Keeleandmete lingvistiline töötlus EstNLTK abil

Raul Sirel

15. september 2015

Sissejuhatavat

- Keeleandmete efektiivseks analüüsiks on tarvis neid eelnevalt töödelda

- Eeltöötluse võib üldjoontes jagada kaheks:
  - tehniline eeltöötlus (sh andmete parsimine)
  - lingvistiline eeltöötlus

- Antud aine raames kasutame lingvistiliseks eeltöötluseks EstNLTK projekti raames loodud ressursse

Lingvistiline eeltöötlus

- Lingvistilise eeltöötluse hulka kuulub:
  - teksti segmenteerimine (sõnedeks ja (osa)lauseteks)
  - lemmatiseerimine
  - sõnaliikide määramine / morfoloogiline analüüs
  - sünonüümia lahendamine (nt Wordneti abil)
  - ...
- Eeltöötluse täpne töövoog sõltub lahendatavast ülesandest või tehtavast analüüsist

EstNLTK ja ***Text*** klass

- Põhiline klass, millega teksti töödelda, on ***Text***.
- Klassil on hulk meetodeid, mille abil teksti analüüsida
- Põhimõtteliselt on tegemist *Python*i sõnaraamatuga (*dict*):

```
>>> from estnltk import Text
>>> text = Text('Tere maailm!')
>>> text
{u'text': u'Tere maailm!'}
```

- Sõnaraamatu põhielementideks on **märgenduskihid**
- Nt toorteksti kuvamiseks tuleb pöörduda sõnaraamatu poole klassimuutujaga “text”:

```
>>> text.text
u'Tere maailm!'
```

EstNLTK märgenduskihtidest

- Märgenduskiht kannab informatsiooni tekstis leiduvate olemite kohta:
  - olemi tüüp (nt sõna, lause, ajaväljend, kohanimi jne)
  - olemi positsioon (ehk asukoht algses tekstis)

- Märgenduskihtide arv ei ole piiratud – põhimõtteliselt võib märgendada kõike, mis pähe tuleb

- Märgenduskihid on antud ***Text*** klassis võti-väärtus paaridega, kus väärtuseks on märgendatud olemite järjend

EstNLTK märgenduskihtidest (2)

- Märgenduskihtide sõnaraamat on jooksvalt uuendatav
- Seega saab märgenduskihte alati lisada või neid kustutada:

```
>>> text = Text('Tere maailm!')
>>> text
{u'text': u'Tere maailm!'}

>>> text.text
u'Tere maailm!‘

>>> text.sentences
[{u'start': 0, u'end': 12}]

>>> text
{u'text': u'Tere maailm!', u'sentences': [{u'start': 0, u'end': 12}], u'paragraphs':
[{u'start': 0, u'end': 12}]}

>>> del text['sentences']
>>> text
{u'text': u'Tere maailm!', u'paragraphs': [{u'start': 0, u'end': 12}]}
```

EstNLTK märgenduskihtidest (3)

- Märgenduskihid on itereeritavad – nt saab itereerida üle tekstist tuvastatud lausete, sõnade, ajaväljendite jne:

```
>>> text = Text('Tere maailm!')
>>> text.words
[{u'start': 0, u'end': 4, u'text': u'Tere'}, {u'start': 5, u'end': 11, u'text':
u'maailm'}, {u'start': 11, u'end': 12, u'text': u'!'}]

>>> for word in text.words:
          print word

{u'start': 0, u'end': 4, u'text': u'Tere'}
{u'start': 5, u'end': 11, u'text': u'maailm'}
{u'start': 11, u'end': 12, u'text': u'!'}
```

Teksti segmenteerimine

Lausestamine

- Lausestamine ehk lausepiiride tuvastamine
- Oluline, sest teksti on mõistlik analüüsida ühe lause piires
- Näiteks informatsiooni ekstraheerimisel:
  - *President Meri keskendus oma kõnes päevapoliitilistele sündmustele.*
  - *Eestit külastas USA president. Meri on tänavu jäävaba.*

- Iga punkt ei ole lausepiir (nt järgarvud), mistõttu tuleb taas analüüsida lokaalset konteksti

Lausestamine EstNLTKs

```
>>> from pprint import pprint
>>> text = Text('Tere maailm!')
>>> text.sentences
...
>>> pprint(text)
{u'paragraphs': [{u'end': 12, u'start': 0}],
 u'sentences': [{u'end': 12, u'start': 0}],
 u'text': u'Tere maailm!'}
```

Üksustamine

- Ingl k *tokenisation*
- A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.[1]
- Üksus on enamasti:
  - sõna (kollane, Jeltsin)
  - arv (42, 3,14, 666., ¼)
  - kirjavahemärk (– , :)
  - erisümbolid (€, ¥)

[1] http://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html

Üksustamine

- Praktiline probleem – kirjavahemärgid on “kleepunud” sõnade külge

- Sisend:

  Meremees Jack ja veel üheksa meest, kes tema sõnade järgi olid ettevõtlikud sellid, vehkisid sisse korraliku paadi ning asusid mööda jõge allapoole teele.

- Väljund:

  Meremees Jack ja veel üheksa meest , kes tema sõnade järgi olid ettevõtlikud sellid , vehkisid sisse korraliku paadi ning asusid mööda jõge allapoole teele .

Üksustamine

- Iga punkt ja koma ei pruugi olla kirjavahemärk:
  - järgarvud
  - kümnendmurrud
  - …
- Seega tuleb analüüsida lokaalset konteksti – millised sõnad ja muud märgid on konkreetse juhtumi ümbruses
- EstNLTK raames rakendatakse selleks regulaaravaldisi

Üksustamine EstNLTKs

```
>>> from pprint import pprint
>>> text = Text('Tere maailm!')
>>> text.words
{u'text': u'Tere maailm!', u'sentences': [{u'start': 0, u'end': 12}], u'paragraphs':
[{u'start': 0, u'end': 12}]}

>>> pprint(text)
{u'paragraphs': [{u'end': 12, u'start': 0}],
 u'sentences': [{u'end': 12, u'start': 0}],
 u'text': u'Tere maailm!',
 u'words': [{u'end': 4, u'start': 0, u'text': u'Tere'},
            {u'end': 11, u'start': 5, u'text': u'maailm'},
            {u'end': 12, u'start': 11, u'text': u'!'}]}
```

Osalausestamine

- Osalausestamise käigus tuvastatakse liitlauses osalausete piirid:
  - Kell sai kaks ja ma hakkasin minema.
  - *Linn, mille poole me liikusime, oli parajasti leekides.*

- Osalausete vahel eristatakse kaht tüüpi seoseid:
  - rinnastusseos
  - alistusseos

- Oluliselt keerulisem probleem kui lausestamine, kuna lausestruktuuri varieeruvus on suur

Osalausestamine

```
>>> text = Text("Kell sai kaks ja ma hakkasin minema.")
>>> text.clause_texts
[u'Kell sai kaks ja', u'ma hakkasin minema.']

>>> text.clauses
[{u'start': [0], u'end': [16]},
 {u'start': [17], u'end': [36]}]




>>> text = Text("Linn, mille poole me liikusime, oli parajasti leekides.")
>>> text.clause_texts
[u'Linn oli parajasti leekides.',
 u', mille poole me liikusime,']

>>> text.clauses
[{u'start': [0, 32], u'end': [4, 55]},
 {u'start': [4], u'end': [31]}]
```

Morfoloogiline informatsioon

Meeldetuletus arvutimorfoloogiast

- Eesti keele muuteparadigmad on mahukad – leidub väga palju käändelisi ja pöördelisi vorme

- Samuti palju mitmetähenduslikkust, nt:
  - *kallas*
    - ainsuse nimetav substantiivist *kallas*
    - lihtmineviku 3. isiku ainsuse vorm verbist *kallama*
  - *kastis*
    - ainsuse seesütlev substantiivist *kast*
    - lihtmineviku 3. isiku ainsuse vorm verbist *kastma*
  - jne

Meeldetuletus arvutimorfoloogiast

- Eesti keele morfoloogiline analüüs:
  - ainult konkreetset sõnavormi vaadeldes võib sellel olla mitu analüüsi
  - kasutatakse reeglipõhist lähenemist, mille käigus määratakse kõik võimalikud analüüsid

- Eesti keele morfoloogiline ühestamine:
  - eesmärk on käia üle morfoloogilise analüsaatori väljund ning valida vorm, mis on antud kontekstis korrektne
  - kasutatakse statistilist keelemudelit (Markovi peitmudel)
  - mõnedel juhtudel jäävad mitmesused siiski alles

Lemmatiseerimine

- Lemmatiseerimine ehk sõnade algvormide määramine
- Eesti keele muuteparadigmad on mahukad – leidub palju käändelisi ja pöördelisi vorme
- Tekstist informatsiooni tuvastamiseks on mõistlik sõnad nende algvormideks teisendada
- Tulemuseks oluliselt väiksem leksikon (sõnavara tähenduses)

- Näiteks eestikeelse vikipeedia artikkel Eestist (https://et.wikipedia.org/wiki/Eesti):
  - lemmatiseerimata kujul 3560 unikaalset sõne
  - lemmatiseeritud kujul 2647 unikaalset sõne (seega ~25% väiksem leksikon)

Lemmatiseerimine EstNLTK's

- Lemmade leidmine klassimuutujaga "lemmas":

```
>>> text = Text('Mis kell sa tulema hakkad?')
>>> text.lemmas
[u'mis', u'kell', u'sina', u'tulema', u'hakkama', u'?']
```

- Lemmad pärinevad morfoloogilise analüsaatori ja ühestaja väljundist

- Informatsioon lisatakse teksti kirjeldavasse sõnaraamatusse

Sõnaliikide määramine EstNLTKs

- Sõnaliikide määramine klassimuutujaga “postags”:

```
>>> text = Text('Ibumetin on levinud valuvaigisti.')
>>> text.postags
[u'H', u'V', u'A|V', u'S', u'Z']
```

- Väljastatakse järjend sõnaliikide märgenditega:
  - ühestamata juhtudel on märgendid eraldatud püstkriipsuga
  - informatsioon lisatakse teksti kirjeldavasse sõnaraamatusse

POS tag	Description	Example
A	omadussõna - algvõrre (adjektiiv - positiiv), nii käänduvad kui käändumatud	kallis või eht
C	omadussõna - keskvõrre (adjektiiv - komparatiiv)	laiem
D	määrsõna (adverb)	kõrvuti
G	genitiivatribuut (käändumatu omadussõna)	balti
H	pärisnimi	Edgar
I	hüüdsõna (interjektsioon)	tere
J	sidesõna (konjunktsioon)	ja
K	kaassõna (pre/postpositsioon)	kaudu
N	põhiarvsõna (kardinaalnumeraal)	kaks
O	järgarvsõna (ordinaalnumeraal)	teine
P	asesõna (pronoomen)	see
S	nimisõna (substantiiv)	asi
U	omadussõna - ülivõrre (adjektiiv - superlatiiv)	pikim
V	tegusõna (verb)	lugema
X	verbi juurde kuuluv sõna, millel eraldi sõnaliigi tähistus puudub	plehku
Y	lühend	USA
Z	lausemärk	-, /, ...

http://estnltk.github.io/estnltk/1.2/tutorials/morf_tables.html#table-pos-tag-descriptions

Morfoloogiline informatsioon

- Lisaks lemmadele ja sõnaliikidele saab leida ka muud informatsiooni sõnade koostise kohta

- Informatsioon on esitatud märgenditena

- Kasutatavad märgendid:
  - http://estnltk.github.io/estnltk/1.2/tutorials/morf_tables.html#table-noun-form-descriptions
  - http://estnltk.github.io/estnltk/1.2/tutorials/morf_tables.html#table-verb-form-descriptions

```
>>> text = Text('Tere maailm!').analysis
>>> pprint(text)
{u'paragraphs': [{u'end': 12, u'start': 0}],
 u'sentences': [{u'end': 12, u'start': 0}],
 u'text': u'Tere maailm!',
 u'words': [{u'analysis': [{u'clitic': u'',
                            u'ending': u'0',
                            u'form': u'',
                            u'lemma': u'tere',
                            u'partofspeech': u'I',
                            u'root': u'tere',
                            u'root_tokens': [u'tere']}],
             u'end': 4,
             u'start': 0,
             u'text': u'Tere'},
            {u'analysis': [{u'clitic': u'',
                            u'ending': u'0',
                            u'form': u'sg n',
                            u'lemma': u'maailm',
                            u'partofspeech': u'S',
                            u'root': u'maa_ilm',
                            u'root_tokens': [u'maa', u'ilm']}],
             u'end': 11,
             u'start': 5,
             u'text': u'maailm'},
            {u'analysis': [{u'clitic': u'',
                            u'ending': u'',
                            u'form': u'',
                            u'lemma': u'!',
                            u'partofspeech': u'Z',
                            u'root': u'!',
                            u'root_tokens': [u'!']}],
             u'end': 12,
             u'start': 11,
             u'text': u'!'}]}
```

EstNLTK ilma morfoloogilise ühestajata

- EstNLTK võimaldab statistilist ühestamist välja lülitada

- Selleks tuleb *Text* objekti loomisel varustada see *disambiguate=False* argumendiga

```
>>> text = Text("Reis kuule.")
>>> pprint(text.analysis)
[[{u'clitic': u'',
   u'ending': u'0',
   u'form': u'sg n',
   u'lemma': u'reis',
   u'partofspeech': u'S',
   u'root': u'reis',
   u'root_tokens': [u'reis']}],
 [{u'clitic': u'',
   u'ending': u'le',
   u'form': u'sg all',
   u'lemma': u'kuu',
   u'partofspeech': u'S',
   u'root': u'kuu',
   u'root_tokens': [u'kuu']}],
 [{u'clitic': u'',
   u'ending': u'',
   u'form': u'',
   u'lemma': u'.',
   u'partofspeech': u'Z',
   u'root': u'.',
   u'root_tokens': [u'.']}]]
```

```
>>> text = Text("Reis kuule.",disambiguate=False)
>>> pprint(text.analysis)
[[{u'clitic': u'',
   u'ending': u'0',
   u'form': u'sg n',
   u'lemma': u'Reis',
   u'partofspeech': u'H',
   u'root': u'Reis',
   u'root_tokens': [u'Reis']},
  {u'clitic': u'',
   u'ending': u's',
   u'form': u'sg in',
   u'lemma': u'Rei',
   u'partofspeech': u'H',
   u'root': u'Rei',
   u'root_tokens': [u'Rei']},
  {u'clitic': u'',
   u'ending': u'0',
   u'form': u'sg n',
   u'lemma': u'Reis',
   u'partofspeech': u'H',
   u'root': u'Reis',
   u'root_tokens': [u'Reis']},
  {u'clitic': u'',
   u'ending': u'0',
   u'form': u'sg n',
   u'lemma': u'reis',
   u'partofspeech': u'S',
   u'root': u'reis',
   u'root_tokens': [u'reis']}],
 [{u'clitic': u'',
   u'ending': u'le',
   u'form': u'sg all',
   u'lemma': u'kuu',
   u'partofspeech': u'S',
   u'root': u'kuu',
   u'root_tokens': [u'kuu']},
  {u'clitic': u'',
   u'ending': u'0',
   u'form': u'o',
   u'lemma': u'kuulma',
```

Informatsiooni ekstraheerimine

Ajaväljendite tuvastamine

- Eesmärk on tuvastada tekstist fragmendid, mis viitavad ajale:
  - aastaarvud
  - kuupäevad
  - nädalapäevad
  - jne

- Ajaväljendite tuvastamiseks EstNLTKs on klassil *Text* meetod *timexes*

Ajaväljendite tuvastamine EstNLTKs

```
>>> text = Text('Konspiraatorid kogunesid reedel Narva kohvikus, et arutada
komandanditunni kehtestamist.')
>>> pprint(text.timexes)
[{u'end': 31,
  u'id': 0,
  u'start': 25,
  u'temporal_function': True,
  u'text': u'reedel',
  u'tid': u't1',
  u'type': u'DATE',
  u'value': u'2015-09-18'}]
```

Nimega üksuste tuvastamine

- Nimega üksuste tuvastamise (ingl k *named-entity recognition*) eesmärk on tuvastada ja kategoriseerida tekstifragmente, mis viitavad:
  - isikutele
  - kohtadele
  - organisatsioonidele
  - jne

Nimega üksuste tuvastamine EstNLTK’s

```
>>> text = Text("Osvald soovib sõita Tallinnast Tartusse. Tallinna ja Tartu vahel
opereerivad lisaks AS Sebele ka AS Taisto Reiside ja teiste ettevõtete bussid.")

>>> text.named_entities
[u'Osvald', u'Tallinn', u'Tartu', u'Tallinn', u'Tartu', u'AS Sebele', u'AS Taisto
Reiside']

>>> pprint(text['named_entities'])
[{u'end': 6, u'label': 'PER', u'start': 0},
 {u'end': 30, u'label': 'LOC', u'start': 20},
 {u'end': 39, u'label': 'LOC', u'start': 31},
 {u'end': 49, u'label': 'LOC', u'start': 41},
 {u'end': 58, u'label': 'LOC', u'start': 53},
 {u'end': 93, u'label': 'ORG', u'start': 84},
 {u'end': 114, u'label': 'ORG', u'start': 97}]
```

Semantilised suhted

WordNet

- Wordnet on teatud tüüpi leksikaal-semantiline andmebaas, mis sisaldab informatsiooni mõistete ja nendevaheliste semantiliste suhete kohta

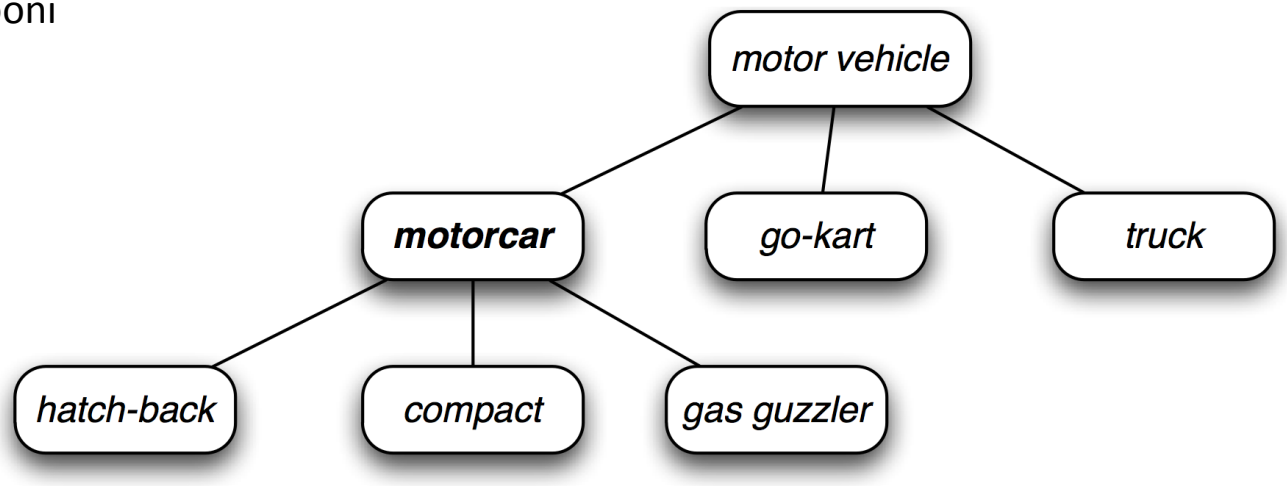
- Wordneti tüüpi tesauruse põhiühik on sünonüümihulk (ka *sünohulk*)
- Sünohulk koosneb kõigist ühte ja sama mõistet väljendavatest sõnadest (või sõnaühenditest)

- Sünohulgad on on ühendatud viidetega, mis vastavad mõistetevahelistele semantilistele või leksikaalsetele suhetele
- Kõige olulisemad suhted on hüponüümia ja hüperonüümia, kuid sageli on märgitud ka meronüümia, antonüümia jne

Wordneti struktuur

- Wordnet on struktuurilt graaf, kus tippudeks on sünuhulgad ja kaarteks nendevahelised relatsioonid

- Eesti WN sisaldab umbes:
  - 65 000 sünohulka (u 100 000 sõna)
  - 200 000 relatsiooni



EstNLTK ja WordNet

- Otsing Wordnetist, tulemus esitatakse järjendina:

```
>>> from estnltk.wordnet import wn
>>> wn.synsets("kohvik")
[u"Synset('kohvik.n.01')"]
```

- Otsingut saab kitsendada sõnaliigi alusel:

```
>>> wn.synsets("selg")
[u"Synset('selg.n.02')",
u"Synset('selg.n.01')"]

>>> wn.synsets("selg",pos=wn.VERB)
[]
```

- Vastavad tähendused on nummerdatud:

```
>>> synsets = wn.synsets("selg")
>>> synsets
[u"Synset('selg.n.02')", u"Synset('selg.n.01')"]

>>> for synset in synsets:
          print synset.definition()

kuju, asendi v. ülesande poolest selga meenutav
osa esemel, kehaosal v. loodusobjektil
inimese v. looma keha tagumine külg kaelast kuni
selgroo lõpuni
```

- Sünohulga sõnaliik:

```
>>> synsets = wn.synsets("selg")
>>> synsets[0].pos
u'n'
```

EstNLTK ja WordNet

- Hüpero- ja hüponüümide (ülem- ja alammõistete) leidmine:

```
>>> wn.synsets("kohvik")
[u"Synset('kohvik.n.01')"]

>>> wn.synsets("kohvik")[0].hypernyms()
[u"Synset('toitlustusettev\xf5te.n.01')"]

>>> wn.synsets("kohvik")[0].hyponyms()
[u"Synset('suvekohvik.n.01')",
u"Synset('vaba\xf5hukohvik.n.01')",
u"Synset('internetikohvik.n.01')",
u"Synset('esinduskohvik.n.01')",
u"Synset('noortekohvik.n.01')",
u"Synset('rannakohvik.n.01')",
u"Synset('tantsukohvik.n.01')",
u"Synset('\xfcli\xf5pilaskohvik.n.01')",
u"Synset('j\xe4\xe4tisekohvik.n.01')",
u"Synset('lastekohvik.n.01')",
u"Synset('keldrikohvik.n.01')"]
```

EstNLTK ja WordNet

- Ühise hüperonüümi (ülemmõiste) leidmine:

```
>>> wn.synsets("kohvik")
[u"Synset('kohvik.n.01')"]

>>> wn.synsets("baar")
[u"Synset('baar.n.01')", u"Synset('\xf5llebaar.n.01')"]

>>> wn.synsets("kohvik")[0].lowest_common_hypernyms(wn.synsets("baar")[0])
[u"Synset('toitlustusettev\xf5te.n.01')"]

>>> wn.synset("kohvik.n.01").lowest_common_hypernyms(wn.synset("baar.n.01"))
[u"Synset('toitlustusettev\xf5te.n.01')"]
```

Lõppu midagi kasulikku, mida polnud kuhugi mujale panna

Analüüsitud teksti itereerimine

```
>>> text = Text("See on esimene lause. See on aga teine lause.")
>>> for sentence in text.divide():
          print "LAUSE ALGUS"
          for word in sentence:
                    print word

LAUSE ALGUS
{u'start': 0, u'end': 3, u'text': u'See'}
{u'start': 4, u'end': 6, u'text': u'on'}
{u'start': 7, u'end': 14, u'text': u'esimene'}
{u'start': 15, u'end': 20, u'text': u'lause'}
{u'start': 20, u'end': 21, u'text': u'.'}
LAUSE ALGUS
{u'start': 22, u'end': 25, u'text': u'See'}
{u'start': 26, u'end': 28, u'text': u'on'}
{u'start': 29, u'end': 32, u'text': u'aga'}
{u'start': 33, u'end': 38, u'text': u'teine'}
{u'start': 39, u'end': 44, u'text': u'lause'}
{u'start': 44, u'end': 45, u'text': u'.'}
```