

TectoMT: Modular NLP Framework

Martin Popel, Zdeněk Žabokrtský
ÚFAL, Charles University in Prague



IceTAL, 7th International Conference on
Natural Language Processing
August 17, 2010, Reykjavik

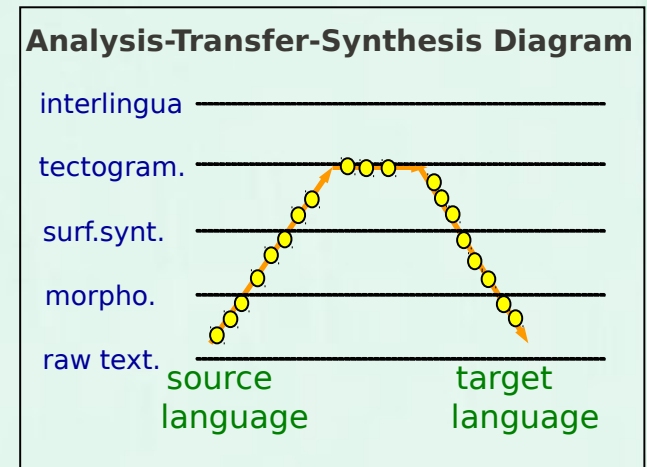
Outline

- Motivation
- Layers of language description in PDT
- TectoMT architecture
- TectoMT internals
- Future plans
- Conclusion

Motivation for creating TectoMT

Originally a framework for a linguistically motivated MT system

- deep syntactic (tectogrammatical) transfer
- started with English to Czech direction
- translation process divided to ~ 90 “blocks”
- combining statistical and rule-based blocks



Goals:

- elegant integration of in-house and third-party NLP tools
- modularity, reusability, cooperation
- ability to easily modify and add code in full-fledged programming language (Perl)

Motivation for creating TectoMT



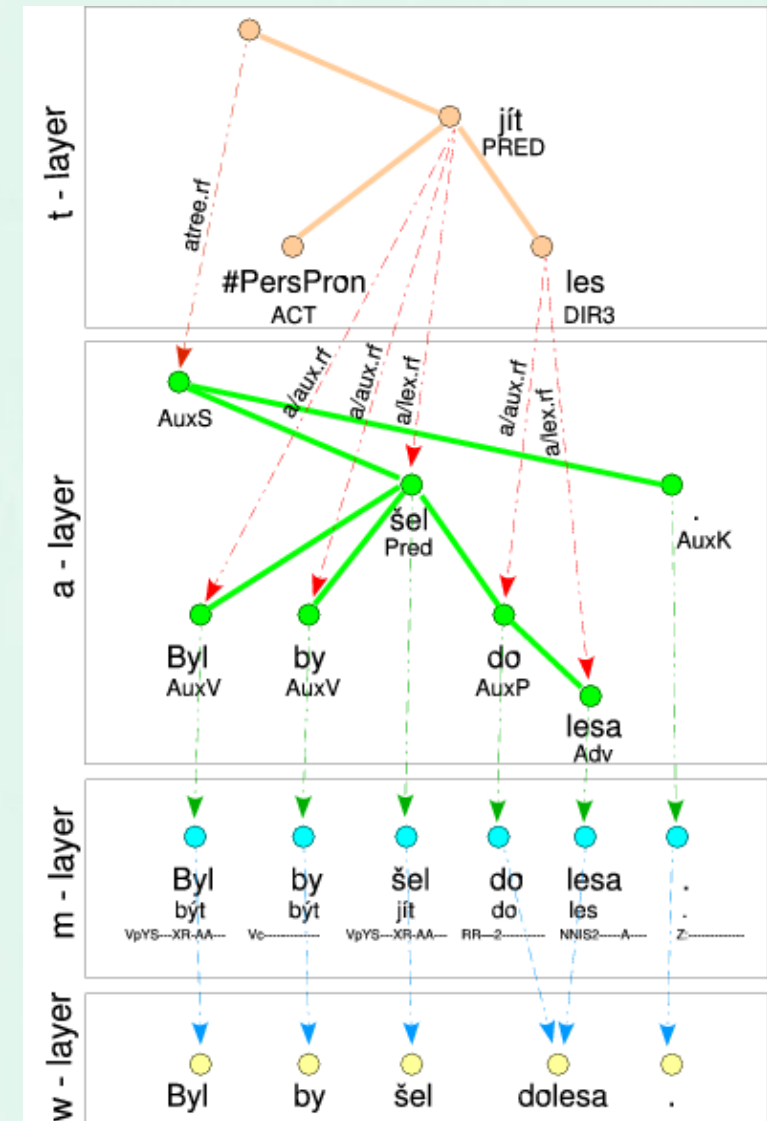
Now used for many other projects,
not limited to MT nor tectogrammatics:

- automatic alignment & annotation of a parallel treebank (CzEng)
- support for manual annotations (PEDT)
- lemmatization, tagging, parsing
- named entity recognition, information retrieval, coreference
- preprocessing for phrase-base MT
 - change word order, append determiners to nouns,...
 - add deep-syntactic features as an input for factored translation
- conversions, evaluations, etc.

4 layers of language description

implemented in Prague Dependency Treebank (PDT)

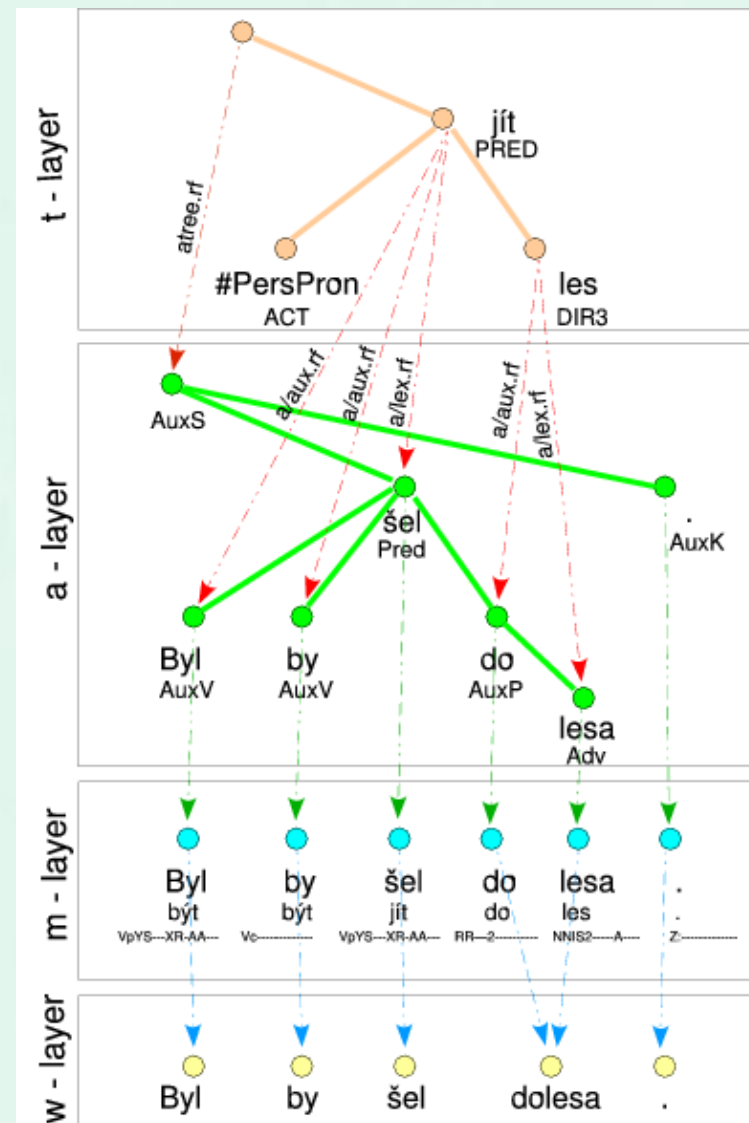
- word layer
raw (tokenized) text



4 layers of language description

implemented in Prague Dependency Treebank (PDT)

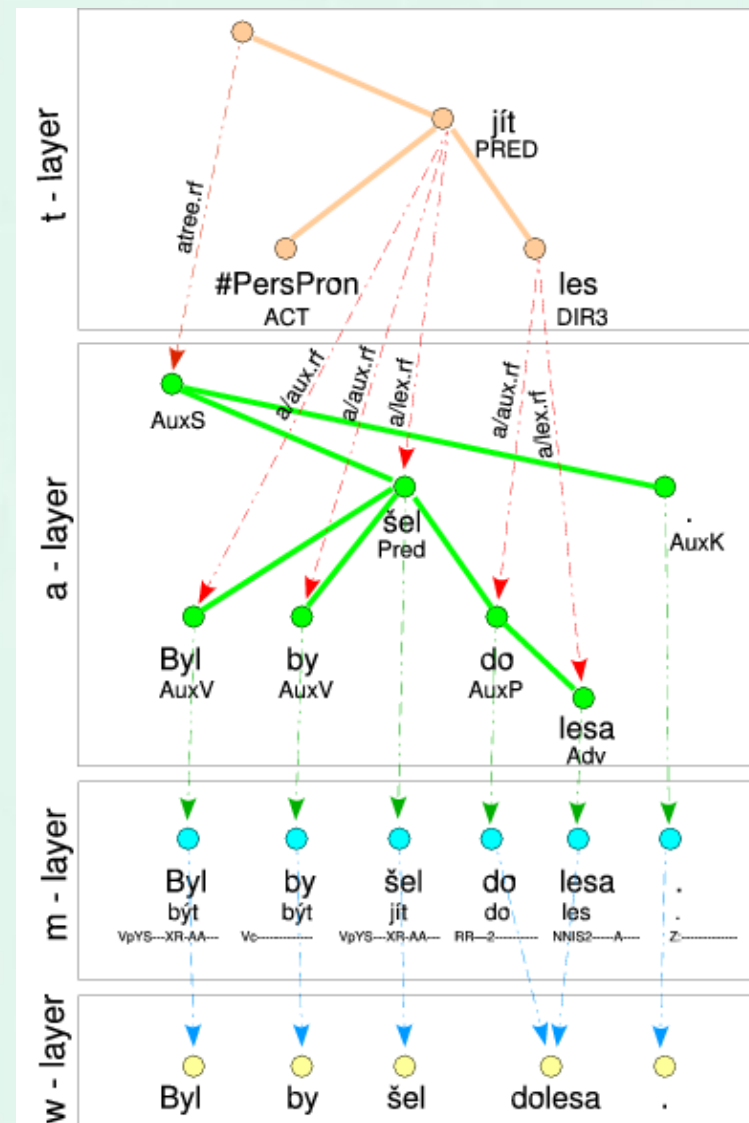
- **morphological layer**
lemma & POS tag for each word
- **word layer**
raw (tokenized) text



4 layers of language description

implemented in Prague Dependency Treebank (PDT)

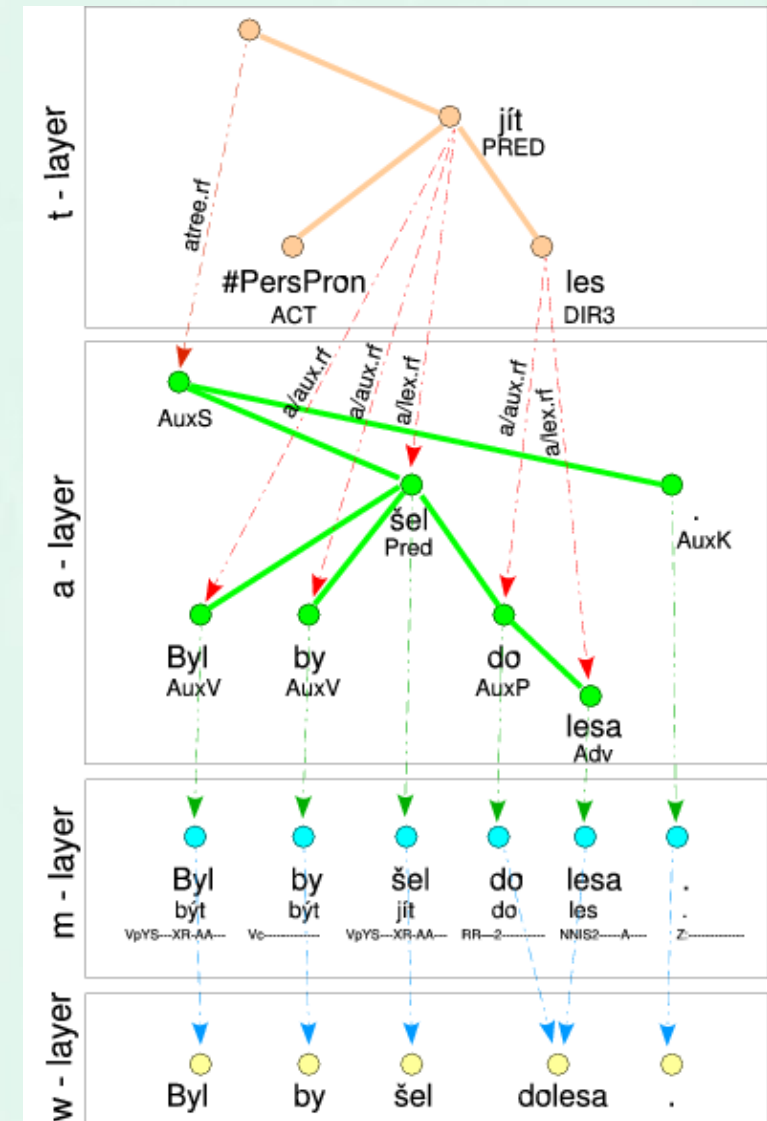
- **analytical layer**
surface-syntactic dependency trees, labeled edges
- **morphological layer**
lemma & POS tag for each word
- **word layer**
raw (tokenized) text



4 layers of language description

implemented in Prague Dependency Treebank (PDT)

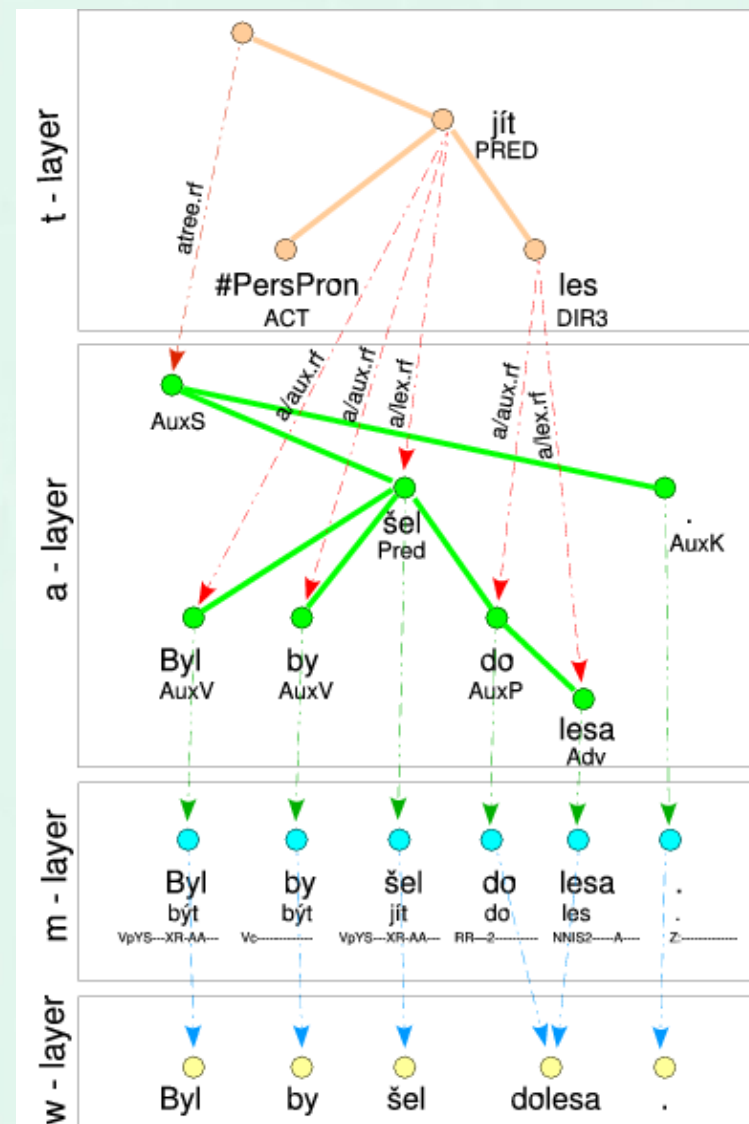
- **tectogrammatical layer**
deep-syntactic dependency trees
- **analytical layer**
surface-syntactic dependency trees, labeled edges
- **morphological layer**
lemma & POS tag for each word
- **word layer**
raw (tokenized) text



4 layers of language description

implemented in Prague Dependency Treebank (PDT)

- **tectogrammatical layer**
deep-syntactic dependency trees
- abstraction from many language-specific phenomena
- autosemantic (meaningful) words
~ **nodes**
- functional words (prepositions, auxiliaries)
~ **attributes**
- syntactic-semantic relations (dependencies)
~ **edges**
- added nodes (e.g. because of pro-drop)
- ...



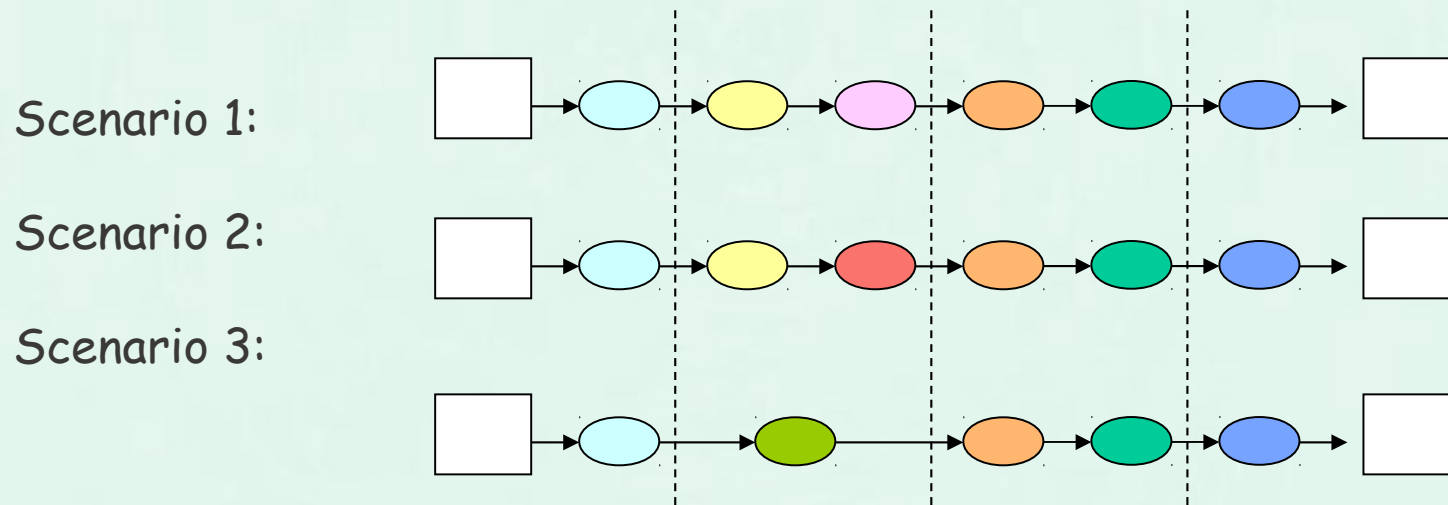
TectoMT architecture processing units



- **block** – elementary processing unit in TectoMT
 - corresponding to a given NLP subtask
 - one Perl class, saved in one file
- **scenario** – a sequence of blocks
 - saved in plain text files
 - just a list of the blocks' names and their parameters
- **application** – represents an end-to-end NLP task
 - conversion of the input to TectoMT internal format (XML)
 - possibly split into more files
 - applying a scenario to the files (loaded in memory)
 - conversion to the desired output format

TectoMT architecture processing units

Blocks can be easily substituted with an alternative solution.



TectoMT architecture processing units



Blocks can be easily substituted with an alternative solution.

Scenario A

`Sentence_segmentation_simple`

`Penn_style_tokenization`

`TagMxPost`

`Lemmatize_mtree`

`McD_parser`

Scenario B

`Each_line_as_sentence`

`Tokenize_and_tag`

`Lemmatize_mtree`

`Malt_parser`

TectoMT architecture

data units



- **Document**
 - stored in one file
 - sequence of sentences
- **Bundle**
 - corresponds to one sentence
 - “bundle of trees”
- **Tree**
 - direction (S=source, T=target)
 - Language (Arabic, Czech, English, German,...)
 - layer of language description (M, A, T)

TectoMT architecture data units



DOCUMENT

sentence 1

sentence 2

...

sentence N

BUNDLE

BUNDLE

BUNDLE

English layers

Czech layers

SEnglishW

Peter does not love Mary.

SCzechW

Petr nemiluje Marii.

SEnglishM

● ● ● ● ●
Peter do not love Mary
NNP VBZ RB VBD NNP

SCzechM

● ● ●
Petr milovat Marie
NNMS1 VB-S—3P-NA NNFS4

SEnglishA

● ● ● ● ●
Peter do not love Mary
Sb AuxV Neg Pred Obj

SCzechA

● ● ●
Petr milovat Marie
Sb Pred Obj

SEnglishT

● ● ●
Peter love Mary
ACT PRED PAT

SCzechT

● ● ●
Petr milovat Marie
ACT PRED PAT

TectoMT architecture data units

DOCUMENT

sentence 1

sentence 2

...

sentence N

BUNDLE

BUNDLE

BUNDLE

English layers

Czech layers

S_{English}W

Peter does not love Mary.

S_{Czech}W

Petr nemiluje Marii.

S_{English}M

● ● ● ●
Peter do not love Mary
NNP VBZ RBD NNP

S_{Czech}M

● ●
Petr miluje Marii
NNMS1 VB-S NA NNFS4

S_{English}A

● ● ● ●
Peter do not love Mary
Sb AuxV NP Pred Obj

S_{Czech}A

● ●
Petr miluje Marii
Sb Pred Obj

S_{English}T

● ● ●
Peter love Mary
ACT PRED PAT

S_{Czech}T

● ● ●
Petr milovat Marie
ACT PRED PAT

TectoMT architecture data units



DOCUMENT

sentence 1

BUNDLE

English layers

Czech layers

SEnglishW

Peter does not love Mary.

TCzechW

Petr nemiluje Marii.

SEnglishM

● ● ● ● ●
Peter do not love Mary
NNP VBZ RB VBD NNP

TCzechM

● ● ●
Petr milovat Marie
NNMS1 VB-S—3P-NA NNFS4

SEnglishA

● ● ● ● ●
Peter do not love Mary
Sb AuxV Neg Pred Obj

TCzechA

● ● ●
Petr milovat Marie
Sb Pred Obj

SEnglishT

● ● ●
Peter love Mary
ACT PRED PAT

TCzechT

● ● ●
Petr milovat Marie
ACT PRED PAT

sentence 2

BUNDLE

...

sentence N

BUNDLE

...

TectoMT architecture data units

DOCUMENT

sentence 1

sentence 2

...

sentence N

BUNDLE

BUNDLE

BUNDLE

English layers

Czech layers

SEnglishW

Peter does not love Mary.

TCzechW

Petr nemiluje Marii.

SEnglishM

● ● ● ●
Peter do not love Mary
NNP VBZ P BD NNP

TCzechM

● ● ● ●
Petr miluje Marii
NNMS1 VB-IP-NA NNFS4

SEnglishA

● ● ● ● ● ●
Peter do not love Mary
Sb AuxV NP-NA Pred Obj

TCzechA

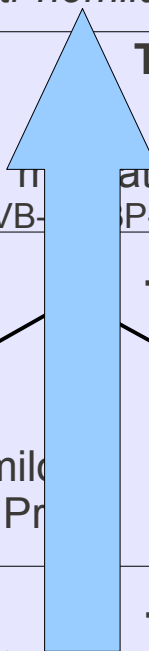
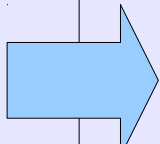
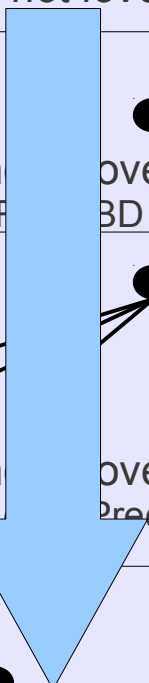
● ● ● ●
Petr miluje Marii
Sb Prj Obj

SEnglishT

● ● ●
Peter love Mary
ACT PRED PAT

TCzechT

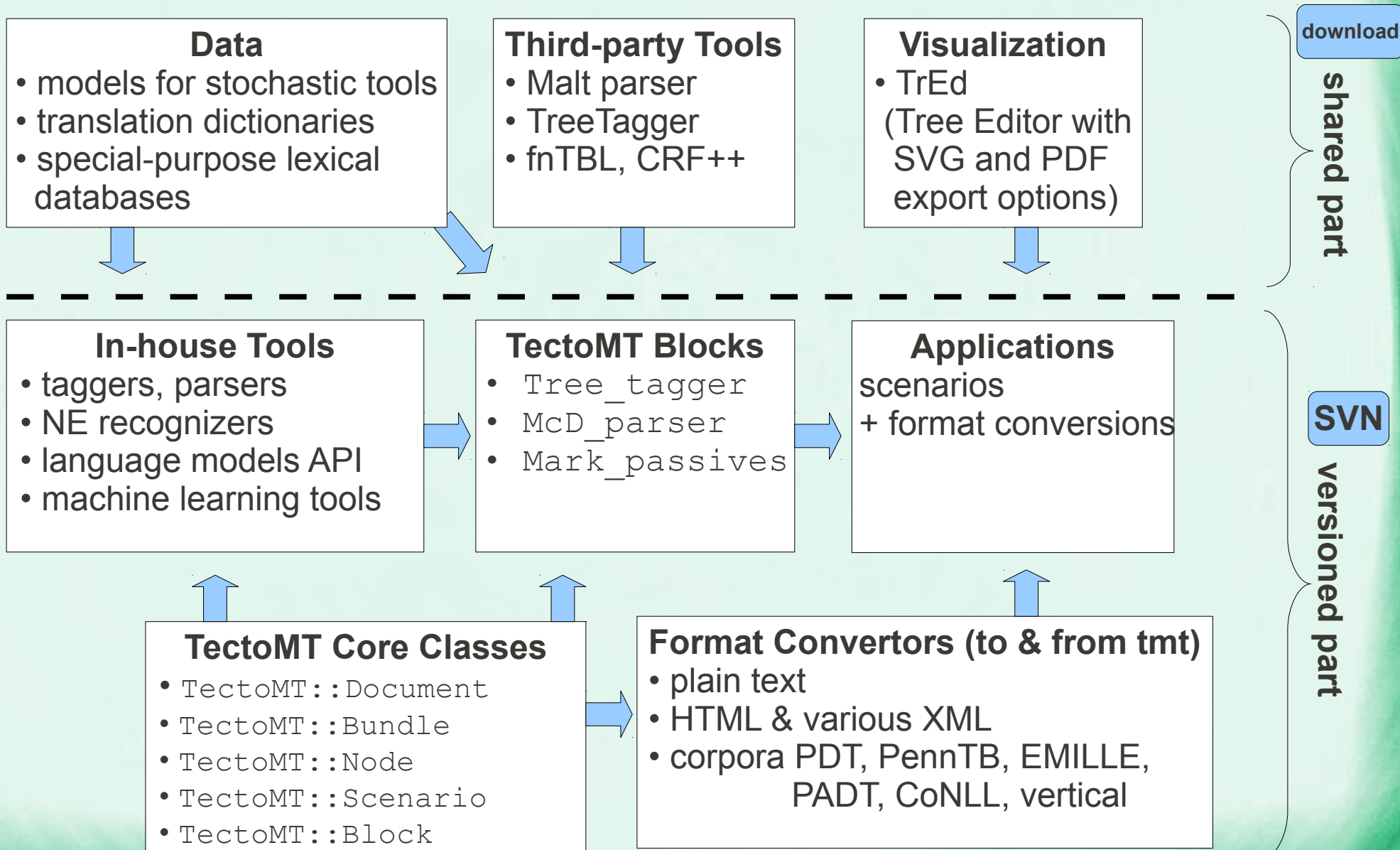
● ● ●
Petr milovat Marie
ACT PRED PAT



Internals – Design decisions

- Perl (wrappers for binaries, Java,...)
- Linux (some applications platform-independent)
- OOP (ClassStd, Moose)
- Open source (GNU GPL for the versioned part)
- Neutral w.r.t. methodology (statistical, rule-based)
- Multilingual
- Open standards (Unicode, XML)

Internals – Components



Internals – Statistics

- Developed since 2005, over ten developers
- Over 400 blocks (140 English, 120 Czech, 60 English-to-Czech, 30 other languages, 50 language independent)
- Taggers (5 English, 3 Czech, 1 German and Russian)
Parsers (Dep. 2 English, 3 Czech, 2 German; Const. 2 English)
Named Entity Recognizers (2 Czech, 1 English)
- Speed example: Best version of English-to-Czech MT
1.2 seconds per sentence plus 90 seconds loading,
with 20 computers in cluster: 2000 sentences in 4 min

Internals – Statistics

- Developed since 2005, over ten developers
- Over 400 blocks (140 English, 120 Czech, 60 English-to-Czech, 30 other languages, 50 language independent)
- Taggers (5 English, 3 Czech, 1 German and Russian, **Tamil**)
Parsers (Dep. 2 English, 3 Czech, 2 German; Const. 2 English)
Named Entity Recognizers (2 Czech, 1 English)
- Speed example: Best version of English-to-Czech MT
1.2 seconds per sentence plus 90 seconds loading,
with 20 computers in cluster: 2000 sentences in 4 min

Future plans (TectoMT → Treex)



- Reimplementation of core components
- CPAN release
- Adding new languages more easily
- Improved parallelization support
- Faster code, smaller files,...

Conclusion

TectoMT main properties

- emphasized efficient development, modular design and reusability
- stratificational approach to the language
- unified object-oriented interface for accessing data structures
- comfortable development

TrEd visualization

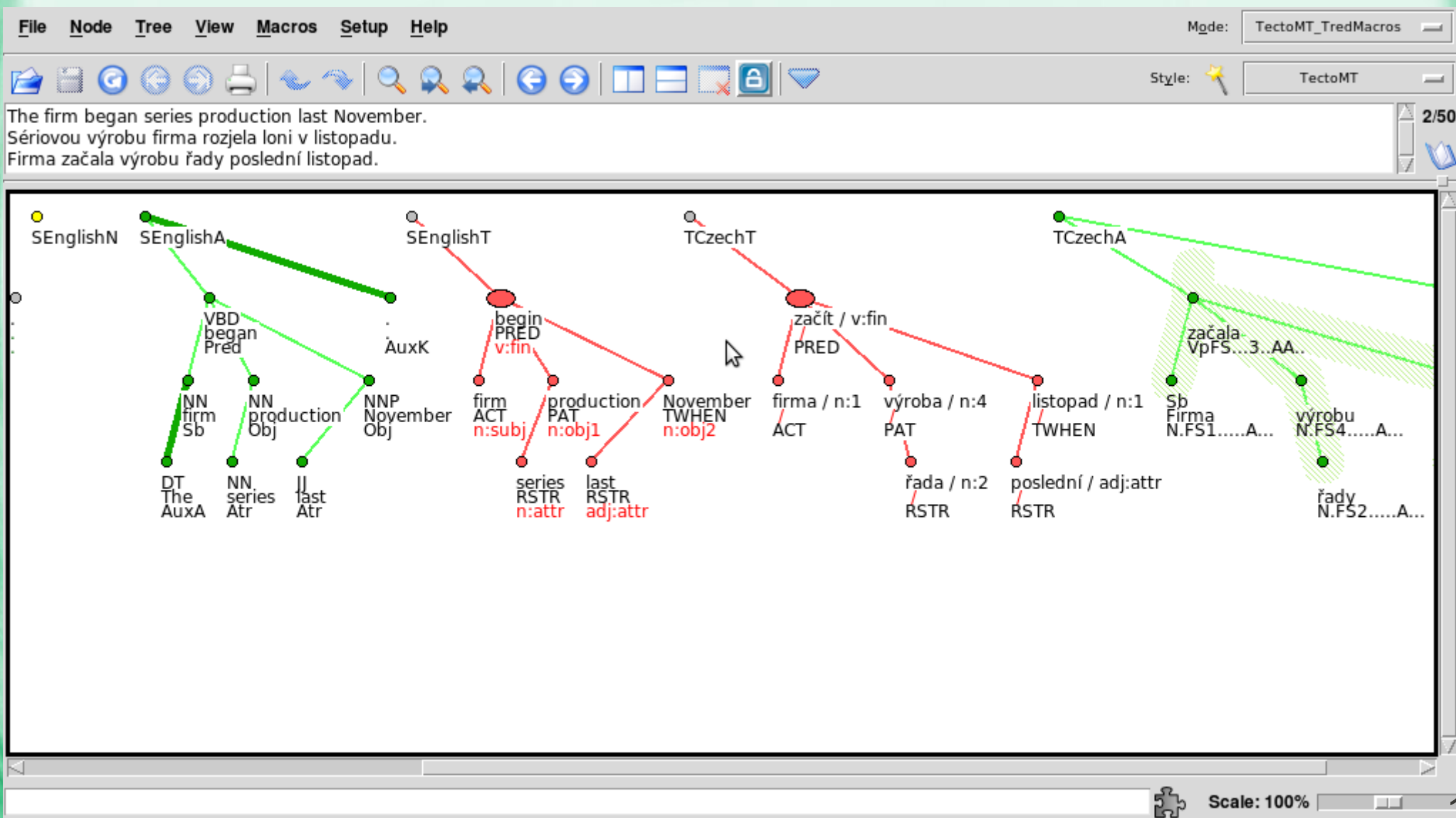
translation

File Node Tree View Macros Setup Help Mode: TectoMT_TredMacros

Style: TectoMT

The firm began series production last November.
 Sériovou výrobu firma rozjela loni v listopadu.
 Firma začala výrobu řady poslední listopad.

2/50



SEnglishN SEnglishA SEnglishT TCzechA TCzechA

VBD began Pred
 NN firm Sb
 NN production Obj
 NNP November Obj
 DT The AuxA
 NN series Atr
 JJ last Atr

begin PRED v:fin
 firm ACT n:subj
 production PAT n:obj1
 November TWHEN n:obj2
 series RSTR n:attr
 last RSTR adj:attr

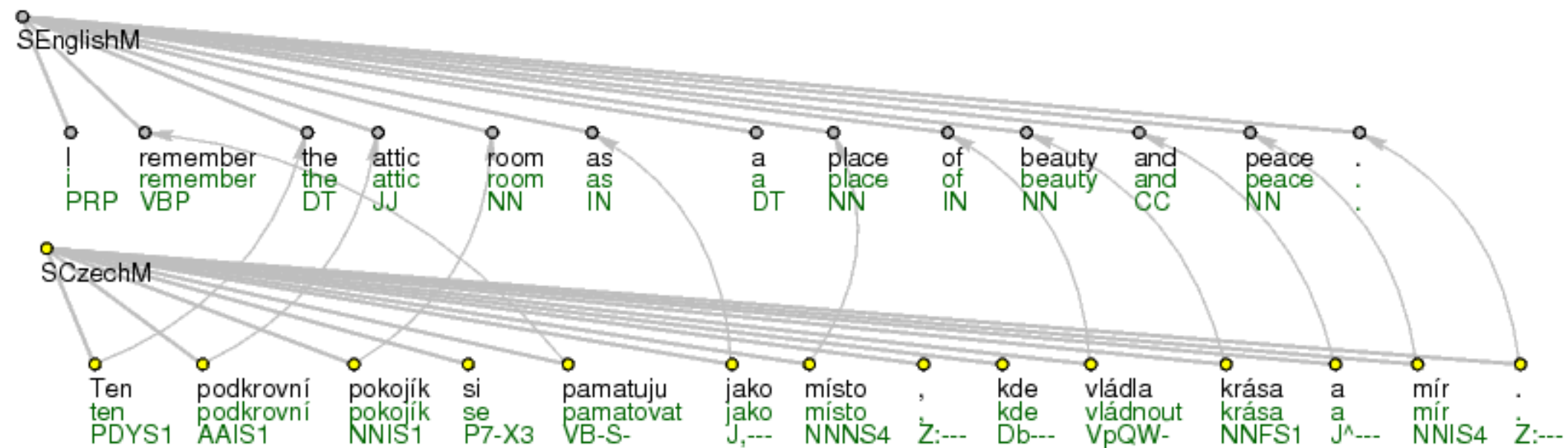
začít / v:fin PRED
 firma / n:1 ACT
 výroba / n:4 PAT
 listopad / n:1 TWHEN
 řada / n:2 RSTR
 poslední / adj:attr RSTR

začala VpFS...3..AA..
 Sb Firma N.FS1.....A..
 výrobu N.FS4.....A..
 řady N.FS2.....A..

Scale: 100%

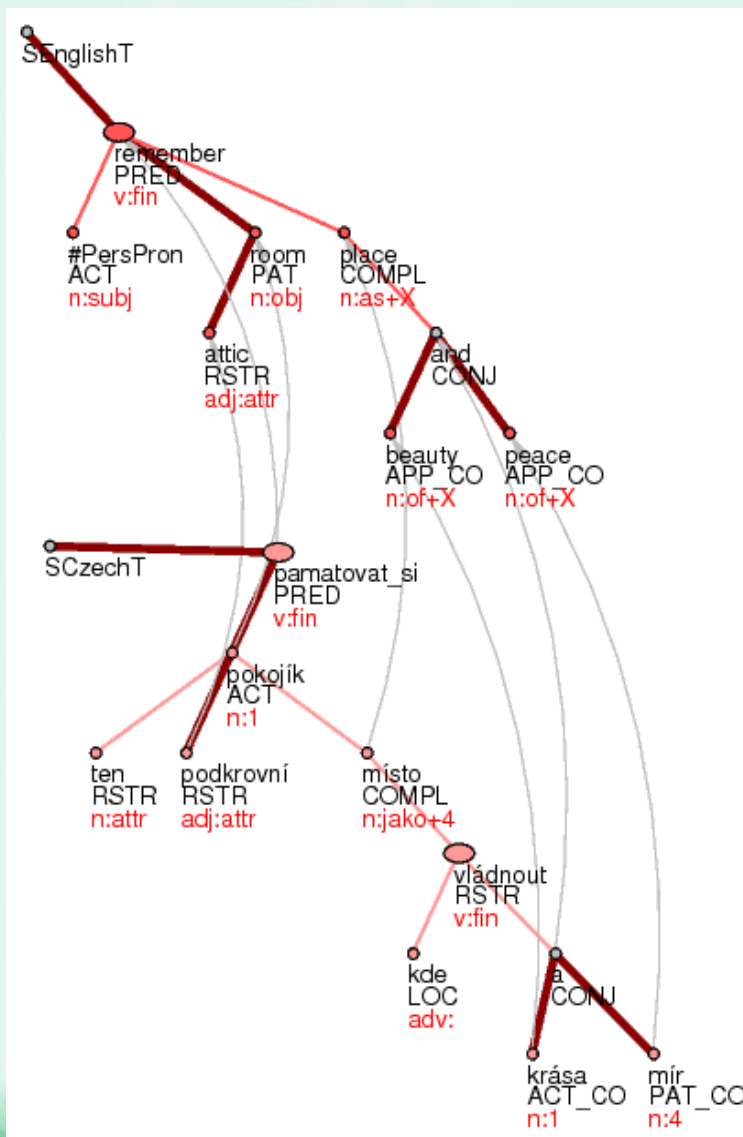
TrEd visualization

word alignment on the morphological layer



TrEd visualization

word alignment on the tectogrammatical layer



Block example – SVO to SOV code

```
package Tutorial::SVO_to_SOV_solution;
use strict; use warnings;
use base qw(TectoMT::Block);

sub process_bundle {
    my ( $self, $bundle ) = @_;
    my $a_root = $bundle->get_tree('SEnglishA');

    foreach my $a_node ( $a_root->get_descendants() ) {
        if ( $a_node->get_attr('m/tag') =~ /^V/ ) { # verb found
            foreach my $schild ( $a_node->get_eff_children() ) {
                if ( $schild->get_attr('afun') eq 'Obj' ) { # object found
                    # Move the object and its subtree so it precedes the verb
                    $schild->shift_before_node($a_node);
                }
            }
        }
    }
}
1;
```

TectoMT core
TectoMT convention
Perl keyword/convention

Thank you

Cooperation is welcomed.



<http://ufal.mff.cuni.cz/tectomt>