EstNLTK libraries for NLP

Sven Laur[♠,♢]

https://github.com/estnltk

[♠]STACC
[♢]University of Tartu

Yet another library for NLP?

There are myriad of libraries for processing text

- GATE, Stanford CoreNLP (Java, general purpose)
- NLTK, SPACY (Python, general purpose)
- Scikit-learn, Gensim (Python, specific tasks)

There are many tools for analysing Estonian language

- Vabamorf, EstHST, EstNeuroMorph (morphology)
- EstCG, EstMalt (syntax)
- Estonian WordNet
- Named Entity Recognition
- . . .

Goals of EstNLTK project

EstNLTK is a Python library distributed under GPLv2 license:

▷ Easy to install, learn and use

▷ Unified framework for text annotations

▷ Programmatic access to existing text analysis tools

▷ Predefined but reconfigurable workflows for common tasks

▷ Unified framework for visualisation and storing annotations

How can I use EstNLTK?

Installation

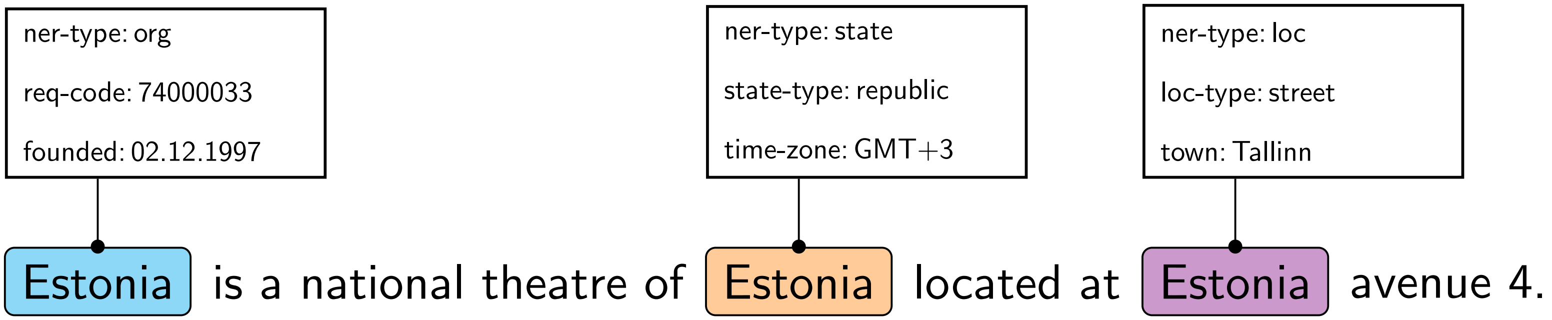
- ▷ Anaconda binary packages (easy)
- ▷ Standard pip installer (I like obscure C++ compiling errors)
- ▷ The latest GIT commit (Random hacks to avoid C++ compiler)

Licensing

- ▷ I believe in free software (GPLv2)
- ▷ I am building a service or software for internal usage (GPLv2)
- ▷ I need more liberal licence to sell a commercial product:
  - ◇ Contact University of Tartu for different license
  - ◇ Write a wrapper to make EstNLTK as a separate process

Basic concepts

- Annotations for text spans (Taggers)



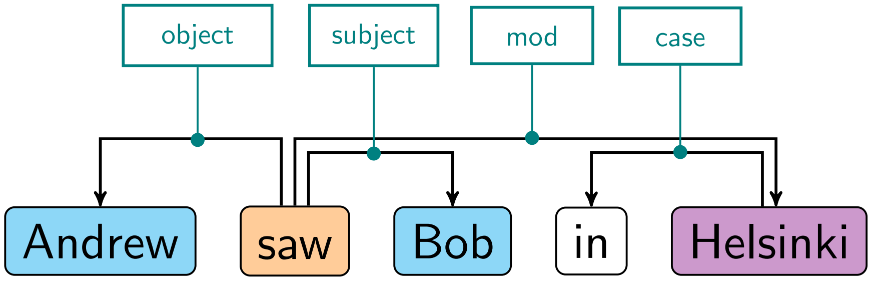
ner-type: org
req-code: 74000033
founded: 02.12.1997

ner-type: state
state-type: republic
time-zone: GMT+3

ner-type: loc
loc-type: street
town: Tallinn

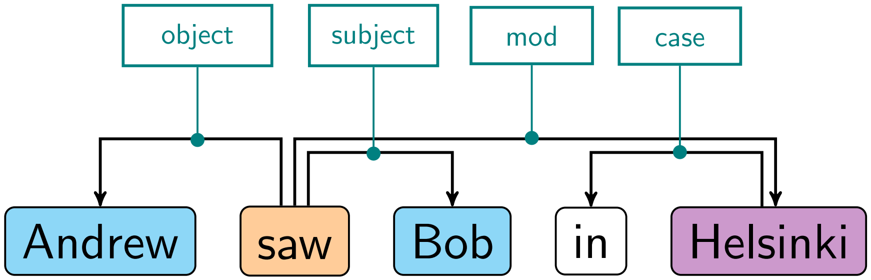
Estonia is a national theatre of Estonia located at Estonia avenue 4.

- Annotations for relations between spans (Syntax & Complex facts)

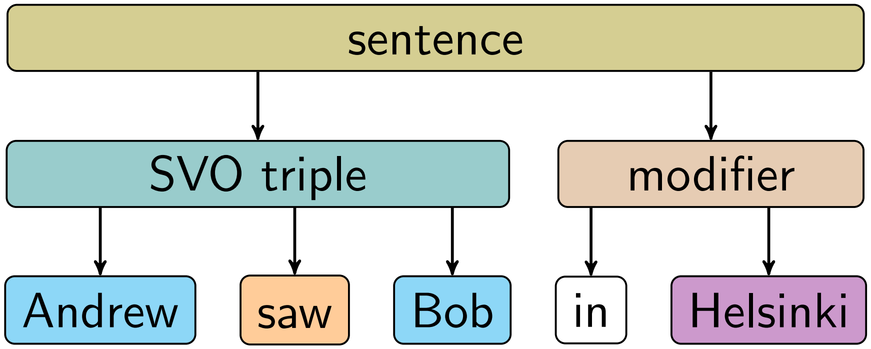


Basic concepts

- ▷ Annotations for relations between spans (Syntax & Complex facts)



- ▷ Span trees on top of text spans (Phrases & Fact extraction)



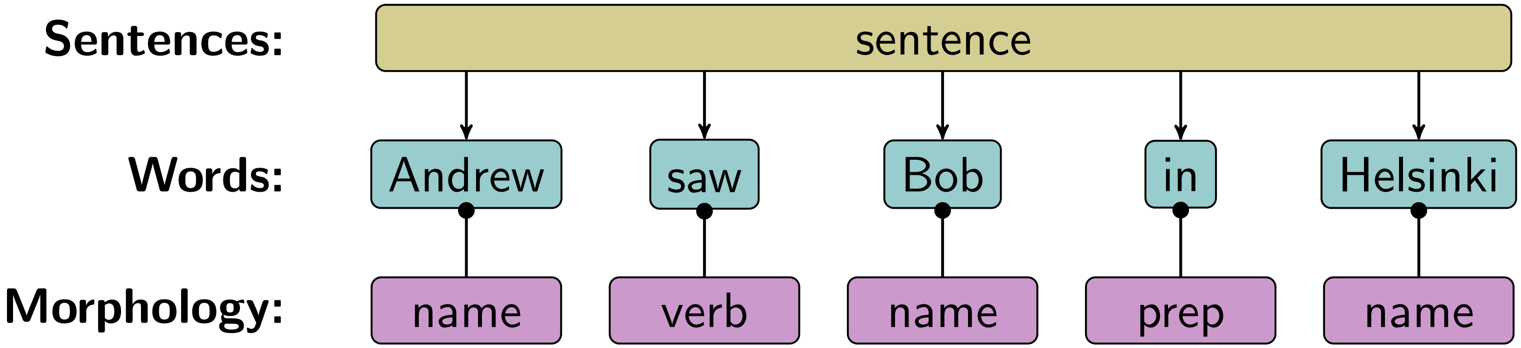
What can you tag with EstNLTK?

- Basic building blocks of text:
  - words, sentences, paragraphs.
- Morphology
  - lemma, part of speech, case, tense, number,. . .
  - in **Estmorf**, Giellatekno, Visl CG formats,
- Important phrases (**EstNLTK 1.4**)
  - named entities
  - noun phrases, adjective phrases, verb chains
  - clauses, time phrases
- Syntax (EstNLTK 1.4)
  - EstCG and EstMalt models

What does EstNLTK 1.6 offer?

- Span hierarchies
- Ambiguous annotations
- New analysis algorithms
- Predefined analysis workflows
- Two-phase fact extraction algorithms
- Postgre collections for storage and search
- Better visualisation and integration with jupyter
- Standard taggers for important phrases
- Native support for syntax analysis

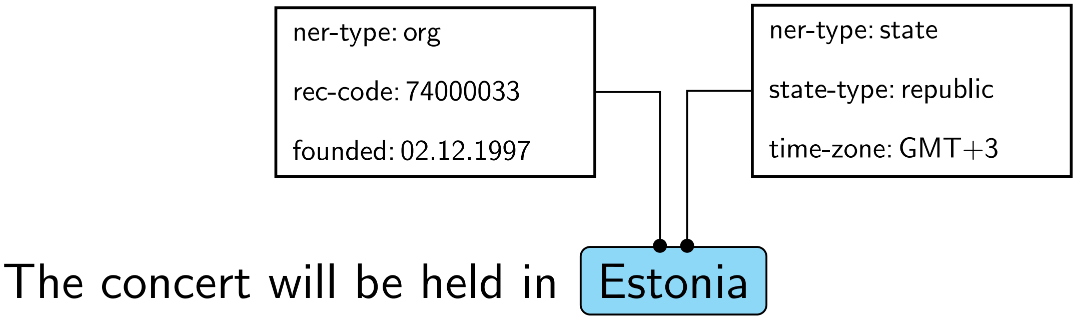
Span hierarchies in EstNLTK 1.6



It is natural to define new spans in terms of other spans:

- ◇ new phrases
- ◇ independent annotations

Ambiguous annotations



ner-type: org

rec-code: 74000033

founded: 02.12.1997

ner-type: state

state-type: republic

time-zone: GMT+3

The concert will be held in Estonia

Sometimes it is impossible to assign unique interpretation to a span

- ◇ A span can have several annotations
- ◇ Annotations of different spans are independent

Robust NLP pipleline

Why?

- Good analysis requires many tedious cleaning steps
- You can incorporate data specific tweaks into the pipeline

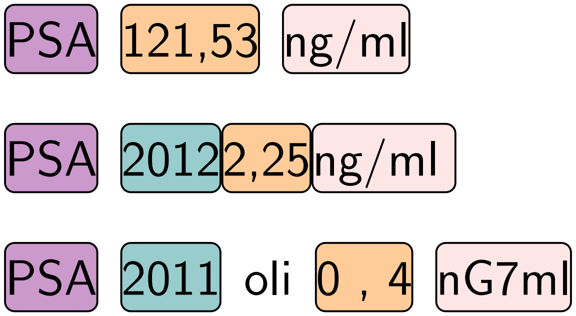
What does the robust NLP pipeline do?

- Identifies compound tokens:
  - numbers, dates, units, urls, emails, xml-tags
  - abbreviations, emoticons, symbol tokens, compound names,. . .
- Identifies normal forms for words:
  - date normalisation
  - corrects spelling mistakes
- . . .

What does the robust NLP pipeline do?

- Identifies compound tokens:
  - numbers, dates, units, urls, emails, xml-tags
  - abbreviations, emoticons, symbol tokens, compound names,. . .
- Identifies normal forms for words:
  - date normalisation
  - corrects spelling mistakes
- Identifies sentence and paragraph borders
  - standard sentence detection
  - post-corrections (numbers, abbreviations, emoticons)
- Performs morphological analysis
  - vabamorf
  - post-corrections for specific words (number-text combos)
- Performs syntax analysis

Two-phase fact extraction



PSA 121,53 ng/ml

PSA 20122,25ng/ml

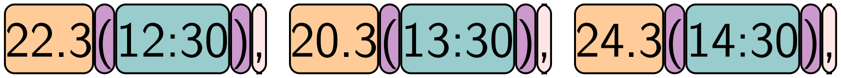
PSA 2011 oli 0 , 4 nG7ml

Fact extraction can be done with finite attribute grammars

- ▷ Tokenisation is often ambiguous
- ▷ Grammar rules filter out spurious token variants
- ▷ Meaning can be given iteratively from bottom up

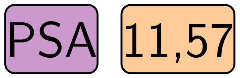
Tweaks to the previous ideas

- It is convenient to allow infinite sequences

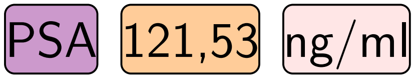


- It is convenient to add rule priorities

measurement → object number

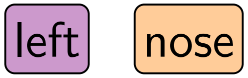


measurement → object number unit

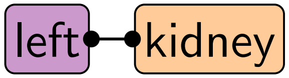


- It is convenient to cancel productions with conflicting attributes

There is only one nose:



There are two kidneys:



Postgre storage

Why?

- ▷ Provides a simple searchable JSON serialisation to EstNLTK objects
- ▷ Allows to speeds up fact extraction after tokenisation is done

What does Postgre collection do?

- ▷ Allows to define new layers
- ▷ Allows to index layers with fingerprints
- ▷ Allows to compare different layers during grammar development

What does Postgre collection contain?

- ▷ Text objects, outer layers, layer fragments
- ▷ Meta attributes for text objects and layers

Contributors to EstNLTK project

Developers

- Siim Orasmaa
- Timo Petmanson
- Uku Raudvere
- Dage Särg
- Paul Tammo
- Aleksandr Tkatšenko

Consulting

- Heiki-Jaan Kaalep
- Kadri Muischnek
- Kairit Sirts
- Tarmo Vaino