T A R T U Ü L I K O O L

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Informaatika eriala

Kadri Kajaste

Eestikeelsete tekstide morfoloogiline ühestamine

Magistritöö

Juhendaja: PhD Kaili Müürisep

Autor: …… “…..“ mai 2009

Juhendaja: ……. “…..“ mai 2009

Professor: …… “…..“ mai. 2009

TARTU 2009

Sisukord

Sissejuhatus....4
1.Tekstide morfosüntaktiline analüüs....6
1.1.Ühestamine....6
2.Ühestamise meetodid....7
2.1.Üldistest ühestamise meetoditest....7
2.2.Ühestamise meetodid eesti keele jaoks....8
3.Markovi mudeli märgendamine....10
3.1.Tõenäosuslik mudel....10
3.2.Viterbi algoritm....11
3.3.TnT- Statistiline sõnaliigi märgendaja....12
3.3.1.Arhitektuuri aluseks olev mudel....12
3.3.2.Silumine....13
3.3.3.Tundmatud sõnad....14
4.Tõenäosuslik sõnaliikide märgendamine otsustuspuu abil....16
4.1.Sissejuhatus....16
4.2.TreeTagger....17
4.2.1.Otsustuspuu moodustamine....17
4.2.2.Lihtsustatud algoritm....18
4.2.3.Otsustuspuu kärpimine....18
4.2.4.Leksikon....19
5.Kitsenduste grammatika....21
5.1.Ülevaade....21
5.2.Reeglite kuju ja tulemused....22
6.Ühestamine ja eesti keel....23
6.1.Hetkeolukord....23
6.2.TreeTagger....27
6.2.1.Treenimine....27
6.2.2.Märgendamine....28
6.3.TnTTagger....32
6.3.1.Failide formaadid....32
6.3.2.Parameetrite genereerimine: tnt-para....33
6.3.3.Märgendamine: tnt....34
7.Programmide rakendamine....36
7.1.Olukord....36
7.2.Töötlus....36
7.3.Treetagger....37
7.4.TnTTagger....39
8.Katsed....41
8.1.Katsed optimaalse märgendusskeemi leidmiseks....41
8.2.Katsed programmide erinevate seadistusvõimalustega....50
8.3.Statistiliste märgendajate kombineerimine reeglipõhisega....53
Järeldused ja edasiarendused....57
Kokkuvõte....58
Part-of-Speech Tagging of Estonian Language....59
Kirjandus....60

Sissejuhatus

Morfoloogiline analüüs ja ühestamine on kesksed probleemid keeletehnoloogilistes rakendustes. Kuna tegemist on kirjaliku keele kõige madalama taseme analüüsiga, kasutavad kõik kõrgemad tasemed, näiteks süntaktiline ja semantiline analüüs, morfoloogilist infot sisendina. Morfoloogiline ühestamine on vajalik võimalikult täpse ja õige tulemuse saamiseks edasise töötluse käigus. Teksti morfoloogilisel töötlemisel määratakse igale sõnale ja märgile lauses esmalt kõikvõimalikud sõnaliigi interpretatsioonid. Seejärel valitakse mitmest erinevast variandist välja just see õige, antud kontestis sobiv sõnaliik - sellist protsessi nimetataksegi morfoloogiliseks ühestamiseks. Kuna keeletöötluses on tegemist suurte tekstikorpustega, siis on vaja ka ühestamist arvutiprogrammide abil automatiseerida. Senini eesti keele jaoks kasutatud automaatsed ühestajad (Puolakainen, 2001; Kaalep,Vaino, 1998) ei ole kaugeltki täiuslikud ja töö eesmärgiks oligi katsetada erinevate meetodite sobivust eesti keelele.

Töös vaadeldakse katset rakendada H. Schmid-i poolt 1994 a. väljatöötatud programmi TreeTagger (Schmid, 1994) ja T. Brantsi poolt 2000. aastal loodud programmi TnTTagger (Brants, 2000) eestikeelsete tekstide automaatsel märgendamisel. TreeTagger töötab otsustuspuude läbimise meetodit rakendades ja TnTTagger kasutab teist järku Markovi mudeleid. Lisaks vaadeldakse töös ka katset ühendada kaks eelpool mainitud statistilist märgendajat reeglipõhise ühestajaga (Puolakainen, 2001). Töö põhiosa seisnes sobivate märgendajate väljavalimises, vajalike tekstide töötlemises märgendajatele sobivale kujule ja siis erinevate katsete teostamises, samuti mitme ühestaja kombineerimise algoritmi väljatöötamises.

Töö koosneb kaheksast peatükist. Esimeses ja teises peatükis antakse ülevaade morfoloogilisest ühestamisest ja erinevatest maailmas kasutatud ühestamismeetoditest ning eesti keelele seni rakendada püütud meetoditest. Kolmandas peatükis tutvustatakse täpsemalt Markovi mudelil põhinevat märgendamist ja TnTTaggeri tööpõhimõtteid. Neljandas peatükis tutvustatakse lähemalt otsustuspuude meetodit, mille alusel TreeTagger töötab. Viiendas peatükis antakse ülevaade kitsenduste grammatikal põhinevast eesti keelele koostatud reeglipõhisest ühestajast. Kuuendas peatükis selgitatakse statistiliste meetoditega ühestamiseks vajalike eestikeelsete korpuste olukorda ning eelpool toodud statistilistel meetoditel põhinevaid programme TreeTagger ja TnTTagger ja nende

kasutusvõimalusi. Seitsmendas peatükis kirjeldatakse tekstide teisendamist programmidele sobivale kujule ja viimases peatükis erinevaid läbiviidud katseid.

Lisana on esitatud ka töö erinevate etappide parimad tulemused (cd plaadil), teisendusprogrammid ja meetodite kombineerimise programm.

1. Tekstide morfosüntaktiline analüüs

1.1.Ühestamine

Tekstide automaatsel töötlusel ja analüüsil on mitu erinevat etappi. Esmalt on vaja tekstid töödelda kasutatavale programmile sobivale kujule, eraldada laused ja sõnad. Siis tuleb eelmise etapi tulemus morfoloogiliselt analüüsida. Sellele omakorda järgneb süntaktiline analüüs. Keele morfoloogilisel analüüsimisel leitakse kõikide tekstis esinenud sõnade ja märkide morfoloogiline informatsioon, s.t leitakse sõnaliik (kas tegemist on nimisõna, tegusõna või mõne muu variandiga), käänduvate sõnade korral arv ja kääne, pöörduvate sõnade korral pööre, aeg, arv, kõneviis jmt.. Loomulike keelte morfoloogilisel märgendamisel tekitavad probleeme sõnade mitmesused: programm ei suuda leida sõna algvormi,, kuna erinevate sõnade vormid langevad kokku. Kõige lihtsam näide oleks sõna „või"- inimesele on lihtne lause konteksti alusel aru saada, kas tegemist on toiduaine (nimisõna), sidesõna või verbivormiga. Kuid teatavasti programm inimese moodi mõelda ei suuda, samuti ei vaata morfoloogiline analüsaator sõna konteksti, vaid piirdub ainult sõnavormi endaga. Seega leiab morfoloogiline analüsaator kõik võimalikud sõnaliikide variandid, mis antud sõnakujule sobida võiksid. (Antud näites siis nimisõna, verb ja sidesõna, nimisõna on omakorda mitmene nimetava ja omastava käände vahel.) Morfoloogilise ühestamise eesmärk on sõna mitmete erinevate morfoloogiliste tõlgenduste vahel õige valimises, arvestades konteksti. Morfoloogiliste mitmesuste tekkeks on kaks võimalust: mitmesus seisneb kahe üheliigilise sõna erinevas tähenduses või muutevormis (nim *raha* - om *raha* - os *raha*), või teisel juhul on ühesuguse kirjapildi taga ka lisaks erinevale tähendusele veel erinev sõnaliik. Lihtsa näitena võiks tuua sõna „kuid", mis võib olla nii sidesõna kui ka nimisõna *kuu* mitmuse osastava vorm.

Teksti märgendamine on vajalik mitmete loomuliku keele töötluse ülesannete jaoks: näiteks grammatikakorrektori jaoks, süntaksianalüüsiks, informatsiooni eraldamiseks, küsimustele vastamiseks ja korpuse märgendamiseks(Loftsson, 2008).

2. Ühestamise meetodid

2.1.Üldistest ühestamise meetoditest

Maailmas on kõige enam uurimistööd tehtud inglise keele automaatse töötlemise vallas ja seega on ka erinevaid meetodeid arendatud eelkõige inglise keelele (või teistele indoeuroopa keeltele) mõeldes. Laialt on kasutuses statistikal põhinevad Markovi mudeli baasil ühestajad. Bigramm-ühestaja vaatab treeningetapil vaadeldavast sõnast ühe võrra ees olevat sõna, tõenäosused moodustuvad paarikaupa. Trigramm mudel aga tegeleb kolmikutega, vaadates sõnast kahe võrra ettepoole. Markovi mudeli baasil ühestajatest on eriti edukad just Markovi peitmudelil põhinevad ühestajad, sest need võimaldavad häid tulemusi saavutada ka väikestel treeningkorpustel (El Beze ja Merialdo, 1999).

Vähem on levinud reeglipõhised ühestajad. Selle meetodi puhul moodustatakse inimesele arusaadavad keelereeglid, mida järkjärgult rakendades välistatakse valed märgendid. Tuntuim reeglipõhine ühestamisformalism on kitsenduste grammatika (Voutilainenen, 1999).

Leidub ka juhtumi-põhist meetodit (ingl. k *case-based method*) kasutavaid ühestajaid. Juhtumi-põhine õppimisparadigma põhineb hüpoteesil, et tunnetuslike ülesannete sooritus (antud juhul loomuliku keele töötlus) baseerub uute olukordade analoogial varasemate kogemuste säilitatud esitusega, mitte varasemate juhtumite alusel loodud reeglite baasil (Daelemans, 1999:291).

Närvivõrkudel baseeruvad ühestajad kasutavad näiteid võrgu treenimiseks. Seda tehakse korduvalt üle kõigi näidete itereerides, võrreldes iga näite puhul võrgu poolt ennustatavat väljundit õige väljundiga ja muutes võrgu sõlmedevahelisi kaalusid vastavalt esitluse kasvule. Samaaegselt hoitakse ühenduste kaalude maatriksit ja unustatakse kasutatud näited (Daelemans, 1999:300).

Laia kasutust on leidnud ka transformatsioonipõhised märgendajad (Brill,1995). See meetod ühendab endas nii reeglipõhiste kui ka stohhastiliste märgendajate jooni. Reeglipõhiste märgendajatele sarnaselt põhineb transformatsioonipõhine meetod reeglitel, mis määravad, milline märgend millisele sõnale määrata. Kuid stohhastilistele märgendajatele sarnaselt on transformatsioonipõhine meetod masinõppe tehnika: reeglid tuletatakse automaatselt treeningandmetest. (Juravsky, 2008:185)

2.2.Ühestamise meetodid eesti keele jaoks

Eesti keele morfoloogilisel ühestamisel on seni kasutatud kahte põhilist meetodit: reeglipõhist ja statistilist.

Esimesel juhul koostatakse lingvistiliste reeglite komplekt sõnade järgnevuse vms alusel sõnavormi määramiseks, eesti keele morfoloogilisel ühestamisel kasutatakse kitsenduste grammatikat (Puolakainen, 2001).

Kitsenduste grammatika on oma loomult reduktsionistlik, s.o analüüsi alguses lisatakse igale sõnale kõik võimalikud analüüsivariandid ja seejärel hakatakse konteksti mittesobivaid eemaldama. Seetõttu nimetataksegi selles formalismis kasutatavaid reegleid kitsendusteks. Iga kitsendus esitab mõnda spetsiifilist keelereeglilaadset fakti, üldisem grammatikareegel kujuneb alles nende koosmõjust. Grammatikasse on võimalik lisada ka heuristilisi reegleid, mis kirjeldavad pigem keelesüsteemi tendentse kui üheselt tõeseid keelereegleid. (Müürisep, 2000) Kitsenduste grammatikat kasutav ühestaja ühestab küll väga õigesti, kuid jätab paljud sõnad mitmeseks. Ühestaja töö hindamiseks kasutatakse kahte mõistet: saagis ja täpsus. Saagis[1] (*recall*) näitab leitud õigete analüüside suhet võrreldes käsitsi leitud analüüsidega. Täpsus (*precision*) näitab õigete analüüside osakaalu kõigist leitud analüüsidest (Müürisep, 2000). Eesti keele kitsenduste grammatika rakendamisel morfoloogilisele ühestamisele oli saagis ligikaudu 97-98% ja täpsus 83-86% juures sõltuvalt tekstiliigist (Puolakainen 2000) .

Teisel juhul püütakse ühestamiseks abi saada teksti statistilisest analüüsist: leitakse sõnade esinemissagedused kontekstides ja peaaegu ei kasutatagi lingvistilisi reegleid. Statistilise ühestamise puhul on üheks enamlevinud meetodiks kujunenud Markovi peitmudel (ingl k *Hidden Markov Model* - HMM). HMM-i rakendamisel eesti keelele kasutati seda tema puhtal klassikalisel kujul. (Kaalep,Vaino, 1998)

Morfoloogilisel ühestamisel HMM-märgendajaga moodustati treeningtekstide alusel kõigepealt sõnaliikide esinemise tõenäosuste tabelid, mida siis seejärel inimese poolt parandati. Nii sai eemaldada lihtsamad eesti keele eripäratõttu tekkinud vead. Väga oluline oli ka märgenditesüsteemi valik. Töös kasutati 88 märgendit, mis olid valitud järgmiselt: eristatati omadussõnu, põhiarvsõnu, järgarvsõnu, nimisõnu, pärisnimesid, isikulisi asesõnu,

[1] Kasutatakse ka mõistet korrektsus.

muid asesõnu, lühendeid, verbe, alistavaid ja rinnastavaid sidesõnu, hüüdsõnu, ees- ja tagasõnu, määrsõnu, punktuatsioonisümboleid ja tundmatuid sõnu.

Käändsõnade puhul eristati 5 käänet: nimetavat, omastavat , osastavat , lühikest sisseütlevat e. aditiivi ja “kõiki muid”. Isikuliste asesõnade puhul eristati lisaks ka kolme isikut. Ei eristatud ainsust ja mitmust. Verbide puhul eristatati kokku 13 märgendit: “ei”, “ära”, esimene pööre, teine pööre, kolmas pööre, kaudne kõneviis, “pole” ja “polnud”, *da*-infinitiiv, 0-lõpuline vorm, tingiva kõneviisi vormid, käskiva kõneviisi vormid, *ma*-infinitiivi vormid, partitsiibid. Ei eristatud ainsust ja mitmust ega aega. Ühestaja sai tabelite koostamiseks sisendiks morfoloogiliselt analüüsitud, kuid ühestamata teksti, morfoloogiliselt analüüsitud ja ühestatud teksti, lisaks veel ühestamisel kasutatavate märgendite loendi ja teisendustabeli morfoloogilistelt märgenditelt ühestaja märgendidele (see tabel seetõttu et morfoloogilised märgendid ei olnud enamasti samad, mis ühestaja märgendid). Treenimiseks kasutati G. Orwelli romaani „1984“ eestikeelset versiooni, milles on 75 000 sõna, ja testimiseks 2000 sõnalist osa Vello Lattiku raamatust “Mihklipäeval.Mihklikuul” . Erinevusi käsitsi ühestatud tekstiga oli umbes 7,1%. (Kaalep,Vaino, 1998) Viimaste märgendaja täienduste järgsete katsete tulemusel (Veskis, Liba, 2008) on ESTYHMM õigete märgendite jaotus 96-94% vahel, olenevalt testkorpusest. Kasutati 118 erinevat märgendit.

Mõlemal eesti keele jaoks olemasoleval ühestajal on omad puudused, reeglipõhise ühestaja põhiprobleemiks on analüüside mitmesus, statistilise ühestaja puuduseks märgendite väike hulk ja nende vähene sobivus mõnede praktiliste rakenduste jaoks. Töös kasutatud statistilised märgendajad on andnud häid tulemusi erinevate keeltega, mistõttu tekkis soov katsetada TreeTaggeri ja TnTTaggeri tulemusi eesti keele puhul.

3. Markovi mudeli märgendamine

3.1. Tõenäosuslik mudel

Märgendamise algoritmi sisendiks on sõnad ja eelnevalt kindlaksmääratud märgendite hulk. Väljundiks on üksik parim märgend iga sõna jaoks. (Juravsky, 2009) Markovi mudeli-põhise märgendamise korral vaadeldakse tekstis asuvat märgendite jada kui Markovi ahelat. Käsitsi märgendatud tekstist koosnevat treeninghulka kasutatakse märgendijadade korrapärasuste leidmiseks. Märgendi t_k, millele järgneb märgend t_j, suurim tõepära hinnang (ingl k. maximum likelihood estimation) leitakse konkreetsele märgendile järgnevate märgendite suhtelistest sagedustest:

$$P(t_k|t_j)= f(t_j, t_k)/f(t_j) \qquad (4.1)$$

Tõenäosuste $P(t_{i+1}|t_i)$ abil saab arvutada konkreetse märgendite jada tõenäosuse. Eesmärgiks on leida kõige tõenäolisem märgendite jada sõnade jada jaoks, täpsemalt – kõige tõenäolisem seisundite jada sõnade jada kohta (on ju Markovi mudeli seisundid antud juhul märgenditeks). Markovi mudeli puhul saab otse jälgida märgendatud korpuse seisundeid (märgendeid). Iga märgend vastab erinevale seisule. Samuti saab suurima tõepära hinnanguga otse määrata konkreetses seisundis oleva sõna w_1 tõenäosust:

$$P(w_l|t_j)= f(w_l, t_j)/f(t_j) \qquad (4.2)$$

Vastavalt Bayesi reeglile saab leida lause $w_{1,n}$ jaoks parima märgenduse $t_{1,n}$:

$$\begin{aligned}\arg\max_{t1,n}P(t_{1,n}|w_{1,n}) &= \arg\max_{t1,n}P(w_{1,n}|\, t_{1,n})P(t_{1,n})/\, P(w_{1,n}) \\ &= \arg\max_{t1,n}P(w_{1,n}|\, t_{1,n})P(t_{1,n}) \qquad (4.3)\end{aligned}$$

Lisaks piiratud horisoni (ingl k. limited horizon) eeldusele, eeldame ka, et sõnad on üksteisest sõltumatud ja sõna identiteet sõltub ainult tema märgendist. Seega:

$$\begin{aligned}P(w_{1,n}|\, t_{1,n})P(t_{1,n}) &= \Pi\, P(w_i|\, t_{1,n})xP(t_n|\, t_{1,n-1})x\, P(t_{n-1}|\, t_{1,n-2})x...xP(t_2|t_1) \\ &= \Pi\, P(w_i|\, t_i)xP(t_n|\, t_{n-1})x\, P(t_{n-1}|\, t_{n-2})x...xP(t_2|t_1) \\ &= \Pi\, [P(w_i|\, t_i)xP(t_i|\, t_{i-1})] \qquad (4.4)\end{aligned}$$

Kokku saame:

$$t_{1,n}= \arg\max_{t1,n}P(t_{1,n}|w_{1,n})= \arg\max_{t1,n}\, \Pi\, [P(w_i|\, t_i)xP(t_i|\, t_{i-1})] \qquad (4.5)$$

Markovi mudel treenimise algoritm:

```
for all tags tj do
      for all tags tk do
            P(tk|tj):= f(tj, tk)/f(tj)
      end
end
for all tags tj do
      for all tags w1 do
            P(w1|tj)= f(w1, tj)/f(tj)
      end
end
```

(Manning,1999)

3.2. Viterbi algoritm

Märgendamiseks on kasutusel efektiivne Viterbi algoritm. Algoritm sisaldab kolme etappi: initsialiseerimine, induktsioon, lõpetamine ja tulemuse väljastamine. Arvutatakse kahe funktsiooni väärtused: $\delta_i(j)$, mis annab meile sõna i tõenäosuse seisundis (märgendiga) j ja $\psi_{i+1}(j)$, mis annab kõige sobivama seisundi (märgendi) sõna i jaoks, kui sõna $i+1$ korral ollakse seisundis j.

Initsialiseerimise etapi käigus väärtustatakse märgendile PERIOD tõenäosus 1.0, sellega eeldatakse, et laused on jagatud perioodideks ja mugavuseks lisatakse esimese lause ette periood.

Induktsiooni sammul:

$$\delta_{i+1}(t_j) = \max_{1\le k\le T}[\delta_i(t_k) \times P(t_j|\, t_k) \times P(w_{i+1}|\, t_j)],\ 1\le j\le T \qquad (4.6)$$

$$\psi_{i+1}(t_j) = \arg\max_{1\le k\le T}[\delta_i(t_k) \times P(t_j|\, t_k) \times P(w_{i+1}|\, t_j)],\ 1\le j\le T \qquad (4.7)$$

Ning lõpetamine, kus $X_1 \ldots X_n$ on sõnade $w_1 \ldots w_n$ märgendid:

$$X_n = \arg\max_{1\le j\le T}\delta_n(t_j) \qquad (4.8)$$

$$X_i = \psi_{i+1}(X_{i+1}),\ 1\le i\le n-1 \qquad (4.9)$$

$$P(X_1,\ldots,X_n) = \max_{1\le j\le T}\delta_{n+1}(t_j) \qquad (4.10)$$

Märgendamise algoritm:

kommentaar: antud: lause pikkusega n

kommentaar: initsialiseerimine

$\delta_1(PERIOD)=1.0$

$\delta_1(t)=0.0$ for $t \neq PERIOD$

kommentaar: induktsioon

for i:=1 to n step 1 do

  for all tags t_j do

    $\delta_{i+1}(t_j) := \max_{1 \leq k \leq T}[\delta_i(t_k) x\ P(t_j|\ t_k) x P(w_{i+1}|\ t_j)]$

    $\psi_{i+1}(t_j) := \arg \max_{1 \leq k \leq T}[\delta_i(t_k) x\ P(t_j|\ t_k) x P(w_{i+1}|\ t_j)]$

  end

end

kommentaar: lõpetamine

$X_{n+1} := \arg \max_{1 \leq j \leq T} \delta_{n+1}(j)$

for j:=n to 1 step -1 do

  $X_j := \psi_{j+1}(X_{j+1})$

end

$P(X_1,...,X_n) := \max_{1 \leq j \leq T} \delta_{n+1}(t_j)$

(Manning,1999)

3.3. TnT- Statistiline sõnaliigi märgendaja

3.3.1. Arhitektuuri aluseks olev mudel

Tnt kasutab sõnaliigi märgendamiseks teist järku Markovi mudeleid. Mudeli seisundid esitavad märgendeid, väljundit esindavad sõnad. Siirde tõenäosused sõltuvad seisunditest, seega märgendite paaridest. Väljundi tõenäosused sõltuvad ainult kõige viimasest kategooriast. Täpsemalt:

$$\underset{t_1 \ldots t_T}{\operatorname{argmax}} \left[\prod_{i=1}^{T} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_i) \right] P(t_{T+1}|t_T) \qquad (4.11)$$

Kui on antud sõnade järjend $w_1...w_t$, pikkusega T. $t_1..t_T$ on märgendite hulga elemendid, lisamärgendid t_{-1}, t_0 ja t_{T+1} on järjendi alguse- ja lõpumärgendid. Lisamärgendite kasutamine parandab natuke märgendamise tulemusi. Kui sisendis pole lause piirid ära märgitud, paneb TnT vastavad märgendid ise, kui kohtab . ! ? või ; märki.

Siirde- ja väljundtõenäosused hinnatakse märgendatud korpuse alusel. Esimese sammuna kasutatakse suurima tõepära tõenäosusi P, mis on tuletatud suhtelistest tõenäosustest:

Unigrammidel:	$\acute{P}(t_3)=f(t_3)/N$
Bigrammidel:	$\acute{P}(t_3\|t_2)=f(t_2, t_3)/f(t_2)$
Trigrammidel:	$\acute{P}(t_3\|t_1 ,t_2)=f(t_1, t_2, t_3)/f(t_1, t_2)$
Sõnavara:	$\acute{P}(w_3\|t_3)=f(w_3, t_3)/f(t_3)$

Kõikide t_1,t_2,t_3 kohta märgendite hulgas ja w_3–e kohta leksikonis. N on kogu sõnade ja märkide hulk treeningkorpuses. Suurima tõepära tõenäosus loetakse nulliks, kui vastavad tõenäosused jagatises on nullid. (Brants, 2000)

3.3.2.Silumine

Korpusest genereeritud trigrammide tõenäosusi ei saa tavaliselt andmete hajususe tõttu otse kasutada. See tähendab, et iga trigrammi kohta ei ole piisavalt näiteid, et tõenäosusi usaldusväärselt hinnata. Trigrammi esinemise tõenäosuse nulliks määramine, kuna see ei esinenud andmetes kordagi, võib omakorda kaasa tuua ebasoovitava tulemuse. TnT puhul on silumiseks kasutusel uni-,bi- ja trigrammide lineaarne interpolatsioon (ingl k. linear interpolation). Trigrammi tõenäosust hinnatakse järgmiselt:

$$P(t_3|t_1,t_2) = \lambda_1\hat{P}(t_3) + \lambda_2\hat{P}(t_3|t_2) + \lambda_3\hat{P}(t_3|t_1,t_2) \qquad (4.12)$$

P-d on tõenäosuste suurimad tõepära hinnangud ja $\lambda_1+ \lambda_2 +\lambda_3=1$.
Kasutatakse kontekstist sõltumatut varianti lineaarsest intepolatsioonist, st λ-de väärtused ei sõltu konkreetsest trigrammist. Vastupidiselt intuitsioonile annab see paremaid tulemusi, kui kontekstist sõltuv variant. Andmete hõresuse probleemi tõttu ei saa hinnata iga

trigrammi kohta erinevat λ-de hulka. Seetõttu on tavapraktika grupeerida trigramme sageduse järgi ja hinnata seotud λ-de hulka.

λ_1, λ_2 ja λ_3 väärtuseid hinnatakse kustutatud interpolatsioonihinnangu alusel. See tehnika eemaldab järjestikku iga trigrammi treeningkorpusest ja määrab λ-dele parimad väärtused kõikidest teistest n-grammidest korpuses. Uni-, bi- ja trigrammide sageduste loenditest saab efektiivselt määrata kaalud. (Brants,2000) Seda iseloomustab algoritm:

```
set λ1 = λ2 = λ3 = 0
foreach trigram t1, t2, t3 with f(t1, t2, t3) > 0
    depending on the maximum of the following three values:
        case (f(t1,t2,t3)−1)/(f(t1,t2)−1): increment λ3 by f(t1, t2, t3)
        case (f(t2,t3)−1)/(f(t2)−1): increment λ2 by f(t1, t2, t3)
        case (f(t3)−1)/(N−1): increment λ1 by f(t1, t2, t3)
    end
end
normalize λ1, λ2, λ3
```

3.3.3.Tundmatud sõnad

Tnt kasutab tundmatute sõnade puhul suffiksi analüüsi. Märgendite tõenäosused määratakse vastavalt sõnalõppudele. Konkreetse suffiksi tõenäosuste jaotus genereeritakse kõikide sõnade alusel treeningkorpuses, millel on sama, mingi eelnevalt määratud pikkusega, suffiks. Tõenäosused silutakse järjestikuse üldistamisega (ingl k. successive abstraction). Leitakse märgendi t tõenäosus, kui on antud n-tähelise sõna viimased m tähte l_i: $P(t|l_{n-m+1},...l_n)$. Mida üldisem kontekst, seda rohkem suffiksi tähti. Rekursiooni valem on:

$$P(t|l_{n-i+1},\ldots l_n) = \frac{\hat{P}(t|l_{n-i+1},\ldots l_n) + \theta_i P(t|l_{n-i},\ldots,l_n)}{1+\theta_i} \quad (4.13)$$

Kus i=m..0, kasutades suurima tõepära hinnangut Ṕ leksikoni sagedustest,kaaludest θ_i ja initsialiseerimist

$$P(t) = \hat{P}(t). \tag{4.14}$$

Suurim tõepära hinnang i-pikkusega suffiksi jaoks saadakse korpuse sagedustest:

$$\hat{P}(t|l_{n-i+1}, \ldots l_n) = \frac{f(t, l_{n-i+1}, \ldots l_n)}{f(l_{n-i+1}, \ldots l_n)} \tag{4.15}$$

Markovi mudeli jaoks vajatakse ümberpööratud tingimuslikke tõenäosusi $P(l_{n-i+1}|t)$, mis saadakse Bayesi pöördjärjestuse abil.

Suffiksi pikkus sõltub konkreetsest sõnast: kasutatakse pikimat sufiksit, mis treeningkorpuses leidub, kuid mitte üle 10 märgi. Θ_i leidmiseks kasutatakse kontekstist sõltumatut lähenemist, selleks antakse θ_i väärtuseks treeningkorpuse märgendite piiramatu suurima tõepära tõenäosuste standardhälbe, st:

$$\theta_i = \frac{1}{s-1} \sum_{j=1}^{s} (\hat{P}(t_j) - \bar{P})^2 \tag{4.16}$$

Kus i=0..m-1, kasutades s märgendit ja keskmist:

$$\bar{P} = \frac{1}{s} \sum_{j=1}^{s} \hat{P}(t_j) \tag{4.17}$$

(Brants,2000)

Suurtähtedega ja väiketähtedega sõnade jaoks kasutatakse erinevat lähenemist, sõltuvalt esitähest hoitakse suffikseid erinevates trie-des. Kuna tundmatud sõnad on ebakorrapärased, siis kasutatakse leksikonis nende sõnade suffikseid, mille sagedus leksikonis on alla kümne.

TnTTaggerit (http://www.coli.uni-saarland.de/~thorsten/tnt/) on kasutatud näiteks inglise keele, saksa keele ja rootsi keele märgendamiseks, tulemused jäid 95-97% vahele. TnTTagger on vabalt kasutatav mitteärilistel eesmärkidel uurimustööde läbiviimiseks, programmi saamiseks tuleb täita litsentsileping, faksida see Ameerika Ühendriikidesse, mille peale saadetakse allalaadimiseks vajalik link ning kasutajanimi. TnTTaggeri põhieelisteks on head tulemused tundmatute sõnadega, programmi töökiirus ja mugav kasutatavus.

4. Tõenäosuslik sõnaliikide märgendamine otsustuspuu abil

4.1.Sissejuhatus

Tuntuima statistilise morfoloogilise ühestamise meetodi Markovi peitmudeli (Manning ja Schütze, 1999) alusel loodud ühestajate põhiprobleemiks on, et neil on raskusi väikese ja hajusa treeninghulga alusel õigete hinnangute tegemisel. Selle probleemi lahenduseks on välja pakutud otsutuspuude meetod. Otsustuspuu määrab automaatselt otsustamiseks vajaliku konteksti suuruse. Kontekstiks ei ole enam mitte ainult trigrammid ja bigrammid, vaid ka piirangud kontekstile nt tag_{-1} = ADJ ja $tag_{-2} \neq$ ADJ ja $tag_{-2} \neq$ DET (Schmid, 1994) s.t. sõnale eelnev sõna on adjektiiv ja üleeelmine sõna ei ole adjektiiv ega ka mitte artikkel.

Otsustuspuu õppimise meetod baseerub eeldusel, et näidetevahelisi sarnasusi saab kasutada otsustuspuude automaatseks eraldamiseks (Daelemans, 1999: 297).

TreeTaggeril on palju ühist n-gramm ühestajatega: nad mõlemad kasutavad hindamisel märgendatud sõnade järjendit.

$$p(w_1w_2...w_n, t_1t_2...t_n) := p(t_n|t_{n-2}t_{n-1})\ p(w_n|t_n)\ p(w_1w_2...w_{n-1}, t_1t_2...t_{n-1}) \quad (3.1)$$

Siin ja edaspidi $p(t_n)$ tähistab märgendi t_n esinemise tõenäosust, $p(w_n)$ sõna w_n esinemise tõenäosust ja F on esinemise sagedus.

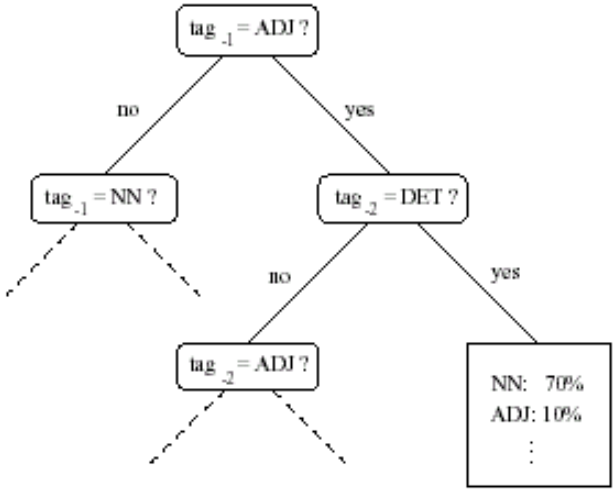
Meetodid erinevad vaid siirdetõenäosuste (märgendite vaheliste tõenäosuste) hindamise poolest. N-gramm mudelid kasutavad sageli hindamisel MLE (Maximum likehood estimation) printsiipi - maksimaalse tõepära hinnangut.(Schmid, 1994)

Selline meetod aga toob kaasa probleeme, sest mõnede harva esinevate trigrammide puhul võib olla raske mõista, kas tegemist on sõnade väga haruldase koosesinemisega või on esinemine lihtsalt grammatiliselt vale. Esimesel juhul oleks tõenäosus võrdne ühega, viimasel juhul nulliga.

4.2. TreeTagger

Vastupidisel kirjeldatud n-gramm-märgendajale kasutab TreeTagger hindamiseks binaarset otsustuspuud. Trigrammi esinemise tõenäosus leitakse mööda otsustuspuud allapoole lehtedeni liikudes (Schmid, 1994). Otsustuspuu näide on toodud joonisel 4.1.

Vaatame näitena olukorda, kus tahame teada, mis sõnaliigi esindaja on sõna, mille ees on artikkel ja omadussõna. Vaatame, kas sellele sõnale eelnev sõna on omadussõna (tag.$_{1}$=ADJ?), liigume mööda jah-vastuse kaart, kas omadussõnale eelnev sõna on artikkel, jah - jõuame leheni, kus selgub et 70% tõenäosusega on tegemist nimisõnaga.



Joonis 4.1 Otsustuspuu näide

4.2.1. Otsustuspuu moodustamine

Otsustuspuu on andmestruktuur, milles sõlmed esindavad teste ja kaared nende vahel võimalikke vastuseid. Lahendus leitakse mööda otsustuspuud allapoole liikudes ja leheni jõudes. Tee, mida mööda läbi otsustuspuu liikuda, sõltub vastustest, mida sõlmedes testile antakse (Daelemans, 1999:297). Otsustuspuu moodustatakse rekursiivselt treeninghulga trigrammide alusel kasutades täiendatud versiooni ID3-algoritmist. Igal rekursiooni sammul luuakse test, mille alusel jagatakse trigrammide hulk kahte alamhulka sõltuvalt kolmanda märgendi maksimaalsest tõenäosusest. Testi käigus vaadeldakse ühte kahest eelnevast märgendist ja kontrollitakse, kas see on identne märgendiga t. Test on järgmises vormis:

$tag_{-i} = t;\ i \in \{1, 2\};\ t \in T,$ kus T on märgendite hulk.
(Schmid, 1994)

Igal rekursiooni sammul võrreldakse kõiki teste ja nendest kõige rohkem informatsiooni sisaldav liidetakse vastava sõlme külge. Siis laiendatakse sõlme rekursiivselt igal testi q poolt defineeritud alamhulgal saades tulemuseks jah- ja ei-alampuud. Võrdlemise aluseks on kolmanda märgendi kohta kogutud informatsiooni hulk. Informatsiooni juurdekasvu kasumi maksimiseerimine on samaväärne keskmise informatsioonihulga I_q minimiseerimisega. (Schmid, 1994)

4.2.2.Lihtsustatud algoritm

Otsustuspuude genereerimist selgitab järgmine lihtsustatud algoritm (Daelemans, 1999: 298)
Olgu antud näidete hulk T.
Kui T sisaldab üht või enamat näidet, mis kuuluvad kõik samasse klassi C_j ,
siis on T otsustuspuu leht kategooriaga C_j.

Kui T sisaldab erinevaid klasse, siis

- Vali tunnus ja jaota T alamhulkadesse, mis sisaldavad valitud tunnuse väärtusi. Otsustuspuu sisaldab juhtumi nime sisaldavat sõlme ja alamhulgale viitavat haru.
- Rakenda protseduuri selliselt moodustatud alamhulkadele.

Algoritmi põhiraskus on õige tunnuse valimises. Kogu andmehulga jaoks puude koostamine on NP-täielik ülesanne, seepärast on vajalik heuristiline tunnuse valik.

4.2.3.Otsustuspuu kärpimine

Pärast esialgse otsustuspuu moodustamist puu kärbitakse. Kui mõlemad alamsõlmed on lehed ja kaalutud informatsiooni kasum (weighted information gain) sõlmel on alla mingit teatud lävendit, siis eemaldatakse alamsõlmed ja sõlm saab ise leheks. Kaalutud informatsiooni kasum G on defineeritud kui :

$$G=f(C)(I_0-I_q) \quad (3.2)$$

$$I_0= \sum p(t|C)\log_2 p(t|C) \quad (3.3)$$

Kus I_0 on informatsiooni hulk, mis on vajalik vastaval sõlmel ühestamiseks ja I_q on informatsiooni hulk, mis on ikkagi vajalik peale testi q tulemuste teada saamist. On oluline, et informatsiooni kasumi kriteeriumit ei kasutataks puu loomise faasis, vaid alles peale seda. Nagu teisedki tõenäosuslikud ühestajad nii ka TreeTagger selgitab parima märgendite järjendi välja Viterbi algoritmi abil. (Schmid, 1994)

4.2.4.Leksikon

Leksikon sisaldab iga sõna erinevaid märgendivõimalusi. Sellel on kolm osa: täisleksikon, sufiksi leksikon ja vaikeväärtus.

Kõigepealt otsitakse sõna TreeTaggeri täisleksikonist, kui sõna leitakse, siis tagastatakse vastav tõenäosuse vektor. Vastasel juhul muudetakse suurtähed väiketähtedeks ja otsitakse uuesti täisleksikonist. Kui ükski eelnevatest tegevustest ei ole olnud edukas, tagastatakse vaikeväärtus.

Kui märgendaja töötleb tundmatut teksti, on vägagi tõenäoline, et tegemist tuleb suure hulga tundmatute sõnadega, isegi kui leksikon on suur. Seega vajab märgendaja strateegiat tundmatute sõnade töötlemiseks. Kõige lihtsam võimalus on siduda iga tundmatu sõna iga sõnaliigi märgendiga võrdse tõenäosusega. Kuid teatud sõnaliigi märgendid (näiteks artiklid, kaassõnad ja asesõnad) võib sellest nimekirjast välja jätta, sest kõik need sõnad on suure tõenäosusega leksikonis esindatud. (Schmid, 1995)

Sufiksileksikon on puu kujul, iga puu sõlm (v.a juur) on varustatud tähega. Leht-tippudes on märgendi täenäosuste vektorid. Sufiksipuu läbitakse alustades juurest, igal sammul liigutakse mööda haru, millel on sõna järgmine täht.

Sufiksileksikon moodustatakse automaatselt treeningkorpuse baasil. Sufiksipuu moodustatakse kõigi nimisõna, verbi ja omadussõnana märgendatud vähemalt viietähelise pikkusega sõnade baasil. Lisaks loeti kõigi sufiksite märgendite tõenäosused ja säilitatakse need vastava puu sõlmedes. Siis arvutatakse iga puu sõlmele informatsiooni mõõde (ingl k. information measure) I(S):

$$I(S) = - \sum P(pos|\ S)\log_2 P(pos|\ S) \quad (3.4)$$

S on sufiks, mis vastab vastavale sõlmele ja P(pos|S) on sõna sõnaliigi tõenäosus, mis vastab sufiksile S. Informatsiooni mõõdet kasutatakse sufiksipuu kärpimiseks. Iga lehe jaoks arvutatakse kaalutud informatsiooni kasum (ingl k. weighted information gain) G(aS):

$$G(aS)= F(aS) (I(S)- I(aS)) \quad (3.5)$$

Kus S on vanema sõlme sufiks, aS on vastava sõlme sufiks ja F(aS) on sufiksi aS sagedus.

Kui sufiksipuu mõne lehe informatsiooni kasum on alla antud väärtuse, leht eemaldatakse. Kõigi kustutatud lehtede vanemate märgendite sagedused kogutakse vanema vaikesõlme, kui vaikesõlm osutub ainsaks järelolevaks sõlmeks, kustutatakse ka see. Sellisel juhul saab vanem sõlm leheks ja asutakse kontrollima, kas ka seda saaks kustutada. Kui vaikesõlme ei ole, loetakse otsing leksikonis ebaõnnestunuks ja tagastatakse vaikeväärtus. Vaikeväärtus koostatakse eraldades kõigi kärbitud sufiksipuu lehtede märgendite tõenäosusi juursõlme märgendite sagedustest ja siis normaliseerides tulemuste sagedusi. (Schmid, 1994)

Programm on vabalt allalaaditav aadressilt http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/. Vaatamata sellele, et programm on välja töötatud 1994. aastal, on see jätkuvalt aktuaalne, parameeterfailid on olemas inglise, saksa, itaalia, taani, hispaania, bulgaaria, vene, prantsuse ja vana prantsuse keele jaoks, mitmete keelte kohta on tehtud uurimusi just viimaste aastate jooksul.

5. Kitsenduste grammatika

5.1.Ülevaade

Kitsenduste grammatika on keelest sõltumatu formalism pindmiseks morfoloogial põhinevaks süntaktiliseks analüüsiks. Süntaktiline analüüs koosneb kolmest etapist: morfoloogilisest ühestamisest, osalausepiiride määramisest ja sõnade süntaktiliste funktsioonide määramisest.

Kitsenduste grammatika on oma loomult reduktsionistlik, s.o analüüsi alguses lisatakse igale sõnale kõik võimalikud analüüsivariandid ja seejärel hakatakse konteksti mittesobivaid eemaldama. Seetõttu nimetataksegi selles formalismis kasutatavaid reegleid kitsendusteks. Iga kitsendus esitab mõnda spetsiifilist keelereeglilaadset fakti, üldisem grammatikareegel kujuneb alles nende koosmõjust. Grammatikasse on võimalik lisada ka heuristilisi reegleid, mis kirjeldavad pigem keelesüsteemi tendentse kui üheselt tõeseid keelereegleid. (Müürisep, 2000)

Kitsenduste grammatika formalism jagab teksti töötlemise järgmisteks etappideks:

1. Eeltöötlus, mille käigus tuntakse ära lausete lõpud, tehakse kirjavahemärkide analüüs ja teisendatakse tekst morfoloogiaanalüsaatori jaoks sobivale kujule.
2. Morfoloogiline analüüs - leitakse sõnavormi tüvi ning lõpud ja neile vastav sõnaliik, kääne või pööre. Kui sõnavorm on mitmeti tõlgendatav, antakse selle kõik tõlgendused.
3. Morfoloogiline ühestamine - konteksti info põhjal leitakse sõnavormi paljude tõlgenduste seast korrektne tõlgendus.
4. Osalause piiride määramine - konteksti info, kirjavahemärkide ja morfoloogilise info põhjal leitakse liitlausetes osalausete piirid. See etapp toimub paralleelselt morfoloogilise ühestamisega, sest mõlemad on teineteisest väga sõltuvad.
5. Süntaktiliste märgendite lisamine - morfoloogilise info ja konteksti põhjal lisatakse sõnavormile kõik võimalikud süntaktilised märgendid. Osaliselt võidakse märgendeid lisada ka juba morfoloogilise analüüsi käigus leksikonist või enne morfoloogilist ühestamist.
6. Süntaktiline ühestamine - konteksti põhjal eemaldatakse sõnavormilt kõik lubamatud süntaktilised märgendid.

Kitsenduste grammatika tegeleb nelja viimase etapiga. (Müürisep, 2000) Antud töös vaatleme täpsemalt morfoloogilist ühestamist.

Morfoloogilise analüüsi käigus ei arvestata sõna konteksti, paljud analüsaatori poolt väljastatavad tõlgendused ei ole aga antud kontekstis grammatiliselt lubatavad. Morfoloogilise ühestamise etapil püütakse eemaldada kõik sõna sellised tõlgendused, mis ei sobi antud konteksti. Selleks kasutatakse morfoloogilisi kitsendusi, mis konteksti info põhjal leiavad sõnale korrektse morfoloogilise tõlgenduse. Vaatame näitena järgmist morfoloogiliselt analüüsitud lauset:

```
Aknas
    aken+s //_S_ com sg in //
kustus
    kustu+s //_V_ main indic impf ps3 sg ps af #Intr //
tuli
    tule+i //_V_ main indic impf ps3 sg ps af #Intr //
    tuli+0 //_S_ com sg nom //
$.
    . //_Z_ Fst //
```

Joonis 5.1 Näide morfoloogiliselt analüüsitud lausest

Sellises kontekstis leitakse kohordist *tuli* õige tõlgendus järgmise kitsenduse rakendamisel: eemaldada kohordist verbi pöördeline vorm, kui antud sõnale eelneb vahetult verbi pöördeline vorm, mis on sõnavormi ainus tõlgendus. (Müürisep, 2000)

5.2.Reeglite kuju ja tulemused

Kitsenduste grammatika koosneb reeglitest ehk kitsendustest: iga kitsendus koosneb domeenist, operaatorist, eesmärgist ja kontekstitingimustest. Domeen näitab, kas reeglit rakendatakse ainult kindlale sõnavormile või kõikidele sõnadele. Operaator näitab, kas antud reeglis käsitletav morfoloogiline tõlgendus tuleb ainsana alles jätta või eemaldada. Kontekstitingimused määravad, millise konteksti korral reeglit rakendatakse. Kontekstina vaadeldakse kogu lauset. Kontekstis asuvaid sõnu saab adresseerida vaadeldava sõna suhtes jäigalt (nt. kaks sõna paremale) või määramata (kusagil paremal kontekstis). Kontekstitingimused võivad olla kas jaatavad (kontekstis leidub märgend) või eitavad (kontekstis ei lei-

du märgendit). Konteksti saab piirata osalauseni. Kontekstis võib kasutada ka suhtelisi positsioone, mille korral adresseerimine ei toimu vaadeldava tõlgenduse suhtes, vaid mõne kontekstis asuva sõna suhtes (nt. antud sõnast vasakul leidub verb ja sellest verbist paremal kuni antud sõnani ei leidu kirjavahemärke) (Roosmaa, 2001). Eesti keele kitsenduste grammatika morfoloogilise ühestamise osa koosneb 38-st osalausete määramise reeglist ja 1240-st morfoloogilisest kitsendusest. Ühestaja rakendamine vähendas mitme tõlgendusega sõnade protsenti 4 korda, vigu oli enamikes tekstides 1-2%. (Pualakainen, 2001)

6. Ühestamine ja eesti keel

6.1.Hetkeolukord

Hetkel on eestikeelsete morfoloogiliselt ühestatud tekstide olukord on järgmine: on olemas kahe inimese poolt teineteisest sõltumatult ühestatud korpus ca 500 000 sõnaga. (http://www.cl.ut.ee/korpused/morfkorpus/)
Tekstid kuuluvad järgmistesse klassidesse (sõnade hulka ei ole arvestatud kirjavahemärke):

Liik	**sõnade arv**
Ilukirjandus (eesti autorid)	104 000
G. Orwelli "1984"	75 500
Ajakirjandus	111 000
Seadused	121 000
Horisont	98 000
Info-tekstid	4 000
Kokku	513 000

Joonis 6.1 Morfoloogiliselt ühestatud korpuse tekstide jaotus

Ilukirjanduse tekstid on pärit eesti autorite töödest. Ajakirjanduse tekstid on ajalehtedest „Postimees“, „Sõnumileht“, „Eesti Päevaleht“, „Äripäev“ ja „Maaleht“ ning kuuluvad ajavahemikku 1995-1999. Tõlkekirjandusest on esindatud G.Orwelli ulmeromaan „1984“.

```
<s>
Oli    ole+i //_V_ main indic impf ps3 sg ps af //
k&uuml;lm    k&uuml;lm+0 //_A_ pos sg nom //
selge    selge+0 //_A_ pos sg nom //
aprillip&auml;ev    aprilli_p&auml;ev+0 //_S_ com sg nom //
,    , //_Z_ Com //
kellad    kell+d //_S_ com pl nom //
l&otilde;id    l&ouml;&ouml;+id //_V_ main indic impf ps3 pl ps af //
parajasti    parajasti+0 //_D_ //
kolmteist    kolm_teist+0 //_N_ card sg nom l //
.    . //_Z_ Fst //
</s>
```

Joonis 6.2 Näide eesti keele morfoloogiliselt ühestatud korpusest

`Märgendid <s> ja </s> tähistavad vastavalt kas lause algust või lõppu.`
Read failis on kujul:

```
sõna     tüvi+lõpp // analüüs //
```

- <sõna> on sõna sellisena, nagu ta algselt esines
- <tüvi> on lemma e. algvormi tüvi: käändsõnadel ainsuse nimetav (kui seda ei ole olemas, siis mitmuse nimetav), pöördsõnadel ma-infinitiivi tüvi ilma (ma-lõputa)
- <lõpp> on sõna lõpp, kusjuures mitmuse tunnus on temaga liitunud (nagu seda on käsitletud ka Ülle Viksi "Väikeses vormisõnastikus"); partikkel GI/KI, kui ta esineb, on lihtsalt lõppu "kleepunud"; ka juhul, kui sõnal ei saagi lõppu olla (nt. hüüdsõnal), pannakse sõnale lõpp - nn. null-lõpp
- <analüüs> on üks variantidest, mis on kõik esitatud morfoloogiliste kategooriate tabelis.

Kui on tegemist liitsõna või tuletisega, siis:

- Tüvi on eristatud eelnevast komponendist '_' märgiga;
- Lõpp on eristatud eelnevast komponendist '+' märgiga; nn. null-lõpp ongi '+0'
- Sufiks on eristatud eelnevast komponendist '=' märgiga. Sufiksite märkimine ei ole järjekindel: märgitakse ainult teatud hulka produktiivseid sufikseid.
- Lemmatüvi leitakse ainult viimase parempoolse komponendi alusel

Mitmesõnalised nimed on kujul:

New Yorgis New York+s //_S_ prop sg in // (EKK)

```
Oli
    ole+i //_V_ s, //
k&uuml;lm
    k&uuml;lm+0 //_S_ sg n, //
selge
    selge+0 //_A_ sg g, sg n, //
aprillip&auml;ev
    aprillip&auml;e=v+0 //_A_ sg n, //
    aprillip&auml;e=v+0 //_S_ sg n, //
,
    , //_Z_ //
```

```
kellad
    kell+d //_S_ pl n, //
    kella+d //_V_ d, //
l&otilde;id
    l&otilde;i+d //_S_ pl n, //
    l&otilde;+d //_S_ pl n, //
parajasti
    parajasti+0 //_D_ //
kolmteist
    kolm_teist+0 //_N_ sg n, //
.
    . //_Z_ //
```

Joonis 6.3 Näide samast lausest ühestamata kujul, morfoloogia-analüsaatori väljund

Morfoloogia-analüsaator eristab 17 erinevat sõnaliigi märgendit.

Kasutatakse järgmisi sõnaliigi märgendeid:

- _A_ omadussõna - algvõrre (adjektiiv - positiiv), nii käänduvad kui käändumatud, nt *kallis* või *eht*
- _C_ omadussõna - keskvõrre (adjektiiv - komparatiiv), nt *laiem*
- _D_ määrsõna (adverb), nt *kõrvuti*
- _G_ genitiivatribuut (käändumatu omadussõna), nt *balti*
- _H_ pärisnimi, nt *Edgar*
- _I_ hüüdsõna (interjektsioon), nt *tere*
- _J_ sidesõna (konjunktsioon), nt *ja*
- _K_ kaassõna (pre/postpositsioon), nt *kaudu*
- _N_ põhiarvsõna (kardinaalnumeraal), nt *kaks*
- _O_ järgarvsõna (ordinaalnumeraal), nt *teine*
- _P_ asesõna (pronoomen), nt *see*
- _S_ nimisõna (substantiiv), nt *asi*
- _U_ omadussõna - ülivõrre (adjektiiv - superlatiiv), nt *pikim*
- _V_ tegusõna (verb), nt *lugema*
- _X_ verbi juurde kuuluv sõna, millel eraldi sõnaliigi tähistus puudub, nt *plehku*
- _Y_ lühend, nt *USA*
- _Z_ lausemärk, nt -, /, ...
- _T_tundmatu sõna

(EKK) Teised morfoloogilised märgendid on kirjeldatud põhjalikult Tiina Puolakaineni töös (2001).

6.2. TreeTagger

Programm TreeTagger[2] on välja töötatud ja programmeeritud Stuttgardi ülikoolis ning tema kasutamine on vaba akadeemilistel eesmärkidel. Seda programmi on edukalt kasutatud inglise, saksa, prantsuse, itaalia, bulgaaria, hispaania, kreeka, portugali ja vana prantsuse keele morfoloogiliseks ühestamiseks.

Programm TreeTagger kasutab töötamiseks otsustuspuude meetodit. Programm ise koosneb kahest osast: treeningosast ja test-osast. Treeningosa moodustab ette antud ühestatud tekstifaili alusel keelele vastava parameeterfaili, mille alusel test-osas omakorda ühestamata teksti on võimalik märgendada.

6.2.1. Treenimine

Train-tree-tagger on treenimisprogramm, mis vajab käsurea argumenti järgmisel kujul:
train-tree-tagger <lexicon> <open class file> <infile> <outfile>
{-cl <context length>} {-dtg <min. decision tree gain>}
{-ecw <eq. class weight>} {-atg <affix tree gain>} {-st <sent. tag>}
<lexicon>- täielikku leksikoni sisaldava faili nimi, faili igal real on sõnavorm ja märgend-lemma paaride järjend, tegelikult pole lemma informatsiooni vaja ja selle võib asendada nt. „-“ (TreeTagger readme)

```
aasiku      //_S_sg_gen// -   //_S_sg_nom// -
aasimisest  //_S_sg_el// -
aasisin     //_V_main_indic_impf_ps1_sg_ps_af// -
aasmäe      //_S_sg_gen// -
aassalu     //_S_sg_gen// -   //_S_sg_nom// -   //_S_sg_part// -
aasta //_S_sg_gen// -   //_S_sg_nom// -
aasta-aastalt     //_JD_// -
aastaaeg    //_S_sg_nom// -
aastaaega   //_S_sg_adit// -  //_S_sg_part// -
```

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

Joonis 6.4 Näide leksikonist

<open class file>- fail mis sisaldab võimalike tundmatute sõnade märgendite nimekirja. See fail tavaliselt sisaldab adverbide, adjektiividee, nimisõnade, pärisnimede ja ka verbide märgendeid, aga mitte prepositsioone, artikleid, asesõnu või numbreid.
<infile> on sisendfail, igal real on üks sõna, millele järgneb õige sõnaliik (TreeTagger readme)

```
Kuid  //_JD_//
kas   //_JD_//
kunagi      //_JD_//
on    //_V_aux_indic_pres_ps3_sg_ps_af//
olnud //_V_main_partic_past_ps//
sellist     //_P_sg_part//
aega  //_S_sg_part//
,     SENT
mil   //_P_sg_ad//
oldi  //_V_main_indic_impf_imps_af//
ka    //_JD_//
ühel  //_P_sg_ad//
tasemel     //_S_sg_ad//
?     SENT
```

Joonis 6.5 Näide train-tree-taggeri sisendfailist

Sõnaliikide märgendid on samad, mis varemgi, lauselõpumärki tähistab märgend SENT.
<outfile> väljundfailiks ehk train-tree-taggeri töö tulemuseks parameeterfail, mida kasutatatakse ühestamisprogrammi-TreeTaggeri töös
Erinevate lippudega on võimalik määrata konteksti pikkust (-cl <context length>), minimaalset otsustuspuu kasumit (-dtg <min. decision tree gain>), ekvivalentsusklassi kaalu (-ecw <eq. class weight>), afiksipuu kasutegurit (-atg <affix tree gain>) ja milline on lauselõpumärgend (-st <sent. tag>).(TreeTagger readme)

6.2.2.Märgendamine

Märgendamine toimub tree-tagger programmi abil, mille esimeseks argumendiks on train-tree-tagger-i väljundfail ehk parameeterfail ja teiseks argumendiks sisendfail ning kolmandaks argumendiks on väljundfaili nimi. Sisendfailiks on tekst, mille iga sõna on eraldi real.

```
Aga
patrullil
polnud
suurt
tähtsust
.
```

Joonis 6.6 Näide tree-taggeri sisendfailist

Tree-taggeri standardväljundiks, ehk lõpptulemuseks on sõnaliik iga sõna jaoks eraldi real.

```
//_JD_//
//_S_sg_ad//
//_JD_//
//_A_sg_part//
//_S_sg_part//
SENT
```

Joonis 6.7 Näide treetaggeri standardväljundist

Lisaks on mitmeid erinevaid lippe väljundfaili täiustamiseks:

-token: prindib ka analüüsitava sõna

-lemma: prindib ka sõna algvormi, kui analüüsitavat sõna leksikonis pole asendatakse lemma <unknown> märgendiga.

-sgml: ei märgendata SGML kommentaare, st. ridu, mis algavad '<' ja lõpevad '>'-ga

-threshold <p>: printida kõik märgendid, mille tõenäosus on kõrgem kui <p> korda kõige tõenäolisema märgendi väärtus

-prob: prindib märgendite tõenäosused (nõuab lippu -threshold)

-no-unknown: printida <unknown> asemel tundmatute sõnade puhul analüüsitav sõna

-no-heuristics: mitte kasutada leksikoni heuristikaid

-quiet: mitte printida seisundi teateid

-pt-with-lemma: iga märgendi järel peaks olema tühik ja lemma

-pt-with-prob: iga märgendi järel peaks olema tühik ja märgendi tõenäosuse väärtus

-eos-tag <tag>: SGML märgend <tag> tähistab lause lõppu, eeldab lippu -sgml

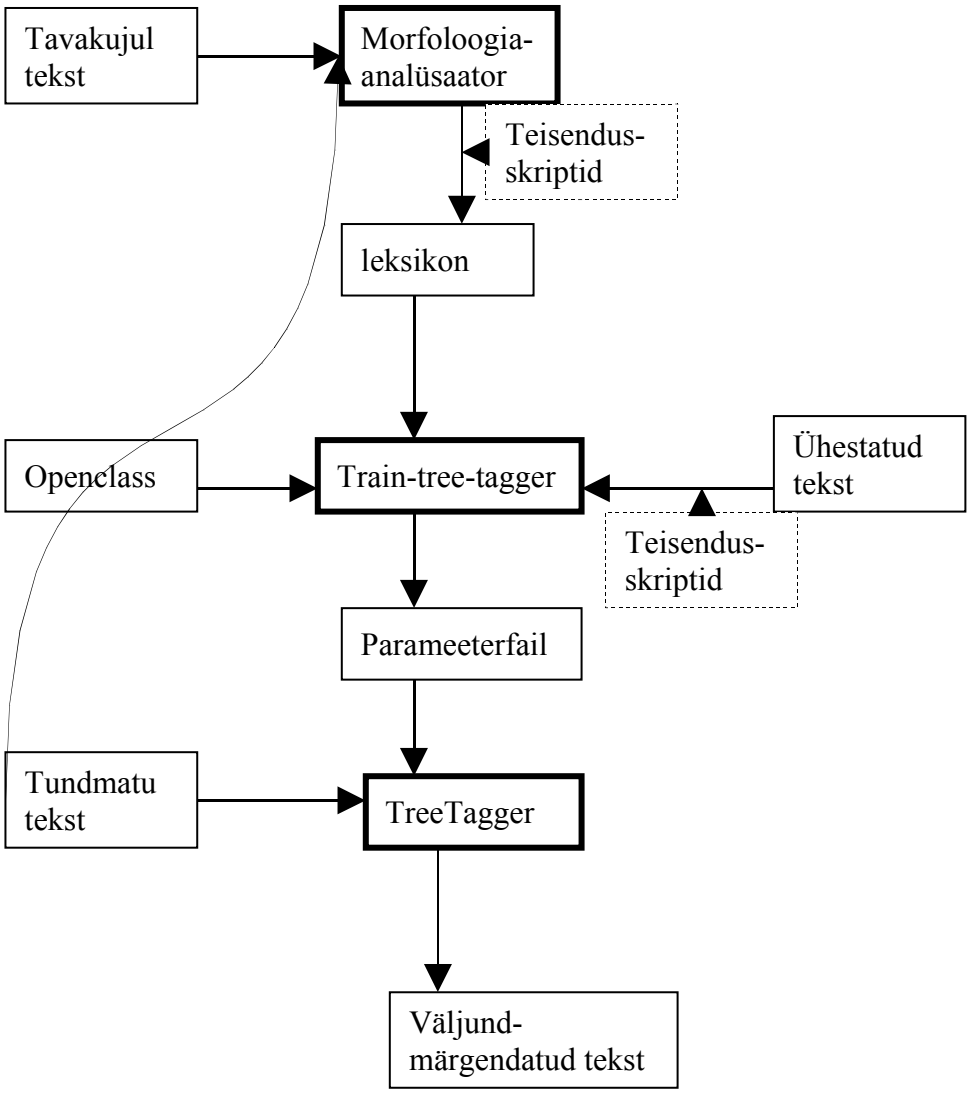
Lisaks on veel mitmeid nn. eksootilisi lippe, näiteks

-proto: prindib iga sõna kohta leksilise informatsiooni

f: kui sõna leiti leksikonist

c: kui sõna väiketähtedega leiti leksikonist

h: sõna sisaldab sidekriipsu ja sidekriipsule järgnev sõna leiti leksikonist, nt öö-elu ei leitud, aga leiti elu (TreeTagger readme)



Joonis 6.8 Tree-Taggeri sisendid ja väljundid

6.3. TnTTagger

TnT Tagger ehk pikemalt Trigrams ‚n' Tags programm on loodud 2000. aastal Thorsten Brantsi poolt Saarlandi ülikoolis. Programm on teist järku Markovi mudelite Viterbi algoritmi implementatsioon. Programmi kasutamine koosneb kahest osast: programmi treenimisest ehk parameeterfaili koostamisest juba ühestatud teksti alusel ja märgendamisest, mille korral programmi sisendiks on ilma märgenduseta fail ja väljundiks märgendatud fail.

6.3.1. Failide formaadid

%-märgiga algavad read loetakse programmi poolt kommentaarideks ja neid ignoreeritakse töö käigus.

Märgendamata fail peab sisaldama iga sõna ja kirjavahemärki eraldi real. Iga real olev sõna peab olema terviklik, s.t ei tohi sisaldada tühikuid ega tabulaatorsümboleid, sest programm ignoreerib kõiki sümboleid, mis neile järgnevad. Fail võib sisaldada ka tühje ridu, neid loetakse programmi poolt lause või lõigu piirideks.

Märgendatud faili formaat on sarnane märgendamata faili formaadiga. Lisandunud on iga rea kohta teine veerg. Veerud on eraldatud mistahes arvu tabulaatorsümbolitega või tühikutega. Esimene veerg sisaldab sõnu ja kirjavahemärke, teine veerg sisaldab märgendit. Kõike peale kahe esimese veeru ignoreeritakse. Ka märgendatud failis tähistab % märk rea alguses kommentaari. Fail võib sisaldada tühje ridu, kuid ei tohi sisaldada ühe veeruga ridu.

Leksikonfail luuakse parameetrite genereerimise sammul tnt-para programmi töö käigus. See sisaldab sõnade ja nende märgendite sagedusi nii, nagu need esinesid treeningkorpuses. Neid sagedusi kasutatakse märgendamisel leksikaalsete tõenäosuste määramiseks. Iga rida leksikonfailis sisaldab nelja või rohkemat veergu, mis on eradatud tabulaatorsümboliga. Esimene veerg sisaldab sõna, teine rida sisaldab selle sõna sagedust treeningkorpuses. Järgnevad veerud sisaldavad antud sõna korpuses esinenud erinevaid märgendamisvõimalusi koos esinemiste arvuga. Kõikide märgendite esinemiste summa on võrdne teises veerus asuva arvuga. Leksikon võib sisaldada ka erisümboleid, mis informeerivad märgendamisprogrammi, kuidas töödelda erinevaid märgendite klasse või panna paika töötlemise detaile. Need märgid algavad sümboliga @. Näiteks leksikonirida

@CARDPUNCT 527 ADJA 488 ADV 32 CARD 3 TRUNC 4 tähistab, et number koos temale järgneva punktiga esines korpuses 527 korda, millest 488-l korral oli märgendiks ADJA, ADV 32 korral jne. Näide on pärit leksikonist, mis genereeriti kasutades Saksa NEGRA korpust.

N-grammide fail luuakse samuti parameetrite loomise sammul tnt-para programmi poolt. See sisaldab uni-, bi- ja trigrammide kontekstisagedusi. N-grammide rida sisaldab kahte (unigrammid), kolme (bigrammid) või nelja (trigrammid) veergu. Kõik peale viimase veeru sisaldavad märgendeid, viimane veerg sisaldab selle konkreetse n-grammi sagedust korpuses.

TnT märgendaja kasutamine koosneb kahest etapist:

1. Parameetrite genereerimine
2. Märgendamine

Esimese etapi käigus luuakse märgendatud treeningkorpuse baasil parameetrite mudel. Teise etapi käigus rakendatakse saadud mudelit uue teksti märgendamiseks.

(Brants, 1999)

6.3.2.Parameetrite genereerimine: tnt-para

Parameetrite genereerimiseks on vajalik eelnevalt märgendatud treeningkorpuse olemasolu eelkirjeldatud formaadis. Vaikimisi genereerib programm leksikaalsed sagedused ja kontekstisagedused vastavalt treeningkorpusele ja salvestab need kahe failina töökataloogi. Failidenimed on samad kui korpuse failil, aga laiendid on vastavalt .lex ja .123.

Vaikeväärtusena kasutab TnT Tagger trigramme, vaikimisi sufiksi pikkus on 10.

Tnt Tagger võimaldab kasutada erinevaid parameetreid ja lippe:

-i ignoreeri suur- ja väiketähti, kõik suurte tähtedega sõnad konverteeritakse väiketähtedeks, kõik sõnad leksikonis on väikeste tähtedega

-c kodeerib suurtähe esinemised, lisab märgenditele c kui tegemist on suure tähega algava sõnaga, ja l-i väiketähega algavatele sõnadele

-Nn genereeri n-grammi (n=1..3)

-l genereeri ainult leksikon, vaikimisi luuakse leksikon ja n-gramm fail, antud lipp võimaldab genereerida ainult leksikoni (Brants, 1999)

6.3.3.Märgendamine: tnt

Märgendamiseks on vajalikud leksikon- ja n-grammfailid ja eelpool kirjeldatud kujul sisendfail. Märgendaja kasutab ainult esimest veergu sisendfailist, ülejäänut ignoreeritakse.

Tnt võimaldab kasutada järgmisi seadistusvõimalusi:

-apikkus, kus pikkus määrab ära maksimaalse sufiksi pikkuse tundmatute sõnade jaoks

-b*fail* kasuta *fail*-i tagavara leksikonina, st kui sõna ei leita leksikonist, siis püütakse seda otsida kõigepealt tagavaraleksikonist, enne kui rakendatakse tundmatute sõnade märgendamise protseduuri.

-n*n* kasuta märgendamiseks *n*-gramme, *n*=1,2,3, vaikimisi –n3

-u*mode* kasuta meetodit *mode* tundmatute sõnade puhul, *mode* on üks järgmistest:

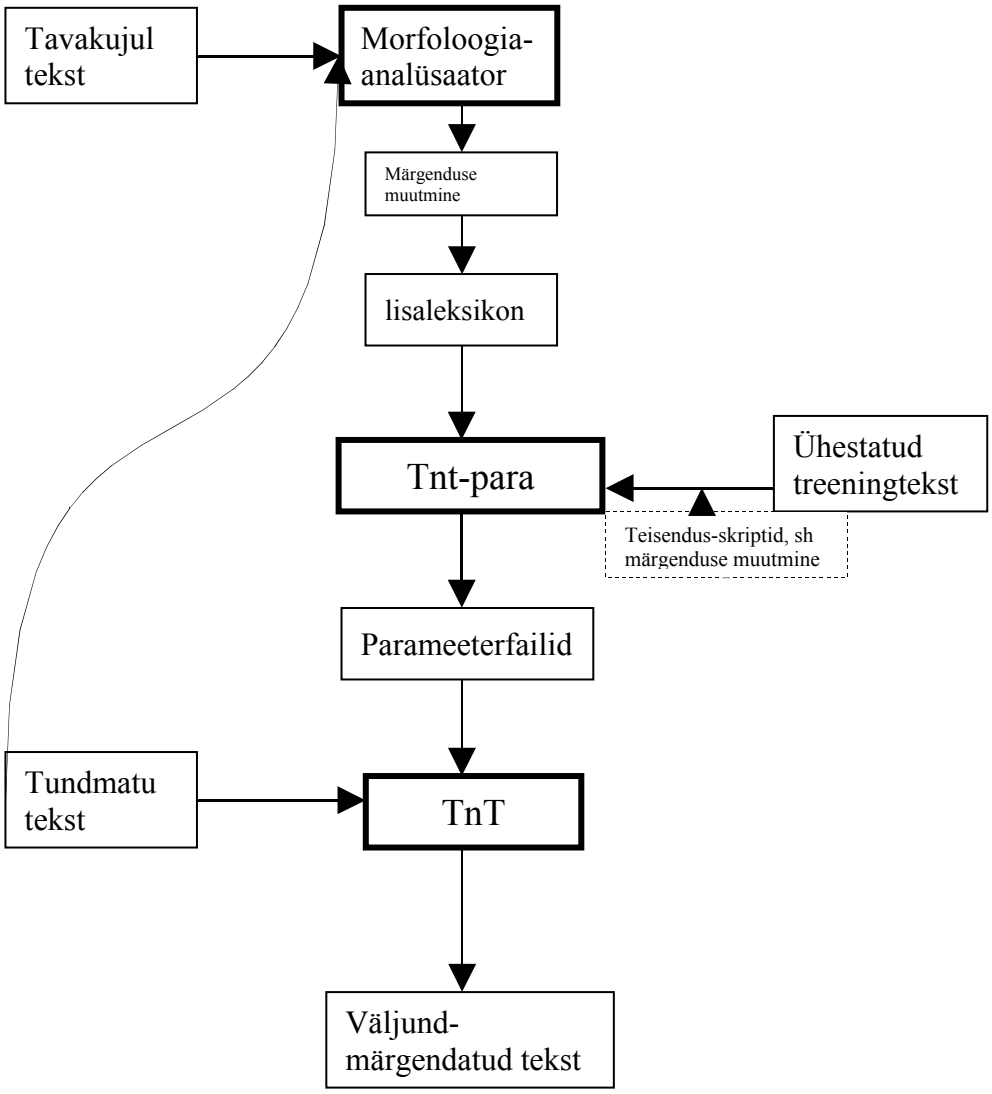
0 tundmatuid sõnu ei lubata, programm katkestatakse veateatega;

1 kasuta leksikoni kannet @UNKNOWN,et leida tundmatute sõnade leksikaalseid tõenäosusi;

2 kombineeri tundmatute sõnade puhul kõikide sõnade statistikat;

3 kombineeri tundmatute sõnade puhul ainult ühe korra esinevate sõnade statistikat;

Programmipaketiga TnTTagger on kaasas ka programm tnt-diff failide erinevuste leidmiseks, programm annab välja erinevuste ja sarnasuste arvud ning suhted protsentides. (Brants, 1999)



Joonis 6.9 TnTTaggeri sisendid ja väljundid

7. Programmide rakendamine

7.1. Olukord

Korpusena on antud magistritöös kasutusel suur osa eesti keele morfoloogiliselt ühestatud korpusest. Antud uurimuse jaoks moodustatud korpus koosneb ilu- ja ajakirjanduse tekstidest, G.Orwelli raamatust „1984“ ja infotekstidest, teised tekstid on liiga spetsiifilised ja võiksid tulemust seega liialt mõjutada.

Tekst	Ridu	Jaotus
Ilukirjandus	127 651	35,56%
Ajalehetekstid	131 187	36,55%
1984	94 905	26,44%
Infotekstid	5208	1,45%
Kokku	358 951	100%

Tabel 7.1 Tekstide jaotus kasutatavas korpuses

Moodustatud korpus tuli jagada kaheks osaks, treeningkorpuseks, mille alusel statistilisi märgendajaid treenida, ja testkorpuseks, mille alusel märgendajate töö õigsust kontrollida. Kogu korpus sisaldab 358 951 rida. Treeningfail koosneb 319 519 reast TEI-märgistusest puhastatud tekstist. Testkorpus 39 444 reast. Korpused on koostatud nii, et kogu korpusest on testkorpusesse võetud iga kümnes lause, ülejäänud laused treeningkorpusesse. See tagab, et tulemusi ei mõjuta tekstide žanrilised erinevused.

7.2. Töötlus

Selleks, et kasutada statistilisi ühestajaid TreeTagger ja TnTTagger, on vaja tekstid töödelda sobivale kujule. Mõlemad programmid vajavad treenimiseks korpust, milles iga sõna on eraldi real, sõnale järgneb tabulaator ja seejärel on sõna märgend.

```
<s>
V&otilde;itlus    v&otilde;itlus+0 //_S_ com sg nom //
k&otilde;nnib    k&otilde;ndi+b //_V_ main indic pres ps3 sg ps af //
sellest    see+st //_P_ sg el //
eestlasest    eestlane+st //_S_ com sg el //
m&ouml;&ouml;da    m&ouml;&ouml;da+0 //_D_ //
.    . //_Z_ Fst //
```

```
</s>
```

Joonis 7.1 Näide ühestatud töötlemata failist

```
Võitlus     //_S_sg_nom//
kõnnib      //_V_main_indic_pres_ps3_sg_ps_af//
sellest     //_P_sg_el//
eestlasest  //_S_sg_el//
mööda //_JD_//
.     SENT
```

Joonis 7.2 Näide samast lausest töödelduna sobivale kujule, muudetud on ka märgendust

Ühestajate jaoks sobiva kuju saamiseks tuli eemaldada lause algus- ja lõpumärgendid (<s> ja </s>), eemaldada sõna tüvi ja muuta SGML- kujul olevad tähemärgid tavakujule.

Samuti tuli kindlaks määrata esialgne märgendite süsteem. Olemasoleva morfoloogiliselt ühestatud korpuse sõnade morfoloogiline informatsioon koosneb erinevate märgendite kombinatsioonist, nimisõnadel näiteks sõnaliik, arv, kääne, verbidel aga sõnaliik, kõneviis, aeg, isik, arv, kõne, valentsimärgendid. Kui need märgendite kombinatsioonid asendada ühe erineva märgendiga, saaksime üle 800 märgendi. Ükski statistiline ühestaja ei suuda sellise märgendite hulgaga toime tulla. Praktikas kasutatakse teiste keelte puhul 30-150 märgendit, äärmisel juhul kuni 300 märgendit.

7.3. Treetagger

TreeTagger ei võimalda kasutada morfoloogiaanalüsaatorit otseselt, vaid vajab leksikonfaili, mis sisaldab sõnavormi ja selle vormi kõiki märgendeid. See tuli genereerida ühestatud korpuseosa põhjal.
Igale korpuses olevale sõnavormile leiti kõik korpuses esinevad märgendusvõimalused.
Lõpuks sorteeriti kogu tulemus tähestikulisse järjekorda.
Tekstide töötlemiseks kasutati Perli skripte.

```
pealetung     //_S_sg_nom// -
```

```
pealetungi      //_S_sg_adit// -       //_S_sg_gen// -       //_S_sg_part// -
pealetungiga    //_S_sg_kom// -
pealetükkiv     //_A_sg_nom// -       //_S_sg_nom// -
pealetükkivalt  //_A_sg_abl// -       //_JD_// -     //_S_sg_abl// -
pealetükkivat   //_A_sg_part// -      //_S_sg_part// -
pealiiniga      //_S_sg_kom// -
pealinikuid     //_S_pl_part// -
pealinn //_S_sg_nom// -
pealinna        //_S_sg_adit// -      //_S_sg_gen// -       //_S_sg_part// -
```

Joonis 7.3 Katke leksikonfailist

TreeTagger vajab treenimiseks ka Openclass faili, mis sisaldab nimekirja morfoloogilistest märgenditest, mille hulgast valitakse märgend sõnale, mida leksikonis ei leidu.

Treenimis-etapi tulemusel genereeris train-tree-tagger faili ehk keelemudeli, mida kasutatakse tree-taggeri töös morfoloogiliselt analüüsimata teksti ühestamisel. Tree-taggeri ehk märgendamisetapi programmi sisendfailiks on tekstifail, milles iga sõna ja kirjavahemärk on eraldi real. Väljundfailina saadakse tekst, millele on lisatud sõnaliigi märgend.

```
Tänav
tegi
järsu
pöörde
ja
lõppes
trepiga
,
mis
viis
alla
kõrvaltänavale
,
kus
putkades
müüdi
,närtsinud
välimusega
juurvilja
.
```

Tänav	// S_sg_nom//
tegi	// V_main_indic_impf_ps3_sg_ps_af//
järsu	// A_sg_gen//
pöörde	// S_sg_gen//
ja	// JD_//
lõppes	// V_main_indic_impf_ps3_sg_ps_af//
trepiga	// S_sg_kom//
,	SENT
mis	// P_sg_nom//
viis	// V_main_indic_impf_ps3_sg_ps_af//
alla	// JD_//
kõrvaltänava-le	// S_sg_all//
,	SENT
kus	// JD_//
putkades	// S_pl_in//
müüdi	// V_main_indic_impf_imps_af//
närtsinud	// A_pos//
välimusega	// S_sg_kom//
juurvilja	// S_sg_part//
.	SENT

Joonis 7.4 Näide treeningfaili sisendist

Tabel 7.2 Näide samast lausest TreeTaggeri poolt ühestatuna, TreeTagger märgendas lause korrektselt

7.4. TnTTagger

TnTTagger vajab samuti treeningfaili, mis sisaldab ühte sõna real koos märgendiga. Märgendaja moodustab treeningfaili alusel leksikoni kõikidest korpuses asuvatest sõnadest ja nende kõikidest korpuses esinevatest märgendusvõimalustest koos esinemissagedustega. Samuti moodustab märgendaja treenimisprotsessi käigus n-grammide faili, mis sisaldab uni-, bi- ja trigramme koos konteksti alusel arvutatud sagedustega.

Tänav	//_S_sg_nom//
tegi	//_V_main_indic_impf_ps3_sg_ps_af//
järsu	//_A_sg_gen//
pöörde	//_S_sg_gen//
ja	//_JD_//
lõppes	//_V_main_indic_impf_ps3_sg_ps_af//
trepiga	//_S_sg_kom//
,	SENT
mis	//_P_sg_nom//
viis	//_V_main_indic_impf_ps3_sg_ps_af//
alla	//_JD_//
kõrvaltänava-le	//_S_sg_all//
,	SENT
kus	//_JD_//
putkades	//_S_pl_in//
müüdi	//_V_main_indic_impf_imps_af//
närtsinud	//_A_pos//
välimusega	//_S_sg_kom//
juurvilja	//_S_sg_adit//
.	SENT

Tabel 7.3 Näide Eelolevast näitelausest TnTTaggeri poolt ühestatuna, eelviimase sõna märgendamisel eksiti, õige märgend oleks olnud //_S_sg_part//

8. Katsed

8.1.Katsed optimaalse märgendusskeemi leidmiseks

Eesti keele morfoloogiliselt ühestatud korpuse märgendus koosneb ca 800 unikaalsest märgendikombinatsioonist. Selline märgendus on väga informatiivne, kuid seda on raske, et mitte öelda võimatu statistiliselt treenida. Üldiselt on teada, et mida suurem on märgendite hulk, seda rohkem peaks olema treeningmaterjali, näiteid, millelt õppida. Kui märgendeid on väga vähe, siis võib treeningkorpuse maht olla küll väiksem, kuid see märgendus ei pruugi olla piisavalt informatiivne edasiseks töötluseks ning võib juhtuda, et lakooniline märgendus takistab ka sobivate mustrite leidmist ning lõppkokkuvõttes on ühestamise kvaliteet halb.

Kirjandusest on teada, et inglise keele jaoks on põhiliselt kasutusel 87 märgendiga Browni märgendite hulk, 45 märgendiga Penn TreeBank märgendus ja 61 märgendiga C5 märgendihulk (Juravsky,2009:164). Kaalepi ja Vaino märgendisüsteem koosnes 88-st märgendist, Kajaste esialgne katsetus 15-st märgendist (Kajaste,2006). Kajaste 2006. aastal kirjutatud bakalaureusetöös kasutatud märgendite hulk oli ilmselgelt liiga minimaalne, ei eristatud käände- ja pöördeinfot, mitmed vead tekkisid just liiga väikese märgendite hulga tõttu, treenimise käigus omandatud mudel oli liiga ühekülgne ja ei andnud piisavalt detailset infot õigete järelduste tegemiseks.

Optimaalse märgendisüsteemi otsinguks oli sobivaim TnTTagger, sest sellega töötamine oli kõige hõlpsam. Programm suudab ise moodustada vajalikud failid, on vaja vaid õigel kujul märgendatud treeningfaili ja hiljem testimiseks samade märgenditega märgendatud teksti. Esialgse etapi eesmärgiks oli selgitada välja perspektiivikamad märgendussüsteemid, et nendega siis hiljem põhjalikumaid teste läbi viia kasutades erinevaid märgendajate võimalusi.

Erinevate märgendussüsteemide puhul olid aluseks kaks süsteemi: nn fs- ja kym-süsteemid. Fs-süsteem on kasutusel Filosofti morfoloogiaanalüsaatori väljundis. Kym-süsteemis on märgendatud morfoloogiliselt ühestatud korpuse tekstid.
Kõikide märgenduste kirjeldused on toodud magistritöö lisas 1 (cd-plaat).

1.Katse

Esimese katsetusena testiti fs-märgendust. Selle katse tulemused on toodud tabelis x. Fs-märgendussüsteemis oli 332 erinevat märgendit, neist leidus treeningkorpuses 289 ja testkorpuses 217. TnTtaggeri korrektsuseks selle märgendisüsteemiga saime 94.34% (vt tabel 8.1).

Võrdseid sõnu	37212	39443	94.34%
Erinevaid sõnu	2231	39443	5.66%

Tabel 8.1. Märgendamise korrektsus modifitseerimata fs-märgendisüsteemiga fs

Enamlevinud vigade ja valesti märgendatud sõnavormide statistika on toodud tabelites 8.2 ja 8.3.

Kordade arv	Vale	Õige
62	// S sg p//	// S sg g//
58	// S sg g//	// S sg n//
56	// K //	// D //
54	// S sg g//	// S sg p//
52	// J //	// D //
52	// H sg g//	// H sg n//
48	// P sg n//	// P sg g//
47	// S sg n//	// S sg g//
46	// D //	// K //

Tabel 8.2. 1.Katse enamlevinud märgendite vead

Kordade arv	Sõna	Vale	Õige
35	ta	// P sg n//	// P sg g//
34	on	// V b//	// V vad//
19	siis	// D //	// J //
16	on	// V vad//	// V b//
12	üks	// N sg n//	// P sg n//
11	tema	// P sg g//	// P sg n//
11	kui	// J //	// D //
10	ei	// V neg//	// D //
9	tagasi	// D //	// K //

Tabel 8.3. 1.Katse enamlevinud valesti märgendatud sõnad

2.Katse

Esimeseks reaalseks katseks uut märgendust leida sai fs märgendusest mitmuse ja ainsuse eristamise eemaldamine. Selle tulemusena jäi treeningkorpusesse 204 märgendit ja testkorpusesse 159 märgendit. Tulemuseks korrektsus 94,36% (vt tabel 8.4). Täpsemad andmed tulemuste statistika kohta on tabelites 8.5 ja 8.6.

Võrdseid sõnu	37220	39443	94.36%
Erinevaid sõnu	2223	39443	5.64%

Tabel 8.4. Märgendamise korrektsus modifitseeritud fs-märgendisüsteemiga fs1

Kordade arv	Vale	Õige
100	// V b//	// V vad//
60	// S g//	// S n//
59	// S p//	// S g//
58	// S g//	// S p//
56	// H g//	// H n//
55	// P n//	// P g//

Kordade arv	Vale	Õige
99 on	// V b//	// V vad//
37 ta	// P n//	// P g//
19 siis	// D //	// J //
14 üks	// N n//	// P n//
11 tema	// P g//	// P n//
11 kui	// J //	// D //

52	//_K_//	//_D_//
50	//_J_//	//_D_//
49	//_S_n//	//_S_g//

Tabel 8.5. 2.Katse enamlevinud märgendite vead

10 siis	//_J_//	//_D_//
10 ei	//_V_neg//	//_D_//
10 Ta	//_P_n//	//_P_g//

Tabel 8.6. 2.Katse enamlevinud valesti märgendatud sõnad

Korrektsus paranes 0.02%, kuid on verbivormi analüüsil vigade hulk suurenes märgatavalt.

3. Katse

Järgmise üritusena sai eelmise katse märgendusest eemaldatud omadussõnade alg-, kesk-, ülivõrde eristamine, kuna kesk- ja ülivõrre esinevad piisavalt harva ning nad ei käitu süntaktiliselt eriti teistmoodi algvõrdes omadussõnadest. Muudatuse tulemusena jäi treeningkorpusesse 183 märgendit ja testkorpusesse 143. Tulemused on toodud tabelis 8.7.

Võrdseid sõnu	37232	39443	94.39%
Erinevaid sõnu	2211	39443	5.61%

Tabel 8.7. Märgendamise korrektsus modifitseeritud fs-märgendisüsteemiga fs2

Tulemused paranesid marginaalselt.

Kordade arv	Vale	Õige
101	//_V_b//	//_V_vad//
61	//_S_g//	//_S_n//
59	//_S_p//	//_S_g//
58	//_S_g//	//_S_p//
55	//_H_g//	//_H_n//
54	//_P_n//	//_P_g//
53	//_K_//	//_D_//
51	//_J_//	//_D_//
48	//_S_n//	//_S_g//
45	//_D_//	//_K_//

Tabel 8.8. 3.Katse enamlevinud märgendite vead

Kordade arv	Vale	Õige
100 on	//_V_b//	//_V_vad//
37 ta	//_P_n//	//_P_g//
19 siis	//_D_//	//_J_//
15 üks	//_N_n//	//_P_n//
11 tema	//_P_g//	//_P_n//
11 kui	//_J_//	//_D_//
10 ei	//_V_neg//	//_D_//
10 Ta	//_P_n//	//_P_g//
9 tagasi	//_D_//	//_K_//
9 siis	//_J_//	//_D_//

Tabel 8.9. 3.Katse enamlevinud valesti märgendatud sõnad

4.Katse

Neljandaks üritasin vähendada märgendite hulka veelgi, selleks võtsin omadus-, arv- ja nimissõnad kokku üheks märgendiks (_S_), alles jäi käändeinfo, lisaks veel kõik eelmised muudatused. Märgendeid treeningkorpusesse jäi 110 ja testkorpusesse 89. Märgendamise tulemused paranesid 94,72%-ni, vt tabel 8.10.

Võrdseid sõnu	37359	39443	94.72%
Erinevaid sõnu	2084	39443	5.28%

Tabel 8.10. Märgendamise korrektsus modifitseeritud fs-märgendisüsteemiga fs3

Kordade arv	Vale	Õige
174	//_S_g//	//_S_n//
155	//_S_n//	//_S_g//
102	//_V_b//	//_V_vad//
84	//_S_g//	//_S_p//
75	//_S_p//	//_S_g//
57	//_K_//	//_D_//
53	//_P_n//	//_P_g//
51	//_S_n//	//_S_p//
51	//_J_//	//_D_//
46	//_S_p//	//_S_n//

Tabel 8.11. 4.Katse enamlevinud märgendite vead

Kordade arv	Vale	Õige
101 on	//_V_b//	//_V_vad//
38 ta	//_P_n//	//_P_g//
19 siis	//_D_//	//_J_//
16 üks	//_S_n//	//_P_n//
11 ühe	//_P_g//	//_S_g//
11 tema	//_P_g//	//_P_n//
11 tagasi	//_D_//	//_K_//
11 kui	//_J_//	//_D_//
10 ei	//_V_neg//	//_D_//
10 Ta	//_P_n//	//_P_g//

Tabel 8.12. 4.Katse enamlevinud valesti märgendatud sõnad

Selline märgendisüsteem võib olla kasulik käänete tuvastamiseks.

5.Katse

Järgmisel katsel jäi märgendusse alles arv, eemaldatud sai omadussõna võrded ja asendatud lühendite tunnus Y üldise nimisõna märgendiga S. Treeningkorpuses oli 234 märgendit, testkorpuses 198. Katse tulemused tabelis .

Võrdseid sõnu	37228	39443	94.38%
Erinevaid sõnu	2215	39443	5.62%

Tabel 8.13. Märgendamise korrektsus modifitseeritud fs-märgendisüsteemiga fs4

Kordade arv	Vale	Õige
59	//_S_sg_p//	//_S_sg_g//
58	//_S_sg_g//	//_S_sg_n//
58	//_K_//	//_D_//
55	//_S_sg_g//	//_S_sg_p//
52	//_J_//	//_D_//
52	//_H_sg_g//	//_H_sg_n//
49	//_P_sg_n//	//_P_sg_g//
48	//_D_//	//_K_//
47	//_S_sg_n//	//_S_sg_g//

Kordade arv	Vale	Õige
36 ta	//_P_sg_n//	//_P_sg_g//
34 on	//_V_b//	//_V_vad//
19 siis	//_D_//	//_J_//
17 on	//_V_vad//	//_V_b//
12 üks	//_N_sg_n//	//_P_sg_n//
11 tema	//_P_sg_g//	//_P_sg_n//
11 kui	//_J_//	//_D_//
10 ei	//_V_neg//	//_D_//
9 tagasi	//_D_//	//_K_//

42	// _H_ sg _g//	// _S_ sg _g//

Tabel 8.14. 5.Katse enamlevinud märgendite vead

8 üle	// _D_ //	// _K_ //

Tabel 8.15. 5.Katse enamlevinud valesti märgendatud sõnad

6.Katse

Samuti sai tehtud katse ka puhta kym märgnedusega. See erineb fs-süsteemist selle poolest, et eristatakse detailsemalt verbide erinevaid vorme, samuti kirjavahemärke. Märgendus on informatiivsem ja rikkalikum, treeningkorpuses leidus 402 erinevat märgendit, testkorpuses 352 (vt tabel 8.16). Samu märgendeid kasutades oli märgendaja korrektuss 93.46% (vt tabel 8.16)

Võrdseid sõnu	36865	39443	93.46%
Erinevaid sõnu	2578	39443	6.54%

Tabel 8.16. Märgendamise korrektsus modifitseerimata kym-märgendisüsteemiga

Kordade arv	Vale	Õige
68	// _V_ main _indic _impf _ps3 _sg _ps _af//	// _V_ aux _indic _impf _ps3 _sg _ps _af//
64	// _S_ com _sg _part//	// _S_ com _sg _gen//
60	// _S_ com _sg _gen//	// _S_ com _sg _nom//
54	// _S_ com _sg _gen//	// _S_ com _sg _part//
50	// _S_ prop _sg _gen//	// _S_ prop _sg _nom//
48	// _S_ com _sg _nom//	// _S_ com _sg _gen//
48	// _P_ sg _nom//	// _P_ sg _gen//
46	// _J_ sub//	// _D_ //
45	// _V_ main _indic _pres _ps3 _sg _ps _af//	// _V_ aux _indic _pres _ps3 _sg _ps _af//
45	// _S_ prop _sg _nom//	// _S_ prop _sg _gen//
40	// _S_ com _sg _nom//	// _S_ com _sg _part//

Tabel 8.17. 6.Katse enamlevinud märgendite vead

Vigade analüüs näitab, et kõige raskem on eristada abitegusõna olema põhiverbist, st otsustada, kas olema vorm moodustab verbi liitja koos partitsiibiga või on iseseisev verb. Samuti põhjustab raskusi kolme esimese käände määramine ning alistava sidesõna ja määrsõna eristamine.

Kordade arv	Vale	Õige
66 oli	// V main indic impf ps3 sg ps af//	// V aux indic impf ps3 sg ps af//
43 on	// V main indic pres ps3 sg ps af//	// V aux indic pres ps3 sg ps af//
35 ta	// P sg nom//	// P sg gen//
26 on	// V main indic pres ps3 sg ps af//	// V main indic pres ps3 pl ps af//
24 siis	// J sub//	// D //
21 pole	// V aux indic pres ps neg//	// V main indic pres ps neg//
16 on	// V aux indic pres ps3 sg ps af//	// V main indic pres ps3 sg ps af//
14 oli	// V aux indic impf ps3 sg ps af//	// V main indic impf ps3 sg ps af//
11 vaid	// J crd//	// D //
10 üks	// N card sg nom l//	// P sg nom//
10 siis	// D //	// J sub//

Tabel 8.18. 6.Katse enamlevinud valesti märgendatud sõnad

7.Katse

Katse sai tehtud Kaalepi ja Vaino statistilise ühestaja katse käigus moodustatud märgenduse abil, mis sisaldas 88 erinevat märgendit. (Kaalep ja Vaino, 1998) Katse tulemuseks oli 5,08% vigaseid analüüse, vt tabel 8.19.

Võrdseid sõnu	37440	39443	94.92%
Erinevaid sõnu	2003	39443	5.08%

Tabel 8.19. Märgendamise korrektsus Kaalep ja Vaino 1998 märgendite hulgaga stat

Kordade arv	Vale	Õige
59	//NCSG//	//NCSN//
56	//VMAS//	//ASX//
55	//NCS1//	//NCSG//
54	//PP1SN//	//PP1SG//
54	//NCSG//	//NCS1//
48	//CS//	//RR//
47	//NPSG//	//NPSN//
47	//NCSN//	//NCSG//
43	//NPSG//	//NCSG//
43	//NCSN//	//NCS1//
39	//NPSN//	//NPSG//

Tabel 8.20. 7.Katse enamlevinud märgendite vead

Kordade arv	Vale	Õige
36 ta	//PP1SN//	//PP1SG//
26 siis	//CS//	//RR//
18 üks	//MCSN//	//PP1SN//
11 kui	//CS//	//RR//
10 siis	//RR//	//CS//
10 Ta	//PP1SN//	//PP1SG//
9 vaid	//CC//	//RR//
8 tagasi	//RR//	//ST//
8 ei	//VME//	//RR//
7 ühe	//PP1SG//	//MCSG//
7 tema	//PP1SG//	//PP1SN//

Tabel 8.21. 7.Katse enamlevinud valesti märgendatud sõnad

8.Katse

Vastavalt vigadele sai märgendusse sisse viidud muudatused: pärisnimede ja nimisõnade eristamine ei ole oluline, seega sai märgendusest eemaldatud S com ja S prop, mille asemel jäi ainult S. Kuna J ja D oli sageli eristamatu, siis sai pandud nendele sõnadele ühiseks märgendiks _JD_. Kindlasti ei ole omadussõnade puhul olulised võrded, seega ei eristata ka neid. Järgarvu tähistasin omadussõnaga sama märgendiga, oluline ei ole eristada rooma ja tavanumbreid. Kuna genitiivatribuuti on väga raske märgendada ja otsustada, tekkis kahtlus, et ka korpuses võib vigu olla, seega sai ära kaotatud G märgend, selle asemele sai igal pool S sg gen. Aluseks oli fs märgendus, treeningkorpuses 206 märgendit ja testkorpuses 153 märgendit, tulemuseks 94,75%(vt tabel 8.22).

Võrdseid sõnu	37371	39443	94.75%
Erinevaid sõnu	2072	39443	5.25%

Tabel 8.22. Märgendamise korrektsus modifitseeritud fs-märgendite hulgaga fs5

Kordade arv	Vale	Õige
111	// S sg n//	// S sg g//
68	// K //	// JD //
66	// S sg g//	// S sg p//
57	// S sg p//	// S sg g//
50	// JD //	// K //
47	// P sg n//	// P sg g//
42	// S sg n//	// S sg p//
38	// V b//	// V vad//
30	// S sg p//	// S sg n//
29	// V nud//	// V neg nud//

Tabel 8.23. 8.Katse enamlevinud märgendite vead

Kordade arv	Vale	Õige
35 ta	// P sg n//	// P sg g//
15 on	// V vad//	// V b//
11 tema	// P sg g//	// P sg n//
10 üks	// N sg n//	// P sg n//
10 ei	// V neg//	// JD //
8 ühe	// P sg g//	// N sg g//
8 tagasi	// JD //	// K //
8 olnud	// V neg nud//	// V nud//
8 mis	// P sg n//	// P pl n//
8 Ta	// P sg n//	// P sg g//

Tabel 8.24. 9.Katse enamlevinud valesti märgendatud sõnad

9. **Katse**

Samad muudatused, mis eelmisel katsel, aga aluseks oli kym märgendus. Märgendeid treeningkorpuses 317 ja testkorpuses 228. Tulemuseks korrektsus 93,99%, vt tabel 8.25.

Võrdseid sõnu	37074	39443	93.99%
Erinevaid sõnu	2369	39443	6.01%

Tabel 8.25. Märgendamise korrektsus modifitseeritud kym-märgendite hulgaga kym1

Kordade arv	Vale	Õige
154	//_V_main_indic_impf_ps3_sg_ps_af//	//_V_aux_indic_impf_ps3_sg_ps_af//
145	//_S_sg_nom//	//_A_sg_nom//
143	//_S_sg_nom//	//_S_pl_nom//
123	//_S_sg_nom//	//_S_sg_gen//
110	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_sg_ps_af//
93	//_S_sg_gen//	//_S_pl_gen//
85	//_S_sg_gen//	//_S_sg_part//
83	//_S_sg_nom//	//_S_pl_gen//

Tabel 8.26 9. Katse enamlevinud märgendite vead TreeTaggeriga

Kordade arv	Vale	Õige
109 on	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_sg_ps_af//
71 on	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_main_indic_pres_ps3_pl_ps_af//
38 pole	//_V_aux_indic_pres_ps_neg//	//_V_main_indic_pres_ps_neg//
36 ta	//_P_sg_nom//	//_P_sg_gen//
25 kõik	//_P_pl_nom//	//_P_sg_nom//
21 on	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_pl_ps_af//
21 olid	//_V_main_indic_impf_ps3_pl_ps_af//	//_V_aux_indic_impf_ps3_pl_ps_af//

Tabel 8.27 9. Katse enamlevinud valesti märgendatud sõnad TreeTaggeriga

Kuna neg märgenditega tekkis palju vigu, siis sai proovitud panna „ära“ ja „ei“le eraldi märgend, samuti on raske eristada, millal „üks“ on arv ja millal asesõna, seega sai pandud ka sellele sõnale eraldi märgend. Need kaks viimast muudatust tõid ainult 22 vea vähenemise, mistõttu nendest muudatustest loobuti.

10. Katse

Viimase katsena prooviti lisada treeningfaili alusel moodustatud leksikonfailile lisaleksikon. Selle moodustamiseks tuli ühestatud failidest moodustatud korpus viia tagasi kujule, kus tekst poleks enam üks sõna ühel real, samuti tuli eemaldada morfoloogiline info. Selliselt puhastatud tekst anti morfoloogilisele analüsaatorile analüüsimiseks. Morfoloogiliseks analüüsiks kasutasin programmi ESTMORF (Kaalep,1999). Morfoloogia-analüsaatori väljundfail tuli omakorda töödelda leksikonfailiks. Selleks tuli morfoloogiliselt analüüsitud ühestamata tekst teisendada kujule, kus sõna oleks ühel real koos kõigi erinevate tõlgendustega. Kym-märgendusega võrreldes oli mitmeid muudatusi: nagu juba eelmistes katsetes ei tehta vahet pärisnimede ta tavaliste nimisõnade vahel, mitmete verbivormide puhul ei eristata ainsust ja mitmust, ei eristata omadussõna võrdeid

ja genitiivatribuuti, ei eristata arve ja lühendeid. Selline lisaleksikon simuleerib olukorda, nagu oleks morfoloogiaanalüsaator kaasatud analüüsi ja seetõttu peaks vähenema tundmatute sõnade hulk. Treeningkorpuses oli 248 märgendit ja testkorpuses 189 märgendit. Katsed viisin läbi nii TreeTaggeri, kui TnTTaggeriga. TreeTaggeri tarbeks moodustasin vastavate märgenditega varustatud test- ja treeningkorpuse alusel lekskonfaili. TnTTagger moodustab ise treeningfaili alusel leksikoni, kuid tulemuste parandamiseks saab kasutada lisaleksikoni, mille moodustasin kogu korpuse alusel. Selline lisaleksikoni lisamine annab TnTTaggerile TreeTaggeriga sarnasema ja võrdsema algpositsiooni.

	TnTTagger	TreeTagger
Võrdseid sõnu	94.69%	90.44%
Erinevaid sõnu	5.31%	9.56%

Tabel 8.28. Märgendamise korrektsus modifitseeritud kym-märgendite hulga ja lisaleksikoniga - kym1-leks

Kordade arv	Vale	Õige
123 oli	// V main indic impf ps3 sg ps af//	// V aux indic impf ps3 sg ps af//
102 on	// V main indic pres ps3 sg ps af//	// V aux indic pres ps3 sg ps af//
51 oleks	// V aux cond pres ps neg//	// V aux cond pres ps3 pl ps af//
50 on	// V main indic pres ps3 sg ps af//	// V main indic pres ps3 pl ps af//
32 ta	// P sg nom//	// P sg gen//
32 pole	// V aux indic pres ps neg//	// V main indic pres ps neg//
28 kaks	// S sg tr//	// A sg nom//
22 olid	// V main indic impf ps3 pl ps af//	// V aux indic impf ps3 pl ps af//
21 kõik	// P pl nom//	// P sg nom//
19 kolm	// S sg nom//	// A sg nom//
17 üks	// P sg nom//	// A sg nom//

Tabel 8.29. 10. Katse enamlevinud valesti märgendatud sõnad TreeTaggeriga

90 oli	// V main indic impf ps3 sg ps af//	// V aux indic impf ps3 sg ps af//
64 on	// V main indic pres ps3 sg ps af//	// V aux indic pres ps3 sg ps af//
39 ta	// P sg nom//	// P sg gen//
27 on	// V main indic pres ps3 sg ps af//	// V main indic pres ps3 pl ps af//
23 üks	// A sg nom//	// P sg nom//
16 pole	// V aux indic pres ps neg//	// V main indic pres ps neg//
16 olid	// V main indic impf ps3 pl ps af//	// V aux indic impf ps3 pl ps af//
14 mis	// P sg nom//	// P pl nom//
14 kes	// P sg nom//	// P pl nom//

Tabel 8.30. 10. Katse enamlevinud valesti märgendatud sõnad TnTTaggeriga

Kordade arv	Vale	Õige
155	//_S_sg_gen//	//_S_sg_nom//
149	//_S_sg_nom//	//_S_sg_gen//
126	//_V_main_indic_impf_ps3_sg_ps_af//	//_V_aux_indic_impf_ps3_sg_ps_af//
103	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_sg_ps_af//
88	//_S_sg_nom//	//_Y_?//
87	//_S_sg_nom//	//_A_sg_nom//
75	//_K_pre//	//_JD_//
72	//_S_sg_part//	//_S_sg_gen//
65	//_S_sg_nom//	//_S_sg_part//
55	//_S_sg_gen//	//_S_sg_part//

Tabel 8.31. 10.Katse enamlevinud märgendite vead TreeTaggeriga

121	//_S_sg_gen//	//_S_sg_nom//
107	//_S_sg_nom//	//_S_sg_gen//
93	//_V_main_indic_impf_ps3_sg_ps_af//	//_V_aux_indic_impf_ps3_sg_ps_af//
66	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_sg_ps_af//
61	//_K_post//	//_JD_//
52	//_P_sg_nom//	//_P_sg_gen//
50	//_S_sg_part//	//_S_sg_gen//
50	//_S_sg_adit//	//_S_sg_part//
47	//_S_sg_nom//	//_S_sg_part//

Tabel 8.32. 10.Katse enamlevinud märgendite vead TnTTaggeriga

Erinevate märgendite süsteemidega katsete läbiviimise eesmärgiks oli leida maksimaalse granulaarsusega märgendus-süsteem, mis annaks statistilise märgendajaga võimalikult häid tulemusi. Valituks osutus 9.katse märgendus-süsteem, mille aluseks oli kym-märgendus ja ei eristatud pärisnimesid ning nimisõnu, sidesõnad ja määrsõnad said ühise märgendi JD. Ei eristatud omadussõnade võrdeid, genitiivatribuuti ja järgarvud said omadussõnaga sama märgendi, samuti ei ole oluline eristada rooma ja tavanumbreid.

8.2.Katsed programmide erinevate seadistusvõimalustega

Nii TreeTagger, kui TnTTagger pakuvad mitmeid võimalusi vaikeväärtuste muutmiseks. TreeTaggeri puhul katsetasin treeningprogrammi puhul parameetrit -cl ehk konteksti pikkust ja märgendamisosa juures järgmisi parameetreid: -dtg ehk minimaalne

otsustuspuu kasv, -ecw ehk ekvivalentsusklassi kaal ja –atg ehk affiksipuu kasv. TnTTaggeri puhul sai proovitud treenimisprogrammi puhul parameetreid –i ja –c, ehk siis vastavalt ignoreeri suur- ja väiketähtede esinemisi ning kodeeri suurtähed. Märgendamise osas kasutasin parameetreid –n ehk sai määrata millise n väärtusega n-grammiga on tegemist, -a ehk sai määrata tundmatute sõnade puhul kasutatavat suffiksipikkust ja –u ehk tundmatute sõnade käsitlemise moodust.

	fs	fs1	fs2	fs3	fs4	kym	stat	fs5	kym1	kym1-leks
märgendite arv treening-failis	289	204	183	110	234	402	106	206	317	248
vaikeväärtused	84.69%	86.27%	86.29%	91.12%	84.39%	83.83%	87.21%	87.70%	86.50%	90.44%
train -cl 1	84.82%	86.33%	86.33%	91.04%	84.48%	83.95%	87.09%	87.85%	86.77%	**91.08%**
train -cl 3	84.61%	86.30%	86.32%	91.16%	84.31%	83.72%	**87.24%**	87.77%	86.60%	90.65%
–dtg 0.8	84.37%	86.12%	86.12%	90.99%	84.06%	83.23%	87.00%	87.43%	86.16%	90.03%
–ecw 0.18	84.69%	86.27%	86.29%	91.12%	84.40%	83.83%	87.21%	87.70%	86.51%	90.45%
–ecw 0.20	84.69%	86.27%	86.29%	91.13%	84.40%	83.83%	87.22%	87.70%	86.51%	90.45%
–atg 1	84.74%	86.31%	86.33%	91.10%	84.42%	83.91%	87.17%	87.76%	86.57%	90.47%
–atg 0.9	84.73%	86.32%	86.33%	91.12%	84.41%	83.98%	87.20%	87.78%	86.58%	90.48%
cl -3 -ecw 0.18 ja -atg 0.9	84.70%	86.33%	86.35%	**91.17%**	84.32%	83.88%	87.20%	**87.86%**	86.69%	90.68%

Tabel 8.33. Ülevaade erinevate programmi seadistusvõimalustega tehtud katsete tulemustest TreeTaggeriga

	fs	fs1	fs2	fs3	fs4	kym	stat	fs5	kym1	kym1-leks
märgendite arv tree-ningfailis	289	204	183	110	234	402	106	206	317	248
vaikeväärtused (tnt -a10 -d4 -n3 -u3)	94.34%	94.36%	94.39%	94.72%	94.38%	93.46%	94.92%	94.75%	93.99%	94.69%
tnt-para -i -c	93.91%	93.97%	93.99%	94.80%	93.95%	93.01%	94.14%	94.79%	93.99%	93.59%
tnt-para –i	93.91 %	93.92%	93.92%	94.70%	93.95%	93.08%	94.11%	94.74%	94.04%	94.78%
tnt-para –c	94.39%	94.41%	94.47%	94.88%	94.40%	93.55%	94.69%	94.89%	94.09%	93.66%
tnt -n2	94.06%	94.10%	94.12%	94.51%	94.10%	93.22%	94.34%	94.51%	93.70%	94.37%
tnt -u2	94.39%	94.41%	94.47%	94.88%	94.40%	93.55%	94.69%	94.89%	94.09%	94.78%
tnt -a8 -u2	94.41%	94.43%	94.49%	**94.90%**	94.41%	93.57%	94.71%	**94.90%**	94.11%	**94.78%**

Tabel 8.34. Ülevaade erinevate programmi seadistusvõimalustega tehtud katsete tulemustest TnTTaggeriga

8.3.Statistiliste märgendajate kombineerimine reeglipõhisega

Statistiliste märgendajate tulemuste parandamiseks otsustasime kombineerida statistilised märgendajad reeglipõhise ühestajaga. Erinevate märgendajate vead ei pruugi olla samad, seega võib erinevate märgendajate kombineerimine parandada märgendamise tulemusi (Wu, 2006). Kõige levinum lähenemine on töödelda samat lauset erinevate märgendajatega paralleelselt ja seejärel kombineerida nende tulemus hääletamise teel või teatud klassifikaatori alusel otsustada, millist märgendajat antud kontekstis usaldada. Teine võimalus on kasutada märgendajaid järjestikku. Sellist lähenemist on kasutatud tšehhi keele puhul, kasutades reeglipõhist lähenemist iga sõna ilmselgelt võimatute märgendite eemaldamiseks ja seejärel HMM märgendajat õige märgendi valimiseks.(Juravsky,2009)

Lähtuvalt eeldusest, et reeglipõhine märgendaja suudab anda korrektse ühese märgenduse vaid 1-2% vigade arvuga (Puolakainen,2001), oli esialgne plaan lähtuda märgendajate kombineerimisel just reeglipõhisest ühestajast ja ainult juhul, kui reeglipõhine märgendaja jätab märgenduse mitmeseks, kasutada statistilise märgendaja tulemust. Esialgsete katsete tulemusena sai proovitud ühendada eelpool mainitud viisil reeglipõhist ja TnT märgendajat, tulemuseks 94.78% õigeid märgendeid ja reeglipõhist Treetaggeriga saades tulemuseks 93.95%. Vastu ootusi selgus, et reeglipõhine märgendaja teeb 71% vigadest, ehk reeglipõhise ühestest analüüsidest on vigased 4,65%. See teadmine pani otsima uusi võimalusi: sai katsetatud süsteemi, milles kolmel märgendajal (TreeTagger, TnTTagger ja reeglipõhine) on võrdne kaal, kui reeglipõhine annab ühese märgendi, ja kui reeglipõhine ei suuda anda ühest märgendit, valida märgendiks TnTTaggeri märgend. Tulemuseks 95.44% õigeid märgendeid. Tulemuse parandamiseks sai kombineerimise osa täiendatud veelgi: kui reeglipõhine ja TnT või reeglipõhine ja TreeTagger pole võrdsed, siis valida reeglipõhise tulemus, mitte TnT tulemus, nagu eelmise variandi korral. See lähenemine aga tulemust ei parandanud, ehk tulemuseks oli 94.78% õigeid märgendeid. Vigade vaatluse järgi tuli välja, et reeglipõhine suudab õigesti määrata verbe abi- ja põhiverbideks, statistilised märgendajad aga mitte, mistõttu tekitas nn võrdne hääletamissüsteem päris palju vigu, olukorra lahendamiseks sai täiendatud kombineerimise osa nii, et verbide puhul võetakse ikkagi reeglipõhise variant, olgugi et statistilised hääletaks teise valiku poolt. Tulemus paranes, 95.71% õigeid. Viimase täiendusena sai otsustusmehhanismi lisatud täiendus, et reeglipõhise märgendaja mitteühesuse puhul vaadata, kas mõne statistilise märgendaja pakutud variant esineb

reeglipõhise variantide hulgas ja kui esineb, siis eelistada selle statistilise märgendaja märgendit. See täiendus kahjuks küll vigade arvu suurel määral ei vähendanud, tulemuseks 95.72% õigeid märgendeid. Ilmselt on põhjuseks TnTTaggeri ja TreeTaggeri vigade liigne sarnasus.

Algoritm:

Vaatleme kõiki ridu:

- kui reeglipõhine annab mitu märgendit siis
    - kui TreeTaggeri märgend esineb reeglipõhise pakutud märgendite hulgas siis kasuta TreeTaggeri märgendit
    - muidu kasuta TnTTaggeri märgendit
- teistel juhtudel
    - kui reeglipõhise ja TnTTaggeri märgend on samad või reeglipõhise ja TreeTaggeri märgend on samad siis kasuta reeglipõhise märgendit
    - teistel juhtudel
        - kui reeglipõhine pakkus märgendiks verbi, siis kasuta reeglipõhise märgendit

      teistel juhtudel kasuta TnTTaggeri märgendit

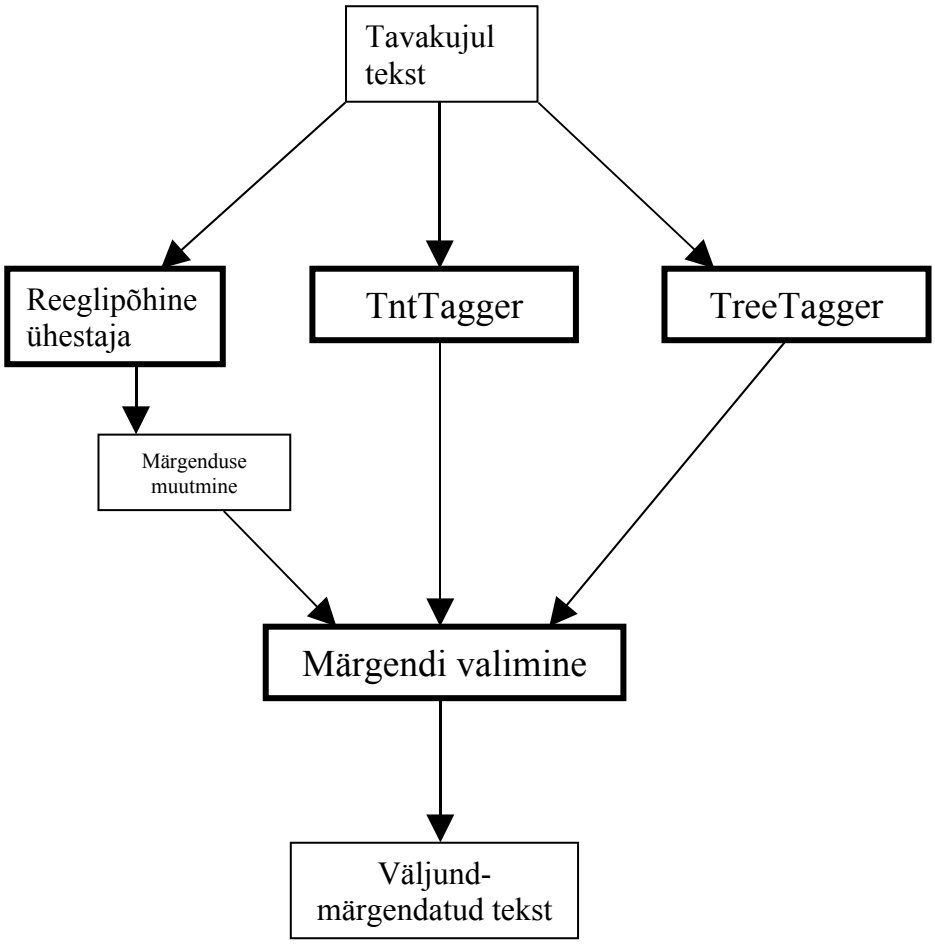
Algoritm on realiseeritud programmeerimiskeeles Perl ja programm on lisas 1 (cd-plaat).

Kordade arv	Vale	Õige
94	// S sg gen//	// S sg nom//
65	// S sg nom//	// S sg gen//
44	// P sg nom//	// P sg gen//
42	// S sg gen//	// S sg part//
41	// S sg nom//	// S sg part//
31	// S sg part//	// S sg gen//
26	// V main partic past imps//	// A pos//
25	// V main partic past ps//	// A pos//
23	// V main indic pres ps3 sg ps af//	// V main indic pres ps3 pl ps af//
23	// S sg gen//	// S sg adit//
22	// K post//	// JD //

Tabel 8.35. Märgendajate kombineerimise enamlevinud märgendite vead

Kordade arv	Vale	Õige
33 ta	//_P_sg_nom//	//_P_sg_gen//
23 on	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_main_indic_pres_ps3_pl_ps_af//
15 oli	//_V_main_indic_impf_ps3_sg_ps_af//	//_V_aux_indic_impf_ps3_sg_ps_af//
14 on	//_V_main_indic_pres_ps3_sg_ps_af//	//_V_aux_indic_pres_ps3_sg_ps_af//
13 üks	//_P_sg_nom//	//_A_sg_nom//
10 ühe	//_P_sg_gen//	//_A_sg_gen//
10 saab	//_V_mod_indic_pres_ps3_sg_ps_af//	//_V_main_indic_pres_ps3_sg_ps_af//
10 on	//_V_aux_indic_pres_ps3_sg_ps_af//	//_V_main_indic_pres_ps3_sg_ps_af//
9 tagasi	//_JD_//	//_K_post//
8 kõik	//_P_pl_nom//	//_P_sg_nom//
8 Ta	//_P_sg_nom//	//_P_sg_gen//

Tabel 8.36. Märgendajate kombineerimise enamlevinud valesti märgendatud sõnad



Joonis 8.1 Statistiliste ja reeglipõhise märgendajate ühendamine

```
"<s>"

"<Uskumatult>"
      "uskumatult" L0 D cap
"<palju>"
      "palju" L0 D
```

```
"<elu>"
       "elu" L0 S com sg gen
       "elu" L0 S com sg nom
       "elu" L0 S com sg part
"<avardavat>"
       "avarda" Lvat V main quot pres ps af <FinV> <NGP-P>
       "avarda=v" Lt A pos sg part partic <v>
"<,>"
       "," Z Com
"<tarka>"
       "tark" L0 S com sg part
"<ja>"
       "ja" L0 J crd
"<kaunist>"
       "kaunis" Lst A pos sg el
"<selgub>"
       "selgu" Lb V main indic pres ps3 sg ps af <FinV> <Intr> <El>
"<nendega>"
       "see" Ldega P dem pl kom
       "tema" Ldega P pers ps3 pl kom
"<suheldes>"
       "suhtle" Ldes V main ger <Intr>
"<.>"
       "." Z Fst
"</s>"
```

Joonis 8.2. Näide reeglipõhise ühestaja väljundist

```
Uskumatult  //_JD_//
palju //_JD_//
elu   //_S_sg_gen//
avardavat   //_A_sg_part//
,     SENT
tarka //_A_sg_part//
ja    //_JD_//
kaunist     //_A_sg_part//
selgub      //_V_main_indic_pres_ps3_sg_ps_af//
nendega     //_P_pl_kom//
suheldes    //_V_main_ger_af//
.     SENT
```

Joonis 8.3. Näide eelmisest lausest kombineeritud ühestaja väljundina, eksitud on kolmanda sõna morfoloogilise analüüsiga, õige oleks märgend //_S_sg_part//, antud juhul on vea põhjustajaks TnTTagger, kuna reeglipõhise kõhklemisel valitakse TnTTaggeri variant, samas poleks siin antud juhul aidanud ka TreeTaggeri variant, see oleks olnud //_S_sg_nom//

Järeldused ja edasiarendused

TreeTaggeri ja TnTTaggeri rakendamisel teistele keeltele on saadud samuti häid tulemusi.

Inglise keelele rakendades testiti TreeTaggerit Penn-Treebank korpusel, kasutades 2 miljonit sõna treenimiseks ja 100 000 sõna testimiseks, tulemuseks oli 96,36% õigeid märgendeid. Penn-Treebank korpus sisaldab 36 erinevat märgendit. (Schmid, 1994) TnTTaggeri rakendamisel Penn Treebank korpusel saadi 96.7% tulemus.(Brants, 2000)

Saksa keele puhul kasutati treenimiseks 20 000 sõna ja testimiseks 5000 sõna. Leksikon koosnes 350 000 sõnast ja parimaks tulemuseks saavutati 97, 53%. (Schmid, 1995) Saksakeelsete ajalehetekstidel tehtud testide alusel oli TnTTaggeri tulemuseks 97.7% (Brants,2000).

Rootsi keele puhul katsetati TreeTaggerit kasutades treenimiseks 1, 1 miljoni sõnalist korpust ja testimiseks 60 000 sõnalist osa, parimaks tulemuseks saavutati 95, 1%. Kasutati 150 erinevat märgendit. (Sjöberh, 2003) TnTTaggeri tulemuseks oli 95.31% õigeid märgendeid, kasutades 139-t märgendit (Megyesi,2001).

TreeTaggerit on kasutatud ka vana prantsuse keele märgendamisel, kus oli kasutada 2,6 miljonilist sõna treenimiseks ja testimiseks 415 000 sõnalist korpust, tulemus on ligi 95% ning kasutati 24 märgentit (Stein, 2003).

On selge, et antud töös kasutatud algoritm mitme märgendaja ühendamiseks on väga lihtsustatud ja primitiimne, edasise töö käigus olekski põhieesmärgiks algoritmi täiustamine. Töö käigus selgus, et TreeTaggeri ja TnTTaggeri koos kasutamine ei anna eeldatult häid tulemusi, kuna nende tehtavad vead kipuvad olema sarnased. Seega tuleks jätkata tööd statistilise märgendaja otsimiseks, mis käituks teisiti. Paremaid tulemusi võiks tuua ka TreeTaggeriga kombineerimisel –prob parameetri kasutamine, see annab ka märgendi tõenäosuse, mille alusel saaks otsustada, kas märgendit kasutada või mitte. Samuti tasuks uurida erinevate märgendajate tugevaid külgi ja püüda igas olukorras märgendi valimisel lähtuda antud olukorda sobivast märgendaja tulemusest.

Kokkuvõte

Tekstide automaatse analüüsi algusetapi kõige tähtsamaks osaks on morfoloogiline töötlemine. Töötlus koosneb kahest osast: sõnaliikide määramisest ja ühestamisest. Ühestamiseks on kasutusel mitmeid meetodeid: kasutatakse reeglipõhiseid, statistikal põhinevaid, närvivõrkudel baseeruvaid jt ühestajaid. Käesolevas töös antakse ülevaade katsetest rakendada eesti keelele statistikal põhinevaid ühestajaid TreeTagger ja TnTTagger ning nende kombineerimisest reeglipõhise ühestajaga.

1994. aastal töötas H. Schmid Stuttgardi ülikoolis välja otsustuspuude meetodit kasutava ühestaja – TreeTaggeri ja T.Brants 2000. aastal Stuttgardi Ülikoolis Markovi Mudelitel baseeruva programmi TnTTagger. Eesti keele jaoks on olemas ligi 500 000 sõnast koosnev morfoloogiliselt ühestatud korpus, mis on vajalik statistiliste morfoloogiliste ühestajate treenimiseks. Töö TreeTaggeri ja TnTTaggeriga koosneb kahest etapist: inimese poolt ühestatud tekstikorpusel treenimisest ja uue, morfoloogiliselt mitmese teksti automaatsest ühestamisest.

Antud töös tegeleti morfoloogianalüsaatori ja treeningtekstide teisendamisega statistiliste programmide rakendamiseks sobivale kujule, programmide treenimisega, nende optimaalse konfiguratsiooni ja märgenduse leidmisega ning saadud keelemudeli hindamisega testkorpusel.

Optimaalseimaid tulemusi võimaldanud märgendus sisaldas 248 erinevat märgendit, TreeTaggeri parimaks tulemuseks oli 91.08%, TnTTaggeril 94.78% ja kolme märgendaja kombineerimisel oli tulemuseks 95.72%.

Part-of-Speech Tagging of Estonian Language

Magister thesis

Kadri Kajaste

Abstract

This magister thesis presents two new statistical methods for Estonian part-of-speech tagging. One of the methods uses decision tree algorithm and the program is called the TreeTagger, the other uses Markov Models and is called TnTTagger.

Disambiguation is a very important part of morphological processing. More than half of the word forms are ambiguous after morphological analysis. In disambiguation phase the correct word form is chosen using the context information.

TreeTagger needs a lexicon and disambiguated text for construction of parameter file. TnTTagger creates lexicon based on training data. A lexicon is a file that contains all kind of word forms with their different potential tags. Morphologically disambiguated corpus of Estonian texts consists of aproximately 500 000 words.

The work on the project consists in preprocessing the corpus data to the demanded form, tests to find the best tagset and finding the best configuration to the taggers. Also combining TreeTagger and TnTTagger with rule-based tagger. The corpora used consists of 358 951 words, for training 319 519 word corpus was used and for evaluation, separate corpus (39 444 words). 248 different tags were used, the best result of TreeTagger was 91.08%, TnTTagger 94.78% and combining three taggers results 95.72%.

Kirjandus

1. T.Brants, *Tnt - A Statistical Part-of-Speech Tagger, User Manual*, Saarland University, 1999.
http://www.coli.uni-saarland.de/~thorsten/publications/Brants-TR-TnT.pdf

2. T. Brants, *Tnt - A Statistical Part-of-Speech Tagger,* Saarland University, 2000.
http://acl.ldc.upenn.edu/A/A00/A00-1031.pdf

3. W.Daelemans, *Machine Learning Approaches. Rmts Syntactic Wordclass Tagging.* (ed H. van Halteren). Kluwer Academic Publishers. Dordrecht/Boston/London. lk 285-304, 1999.

4. M.El-Beze , M.B. Merialdo. *1999. Hidden Markov Models. Rmts Syntactic Wordclass Tagging.* (ed H. van Halteren). Kluwer Academic Publishers. Dordrecht/Boston/London. lk 263-284, 1999.

5. EKK = Eesti Kirjakeele Korpus
http://www.cl.ut.ee/korpused/morfkorpus/ (28.05.06)

6. D.Juravsky, J.H.Martin, *Speech and Language Processing*, Second Edition, Pearson Education International, 2009

7. H.J.Kaalep *ESTMORF, eesti keele morfoloogiline analüsaator ja süntesaator, (lõplik versioon), 1999*
http://www.eki.ee/keeletehnoloogia/projektid/estmorf/estmorf.html (28.05.06)

8. H.J. Kaalep, T. Vaino *Vale meetodiga õiged tulemused? Eesti keele morfoloogiline ühestamine statistika abil*, Keel ja Kirjandus,nr 1, lk 30-38, 1998.
http://www.cl.ut.ee/yllitised/kk_yhest_1998.pdf (28.05.06)

9. K.Kajaste, *Eestikeelsete tekstide statistiline morfoloogiline ühestamine TreeTaggeriga, Bakalaureusetöö,* Tartu Ülikool, Tartu, 2006
http://math.ut.ee/~kaili/teaching/.....

10. H. Loftsson, *Tagging Icelandic text: A linguistic rule-based approach,* Nordic Journal of Linguistics, 31(1), lk 47–72, 2008

11. C. D. Manning, H.Schuetze *Foundations of Statistical Natural Language Processing*, Ch. 9. 1999.

12. B.Megyesi, *Comparing Data-Driven Learning Algorithms for POS Tagging of Swedish*, Royal Institute of Technology, Stockholm, 2001.
http://www.aclweb.org/anthology-new/W/W01/W01-0519.pdf

13. K. Müürisep *Eesti keele arvutigrammatika: Süntaks,* Dissertationes Matematicae Universitatis Tartuensis, Tartu, 2000.
http://math.ut.ee/~kaili/thesis/ (28.05.06)

14. T.Puolakainen *Eesti keele arvutigrammatika: morfoloogiline ühestamine. Dissertationes Matematicae Universitatis Tartuensis, Tartu,* 2001.

15. T.Roosmaa, M. Koit, K.Muischnek, K.Müürisep, T.Puolakainen, H.Uibo, *Eesti keele formaalne grammatika*,Tartu Ülikool, Tartu, 2001

16. H.Schmid *Probabilistic Part-of-Speech Tagging Using Decision Trees,* 1994.
http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf (28.05.06)

17. H.Schmid *Improvements In Part-of-Speech Tagging With an Application To German*, 1995.
http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger2.pdf (28.05.06)

18. J.Sjöberh *Combining POS-taggers for improved accurancy on Swedish text,* NoDaLiDa, Reykjavik, 2003.
www.nada.kth.se/~jsh/publications/combining03.pdf (28.05.06)

19. A. Stein *Part of Speech Tagging and Lemmatisation of Old French Texts,* Stuttgart, 2003.
www.uni-stuttgart.de/lingrom/stein/forschung/altfranz/afrlemma.pdf (28.05.06)

20. K.Veskis, E.Liba, *Automatic Tagger Evaluation, NLP course assignment report,* Tartu Ülikool, Tartu, 2008.

21. A.Voutilainen, Hand-Crafted Rules. Rmts Syntactic *Wordclass Tagging. (ed H. van Halteren)*. Kluwer Academic Publishers. Dordrecht/Boston/London. lk 217-246, 1999.

22. J.Wu, *Using Multible Taggers to Improve English Part-of-Speech Tagging by Error Learning,* Johns Hopkins University, Baltimore, 2006.
http://www.cs.jhu.edu/~junwu/tagreport.ps

Lisa 1. DVD: Magistritöös kasutatud korpus, märgendite hulgad, erinevate testide tulemused erinevate programmide parameetrite ja märgendihulkadega.