaatne töövahend programmeeimiskeele Python abil, mida kasutada netikeelt kasutavate keskkondade postituste analüüsimisel.

Netikeele metagraafial on tähtis koht keeletöötluses. Ometi on netikeele uurimustest see teema tihtipeale välja jäetud. Netikeele metagraafias puudub üldine teoreetiline alus, millele arvutilingvistiline lähenemine võiks toetuda. Kui ilukirjanduslikes tekstides näitab lause algust ja lõppu suurtähed ja punktuatsioon ning nende järgi orienteerub ka keeletöötlusprogramm, siis netikeeles küll kasutatakse punktsiooni ja teisi märgistusi, kuid nende kasutust ei kontrollita ning ühtki ettekirjutust ei järgita. Teadmata, kuidas lauseid moodustatakse erinevates internetikeskkondades, pole võimalik rakendada ka korrektselt autoomaatseid keeletöötlusprogramme.

Metagraafia on muutuv nähtus. Teadaolevalt esimene tekstisisese märgistuse kasutus on pärit Vana-Kreekast, sellest ajast on metagraafia läbinud erinevaid arenguetappe ning alles renessansist pärinevad esimesed sõnastatud vahemärgireeglid. Paljud metagraafia märgid on unustatud, kuid nii mõnigi märk on netikeeles uuesti kasutusele tulnud. Töös vaadeldi üht 17. sajandist pärit teksti. Ilmnes, et tänapäeval kasutusel olev netikeel ja selle metagraafia sarnaneb mõnes mõttes palju enam trükikunstieelse teksti kui mõne ilukirjandusliku tekstiga. Siiski tuleb punktuatsioonis esile üks suur erinevus – emotikonide kasutus netikeeles. Emotikonid käituvad lauseis tihtipeale kirjavahemärkidena. Toetudes eelnevatele uurimustele samal teemal, käsitleti ka selles bakalaureusetöös emotikone metagraafia osana.

Mõistmaks kuidas netikeeles on märgitud lause algust ja lõppu, uuriti selle bakalaureusetöö empiirilises osas eestikeelsete internetikeskkondade tekste. Vaatluse all olid aastatest 2002–2004 pärit kommentaariumide, foorumite ja jututubade postitused. Analüüsi teostamiseks loodi Pythonis programm, mis otsis vastuseid küsimustele, mis puudutasid metagraafiat aga ka suurtähtede kasutust. Uuriti erinevate mustrite esinemist postitustes ning tulemused väljastati enamasti selle kohta, et mitmes postituses esines teatud mustrit. Tulemuste kohta anti nii üldarv kui ka protsentjaotus vaadeldava netikeskkonna lõikes. Vastuseid otsiti küsimustele nagu mitmes postituses kasutati suurtähti, mitmes leidus kirjavahemärkide ees tühikuid, mitmes järel, mitmes postituses oli märgata kirjakeelele omast punktuatsiooni lause lõpus, mitemes postituses märkis uue lause algust uus rida ja sealjuures mitmed nendest juhtudest algas suurähega mitu väikse tähega, mitmes postituses märkis lause lõppu emotkon ja sealjuures mitmed nendest juhtudest algas järgnev tekst suurtähega, mitu väikese tähega, mitmes postituses esinesid emotikonid üksikult ning mitu postitust leidus, kus emotikon oli tekstijupi või postituse lõpus.

Tulemustest selgus, et suurtähti kasutati enam kommentaarides ja foorumite postitustes ning vähem jututubades. Ka teooriaosas oli välja toodud märge selle kohta, et sünkroonsetes netikeskkondades on kiirus tähtis ja seetõttu ei pöörata väga tähelepanu suurtähtede kasutusele ja ka muud vead jäävad tihtipeale sisse. Sellest johtuvalt uuriti, kui tihti on jäetud kustutamata tühikud punktuatsiooni ees. Selliseid juhtumeid leidus kõige rohkem kommentaarides ja kõige vähem jututubades. Tekstijuppe, mis lõppesid lauselõpumärgiga ning millele järgnes tühik ja suurtäht leidus enim kommentaarides ja kõige vähem jututubades. Probleemseks osutusid sellised tekstijupid, kus oli *enteri* klahvi kasutatud tähistamaks lause lõppu. Kuigi leiti kui paljudes postitustes järgnes sellisele tavale suurtäht ja kui paljudes väike täht, siis oli võimatu öelda, kuidas eelnenud tekstijupp lõppes. Emotikonide kasutamine teksti keskel ei olnud populaarne. Siiski leidus juhtumeid, kus peale emotikoni järgnes tekst. Uuriti kas järgnenud tekst algas suure või väikese algustähega. Ka üksikult esinevaid emotikone leidus vähestel

juhtudel ning ei olnud ka väga palju postitusi, kus emotikon oleks lõpetanud tekstijupi või postituse.

Töö lõpus soovitatakse kokkuvõtvalt arvestada sellega, et punktuatsiooni kasutuses ei ole võimalik selle bakalaureusetöö empiirilise osa analüüsi põhjal anda selgeid eeskirju netikeele metagraafia kasutuse kohta. Küll aga on see töö heaks aluseks edaspidises analüüsis ja uurimises samal teemal. Tulevikus tuleb eeskätt programme täiustada ning tekste koguda hilisemast ajast. Ühtlasi soovitatakse tekste töödelda toormaterjali kasutades, mitte töötada eeltöödeldud materjaliga, vältimaks erinevate meetodite ja stiilide lõimumist.

Kirjandus

Allman, William F. 2012. The Accidental History of the @ Symbol. - Smithsonian Magazine, September; **http://www.smithsonianmag.com/science-nature/the-accidental-history-of-the-symbol-18054936/?no-ist**=. Vaadatud 14.05.2014

Baym, Nancy 1995. The emergence of community in computer-mediated communication. – CyberSociety. Computer-Mediated Communication and Community, lk 138–163.

Cherny, Lynn 1999. Conversation and community: Chat in a virtual world. Stanford: Center for the Study of Language and Information.

Crystal, David 2001. Language and the Internet. Cambridge University Press.

Dresner, Eli, Susan C. Herring 2010. Functions of Nonverbal in CMC: Emoticons and Illocutionary Force. – Communication theory, nr 20/3, lk 249–268.

EKG II = Erelt, Tiiu, Ülle Viks, Mati Erelt, Reet Kasik, Helle Metslang, Henno Rajandi, Kristiina Ross, Henn Saari, Kaja Tael, Silvi Vare 1993. Eesti keele grammatika II. Süntaks. Lisa: kiri. Toim. M. Erelt, T. Erelt, H. Saari, Ü. Viks. Tallinn: Eesti Teaduste Akadeemia Keele ja Kirjanduse Instituut.

EKK = Erelt, Mati, Tiiu Erelt, Kristiina Ross 2000. Eesti keele käsiraamat. Teine, täiendatud trükk. Tallinn: Eesti Keele Sihtasutus.

Hammond, Michael 2003. Programming for Linguists: Perl for language researchers. Blackwell Publishing.

Hennoste, Tiit 2013. kuule, ma eemale nüüd. – Sirp, nr 46; http://www.sirp.ee/index.phpoption=com_content&view=article&id=20202:2013-12-05-17-05-50&catid=9:sotsiaalia&Itemid=13&issue=3468. Vaadatud 18.12.2014.

Herring, Susan C. 2007. A faceted classification scheme for computer-madiated discourse. – Language@Internet, nr 4; http://www.languageatinternet.org/articles/2007/761. Vaadatud 10.01.2014.

HL = History and License – Python v2.7.6. documentation; https://docs.python.org/2/license.html. Vaadatud 21.04.2014.

Houston, Keith 2013. The Ancient Roots of Punctuation. – The New Yorker; http://www.newyorker.com/online/blogs/books/2013/09/origins-of-hashtag-manicule-diple-pilcrow-ampersand-explained.html. Vaadatud 20.01.2014.

Jones, Bernard 1996. Towards a Syntactic Account of Punctuation. – '96 Proceedings of the 16th conference on Computational linguistics, nr 2, lk 604–609; http://acl.ldc.upenn.edu/C/C96/C96-2102.pdf. Vaadatud 19.01.2014.

Jurafsky, Daniel, James H. Martin 2000. Speech and Language Processing. Prentice-Hall, Inc.

King, Brian 2009. Building and Analysing Corpora of Computer-Mediated Communication. – Contemporary Corpus Linguistics, lk 301–321.

Levinson, Joan 1986. Punctuation and the Ortographic Sentence: a Linguistic Analysis. City University of New York dissertation.

Maynor, Natalie 1994. The language of electronic mail: Written speech? – Publications of the American Dialect Society Series.

Nunberg, Geoffrey 1990. The Linguistics of Punctuation. Center for the Study of Language and Information: Leland Stanford Junior University.

Nunberg, Geoffrey 2008. All Thumbs. - Fresh Air commentary; http://people.ischool.berkeley.edu/~nunberg/texting.html. Vaadatud 18.02.2014.

Oja, Anni 2009. Delfi kommentaaride keelest. – Sirp, nr 47; http://www.sirp.ee/index.phpoption=com_content&view=article&id=9908:delfi-kommentaaride-keelest&catid=18:varamu&Itemid=15&issue=3279. Vaadatud 29.11.2013.

Oja, Anni 2006. Eesti keel internetis. Keel ja arvuti, lk 259–267. Tartu: Tartu Ülikooli kirjastus.

Provine, Robert R., Robert J. Spencer, Darcy L. Mandell 2007. Emotional Expressions Online. – Journal of Language and Psychology, 26/3; http://jls.sagepub.com/content/26/3/299.abstract. Vaadatud 01.12.2013.

Rummo, Tiit 1995. Käitumine Internetis. – Arvutimaailm nr 2, lk 44–47.

Swackhamer, Conrad 1848. Influence of the Telegraph Upon Literature. – The United States Democratic Review, nr 22.

The Python Standard Library; https://docs.python.org/2/library/re.html. Vaadatud 21.04.2014.

Welcome to Python; www.python.org. Vaadatud 08.04.2014

Zhou, Lina, Dongsong Zhang 2005. A Heuristic Approach to Establishing Punctuation. - IEEE Transactions on Professional Communication nr 48/4, lk 391–400.

The Punctuation of Internet Language

Summary

The Internet as a whole is a diverse environment with millions of users from all over the world. Even though in the forums and chatrooms no strict grammar rules are enforced, over time spontaneous language patterns or so called *netspeak* have emerged. In particular, the special use of punctuation, while being easy to read for humans, makes automatic analysis of such texts a challenging task for computers. The fields of Computational Linguistics and Language Technology are especially interested in solving the problem of automatic processing of computer-mediated texts.

By *punctuation* we mean all of the individual non-alphanumeric symbols and compounds of, such as ., !, ?, @, #, as well as compounds of these symbols, such as :), :(, :P etc. The purpose of this study was to investigate how punctuation is used in the netspeak and to provide an approach for theoretical treatment of Internet language in the framework of Natural Language Processing (NLP).

The first part of the thesis gives an overview of the punctuation and symbols theory. We examine the history and compare the usage of punctuation in the Early Modern English and the present-day Internet language. In addition, we discuss the mediums and characteristics of the Internet language. A special emphasis is given on a particular type of compound symbols called emoticons that are used in general for expressing emotions and feelings.

The second part of the thesis introduces the data and methods for analysing Internet language. We discuss the importance of programming in linguistics and give a short

overview of the programming language Python with special attention on its regular expressions module. Finally, in the practical data analysis part, we present the results of Internet language analysis and conclude with some ideas for future work.

To understand the nature of punctuation in Internet language, we analysed texts from three different Estonian computer mediums. Our dataset consisted of posts from commentaries (77 different subjects), forums (64 subjects) and chatrooms (92 chats) from 2002 to 2004. A program was written in Python to describe the statistics and analyse the usage of punctuation and capital letters.

Among many other combinations, for instance, we studied how often there was a space in front of an ending mark (./!/?) and the occurrences of a punctuation mark followed by space that is followed by a capital letter. Both of these cases appeared more frequently in commentaries and the least in chatrooms. In addition, we observed in the data analysis that capital letters were more frequently used in commentaries and forums and less so in chatrooms. This agrees well with the theory that in synchronous computer-mediated communication speed is of high importance. Therefore, the users tend to ignore the usage of capital letters and errors are often left uncorrected. Also, the usage of emoticons was of high interest because they are often treated as punctuation marks.

In the present era of information written content is being constantly produced at an ever growing pace. For future work, we should collect new up to date dataset as more than 10 year old language patterns might not be relevant anymore in the context of the present day. By using the found patterns we could develop new probabilistic language models for automatising text processing and ideally making the Internet language comprehensible for computers.

Lisad

Lisa 1. Kommentaariumide tekstitöötlusprogramm

```
#!/usr/bin/python
# Kommentaarium
# See programm loob listi, milles igaks
# elemendiks saab yhe inimese postitus.


import os
import re
from functions import *

# find data directory
DATA_DIR = "/home/kristiina/Desktop/kommentaar/"
# create new list
super_array = []
# for every file in folder
# read data into one array (super_array)
for file in os.listdir(DATA_DIR):
    super_array.append(parse_file_kommentaar(DATA_DIR + file))


data = []
for f in super_array:
    for i in range(0, len(f)):
        vahelist = []
        # 0.
        vahelist.append(f[i])
        # 1. are capital letters used in the beginnings?
        vahelist.append(parse_capital(f[i]))
        # 2. are there white spaces before punctuation? (esimene juhtum)
        vahelist.append(parse_frontspace(f[i]))
        # 3. are there white spaces after punctuation?
        vahelist.append(parse_endspace(f[i]))
        # 4. does the sentence end with punctuation, white space and next start with
capital?
        vahelist.append(parse_classic(f[i]))
        # 5. are there any small letters after newlines?
        vahelist.append(parse_newlinesmall(f[i]))
        # 6. are there any capital letters after newlines?
        vahelist.append(parse_newlinecapital(f[i]))
        # 7. are there emoticons in the end of the sentences? next sentence starts with
```

```
capital.
        vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[A-Z]"))
        # 8. after emoticons small letters
        vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[a-z]"))
        # 9. single emoticon
        vahelist.append(parse_emoticon("(\n|^)\s?", f[i], "(\n|$)"))
        # 10. emoticon is in the end of text
        vahelist.append(parse_emoticon("[A-Za-z0-9]\s", f[i], "(\n|$)"))
        # 11. counts how many spaces in text (different places to insert punctuation)
        vahelist.append(parse_spacecount(f[i]))
        # 12. counts how many letters (or numbers) in text
        vahelist.append(parse_lettercount(f[i]))
        # 13. find all emoticons from post
        vahelist.append(parse_emoticon("\w*.*\s*", f[i], "\w*.*\s*", re.DOTALL))
        data.append(vahelist)

#print len(data)

#count0 = 0

count1 = 0
count2 = 0
count3 = 0
count4 = 0
count5 = 0
count6 = 0
count7 = 0
count8 = 0
count9 = 0
count10 = 0

for d in data:
    if d[1] > 0:
        count1 += 1
    if d[2] > 0:
        count2 += 1
    if d[3] > 0:
        count3 += 1
    if d[4] > 0:
        count4 += 1
    if d[5] > 0:
        count5 += 1
    if d[6] > 0:
        count6 += 1
    if d[7] > 0:
        count7 += 1
    if d[8] > 0:
        count8 += 1
    if d[9] > 0:
        count9 += 1
    if d[10] > 0:
```

```
        count10 += 1

#how many posts had n emoticons in it?

# for d in data:
#   if d[1] == 0:
#       count0 += 1
#   if d[1] == 1:
#       count1 += 1
#   if d[1] == 2:
#       count2 += 1
#   if d[1] == 3:
#       count3 += 1
#   if d[1] == 4:
#       count4 += 1
#   if d[1] == 5:
#       count5 += 1
#   if d[1] == 6:
#       count6 += 1
#   if d[1] == 7:
#       count7 += 1
#   if d[1] == 8:
#       count8 += 1
#   if d[1] == 9:
#       count9 += 1
#   if d[1] > 10:
#       count10 += 1

#print count0
print count1
print count2
print count3
print count4
print count5
print count6
print count7
print count8
print count9
print count10

# count = 0
# for d in data:
#   count += d[1]
# print count
```

Lisa 2. Foorumite tekstittötlusprogramm

```
#!/usr/bin/python
# Foorum
# See programm loob listi, milles igaks
# elemendiks saab yhe inimese postitus.


import os
import re
from functions import *

# find data directory
DATA_DIR = "/home/kristiina/Desktop/foorum/"
# create new list
super_array = []
# for every file in folder
# read data into one array (super_array)
for file in os.listdir(DATA_DIR):
    super_array.append(parse_file_foorum(DATA_DIR + file))

data = []
for f in super_array:
    for i in range(0, len(f)):
        vahelist = []
        # 0.
        vahelist.append(f[i])
        # 1. are capital letters used in the beginnings?
        #vahelist.append(parse_capital(f[i]))
        # 2. are there white spaces before punctuation? (esimene juhtum)
        #vahelist.append(parse_frontspace(f[i]))
        # 3. are there white spaces after punctuation?
        #vahelist.append(parse_endspace(f[i]))
        # 4. does the sentence end with punctuation, white space and next start with
capital?
        #vahelist.append(parse_classic(f[i]))
        # 5. are there any small letters after newlines?
        vahelist.append(parse_newlinesmall(f[i]))
        # 6. are there any capital letters after newlines?
        vahelist.append(parse_newlinecapital(f[i]))
        # 7. are there emoticons in the end of the sentences? next sentence starts with
capital.
        #vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[A-Z]"))
        # 8. after emoticons small letters
        #vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[a-z]"))
        # 9. single emoticon
        #vahelist.append(parse_emoticon("(\n|^)\s?", f[i], "(\n|$)"))
        # 10. emoticon ends the sentence
```

```
        #vahelist.append(parse_emoticon("[A-Za-z0-9]\s", f[i], "(\n|$)"))
        # 11. counts how many spaces in text (different places to insert punctuation)
        #vahelist.append(parse_spacecount(f[i]))
        # 12. counts how many letters (or numbers) in text
        #vahelist.append(parse_lettercount(f[i]))
        # 13. find all emoticons from post
        #vahelist.append(parse_emoticon("", f[i], ""))
        data.append(vahelist)

#print len(data)

#count0 = 0
count1 = 0
count2 = 0
#count3 = 0
#count4 = 0
#count5 = 0
# count6 = 0
# count7 = 0
# count8 = 0
# count9 = 0
# count10 = 0

for d in data:
    if d[1] > 0:
        count1 += 1
    if d[2] > 0:
        count2 += 1
    # if d[3] > 0:
    #   count3 += 1
    # if d[4] > 0:
    #   count4 += 1
    # if d[5] > 0:
    #   count5 += 1
    # if d[6] > 0:
    #   count6 += 1
    # if d[7] > 0:
    #   count7 += 1
    # if d[8] > 0:
    #   count8 += 1
    # if d[9] > 0:
    #   count9 += 1
    # if d[10] > 0:
    #   count10 += 1

#how many posts had n emoticons in it?
"""
for d in data:
    if d[1] == 0:
        count0 += 1
    if d[1] == 1:
```

```
            count1 += 1
        if d[1] == 2:
            count2 += 1
        if d[1] == 3:
            count3 += 1
        if d[1] == 4:
            count4 += 1
        if d[1] == 5:
            count5 += 1
        if d[1] == 6:
            count6 += 1
        if d[1] == 7:
            count7 += 1
        if d[1] == 8:
            count8 += 1
        if d[1] == 9:
            count9 += 1
        if d[1] > 10:
            count10 += 1
"""
#print count0
print count1
print count2
# print count3
# print count4
# print count5
# print count6
# print count7
# print count8
# print count9
# print count10

# count = 0
# for d in data:
#   count += d[1]
# print count
```

Lisa 3. Jututubade tekstitöötlusprogramm

```
#!/usr/bin/python
# Jututuba
# See programm loob listi, milles igaks
# elemendiks saab yhe inimese postitus.


import os
import re
from functions import *
# find data directory
DATA_DIR = "/home/kristiina/Desktop/jutukas/"
# create new list
super_array = []
# for every file in folder
# read data into one array (super_array)
for dir in os.listdir(DATA_DIR):
    for file in os.listdir(DATA_DIR + dir):
        super_array.append(parse_file_jutukas(DATA_DIR + dir + "/" + file))

data = []
for f in super_array:
    for i in range(0, len(f)):
        vahelist = []
        # 0.
        vahelist.append(f[i])
        # 1. are capital letters used in the beginnings?
        #vahelist.append(parse_capital(f[i]))
        # 2. are there white spaces before punctuation? (esimene juhtum)
        #vahelist.append(parse_frontspace(f[i]))
        # 3. are there white spaces after punctuation?
        #vahelist.append(parse_endspace(f[i]))
        # 4. does the sentence end with punctuation, white space and next start with
capital?
        #vahelist.append(parse_classic(f[i]))
        # 5. are there any small letters after newlines?
        vahelist.append(parse_newlinesmall(f[i]))
        # 6. are there any capital letters after newlines?
        vahelist.append(parse_newlinecapital(f[i]))
        # 7. are there emoticons in the end of the sentences? next sentence starts with
capital.
        #vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[A-Z]"))
        # 8. after emoticons small letters
        #vahelist.append(parse_emoticon("[A-Za-z]\s", f[i], "\s[a-z]"))
        # 9. single emoticon
        #vahelist.append(parse_emoticon("(\n|^)\s?", f[i], "(\n|$)"))
```

```
        # 10. emoticon ends the sentence
        #vahelist.append(parse_emoticon("[A-Za-z0-9]\s", f[i], "(\n|$)"))
        # 11. counts how many spaces in text (different places to insert punctuation)
        #vahelist.append(parse_spacecount(f[i]))
        # 12. counts how many letters (or numbers) in text
        #vahelist.append(parse_lettercount(f[i]))
        # 13. find all emoticons from post
        #vahelist.append(parse_emoticon("\w*.*\s*", f[i], "\w*.*\s*"))
        data.append(vahelist)
        #for v in vahelist:
            #print v

#print len(data)

#count0 = 0

count1 = 0
count2 = 0
# count3 = 0
# count4 = 0
# count5 = 0
# count6 = 0
# count7 = 0
# count8 = 0
# count9 = 0
# count10 = 0
for d in data:
    if d[1] > 0:
        count1 += 1
    if d[2] > 0:
        count2 += 1
    # if d[3] > 0:
    #   count3 += 1
    # if d[4] > 0:
    #   count4 += 1
    # if d[5] > 0:
    #   count5 += 1
    # if d[6] > 0:
    #   count6 += 1
    # if d[7] > 0:
    #   count7 += 1
    # if d[8] > 0:
    #   count8 += 1
    # if d[9] > 0:
    #   count9 += 1
    # if d[10] > 0:
    #   count10 += 1

#how many posts had n emoticons in it?
"""
for d in data:
```

```
    if d[1] == 0:
        count0 += 1
    if d[1] == 1:
        count1 += 1
    if d[1] == 2:
        count2 += 1
    if d[1] == 3:
        count3 += 1
    if d[1] == 4:
        count4 += 1
    if d[1] == 5:
        count5 += 1
    if d[1] == 6:
        count6 += 1
    if d[1] == 7:
        count7 += 1
    if d[1] == 8:
        count8 += 1
    if d[1] == 9:
        count9 += 1
    if d[1] > 10:
        count10 += 1
"""
#print count0
print count1
print count2
# print count3
# print count4
# print count5
# print count6
# print count7
# print count8
# print count9
# print count10
"""
count = 0
for d in data:
    count += d[1]
print count
"""
```

Lisa 4. Funktsioonid

```
#!/usr/bin/python
import re

# find written text between tags
def parse_file_kommentaar(filename):
  #if you want to have parsed files in a new file, use commented commands
  #new_file = open("kommentaar_newfile.txt", "w+") # opening empty file for read+writing
  # opening original file for reading
  fd = open(filename, "r")
  data = fd.read()
  # find text between tags; find newlines using "." (re.DOTALL); save as x
  x = re.findall(r"</teema>(.*?)</kiri>", data, re.DOTALL)
  # for line in x:
    # write every element in x into new_file
  return x

# pick out sentences needed and add as elements to array x
def parse_file_foorum(filename):
    x = [] # empty list, every element will be users text
    fd = open(filename)# opening file
    my_string = "" # empty string for saving users text as element in R
    re_list = [re.compile("^\s*[<>-]"),# begins with space, <, > or -
               re.compile("^[\s>]"), # space as a line
               re.compile("^.*<.*>.*$"), # < or > in the middle of line
               re.compile(".*wrote:.*$"), # lines that include the word "wrote:"
               re.compile(".*@.*"), # e-mail addresses
               re.compile("^0.0.$"), # lines that include "0.0."
               re.compile("^[\s]*[\w\d\?\.,!]*[\s]*$")] # one word lines, 0 or more
spaces in the beginning and end
    for line in fd:
        line = line.rstrip('\r\n') # remove reavahetused
        if any(r.match(line) for r in re_list): # if at least one of the regex's matches
            if my_string != "": # if string is empty
                my_string = my_string.rstrip('\r\n')
                x.append(my_string)
                my_string = ""

        else:
            # add line to x and if next line doesn't mach regex,
            # add next line as the same element to x.
            my_string += " " + line
    return x

def parse_file_jutukas(filename):
  # opening original file for reading
```

```
  fd = open(filename, "r")
  data = fd.read()
  # find text between tags; find newlines using "." (re.DOTALL); save as x
  #x = re.findall(r"^\[[0-9]{2}:[0-9]{2}\]\s\<.*\>\s(.*)", data, re.DOTALL)
  x = re.findall(r"\[[0-9]{2}:[0-9]{2}:?[0-9]{0,2}\]\s<[^>]+>\s(.*)\r\n", data)
  return x

def parse_capital(textname):
  # are capital letters used at all?
  re_capital = re.findall(r"[A-Z]", textname)
  return len(re_capital)
  #if re_capital.match(textname):
  #  return textname

def parse_frontspace(textname):
  # white space before punctuation
  re_frontspace = re.findall(r"[A-Za-z0-9]+\s+[\.\?!,]+", textname)
  return len(re_frontspace)

def parse_endspace(textname):
  # white space after punctuation
  re_endspace = re.findall(r"[A-Za-z0-9]+[\.\?!,]+\s+", textname)
  return len(re_endspace)

def parse_classic(textname):
  # patterns that match classical sentences:
  # punctuation mark - space - capital letter
  re_classic = re.findall(r"[A-Za-z0-9]+[\.\?!]\s[A-Z]",textname)
  return len(re_classic)

def parse_newlinesmall(textname):
  # new line - small letter
  re_newlinesmall = re.findall(r"^\r?\n?\s?[a-z]", textname)
  return len(re_newlinesmall)

def parse_newlinecapital(textname):
  # new line - capital letter
  re_newlinecapital = re.findall(r"^\r?\n?\s?[A-Z]", textname)
  return len(re_newlinecapital)

def parse_emoticon(start, textname, ending):
  # emoticon as punctuation mark,
  # ending either:
  # emoticon + Capital letter,
  # emoticon + small letter or
  # emoticon + newline
  # emoticon is the last element of sentence
  re_emoticons = re.findall(start + get_emoticon_regex() + ending, textname, re.DOTALL)
  return len(re_emoticons)

def get_emoticon_regex():
```

```
  return r"(:-\)+|:\)+|:o\)+|:\]+|:3+|:c\)+|:>+|=\]+|8\)+|=\)+|:\}+|:\^\)
+|:-D+|:D+|8-D+|8D+|x-D+|xD+|X-D+|XD+|=-D+|=D+|=-3+|=3+|B\^D+|>:\[+|:-\(+|:\
(+|:-c+|:c+|:-<+|:<+|:-\[+|:\[+|:\{+|;\(+|:-\|\|+|:@+|>:\(+|:'-\(+|:'\(+|:'-\)+|:'\)
+|D:<+|D:+|D8+|D;+|D=+|DX+|v\.v+|D-':
+|>:O+|:-O+|:O+|:-o+|:o+|8-0+|O_O+|o-o+|O_o+|o_O+|o_o+|O-O+|:\*|:\^\*+|\(\s'\}\
{'\s\)|;-\)+|;\)+|\*-\)+|\*\)+|;-\]+|;\]+|;D+|;\^\)+|:-,
+|>:P+|:-P+|:P+|X-P+|x-p+|xp+|XP+|:-p+|:p+|=p+|:-b+|:b+|d:+|>:\\+|>:/+|:-/+|:-\.+|:/
+|:\\+|=/+|=\\+|:L+|=L+|:S+|>\.<+|:\|+|:-\|+|:\$+|:-X+|:X+|:-#+|:#+|O:-\)
+|0:-3+|0:3+|0:-\)+|0:\)+|>:\)+|>;\)+|>:-\)+|\|;-\)+|\|-O+|%-\)+|%\)+|<3+|</3+)+"

def parse_spacecount(textname):
  count = 0
  re_isspace = re.compile("\s+")
  for space in textname:
    if re_isspace.match(space):
      count += 1
  return count

def parse_lettercount(textname):
  count = 0
  re_isletter = re.compile("\w")
  for letter in textname:
    if re_isletter.match(letter):
      count += 1
  return count
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Kristiina Toots (sünnikuupäev: 03.11.1989)**

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

   „**Netikeele metagraafia“**, mille juhendaja on **vanemteadur Heiki-Jaan Kaalep**

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 16.05.2014