TectoMT

Software framework for developing MT systems (and other NLP applications)

Zdeněk Žabokrtský
ÚFAL MFF UK

What is TectoMT

- TectoMT is …
  - a highly modular extendable **NLP software system**
  - composed of numerous NLP tools integrated into a **uniform infrastructure**
  - aimed at (not limited to) developing Machine Translation systems
- TectoMT is not …
  - **a specific method of MT** (even if some approaches can profit from its existence more than others)
  - **an end-user application** (even if releasing of single-purpose stand-alone applications is possible and technically supported)

Motivation for creating TectoMT

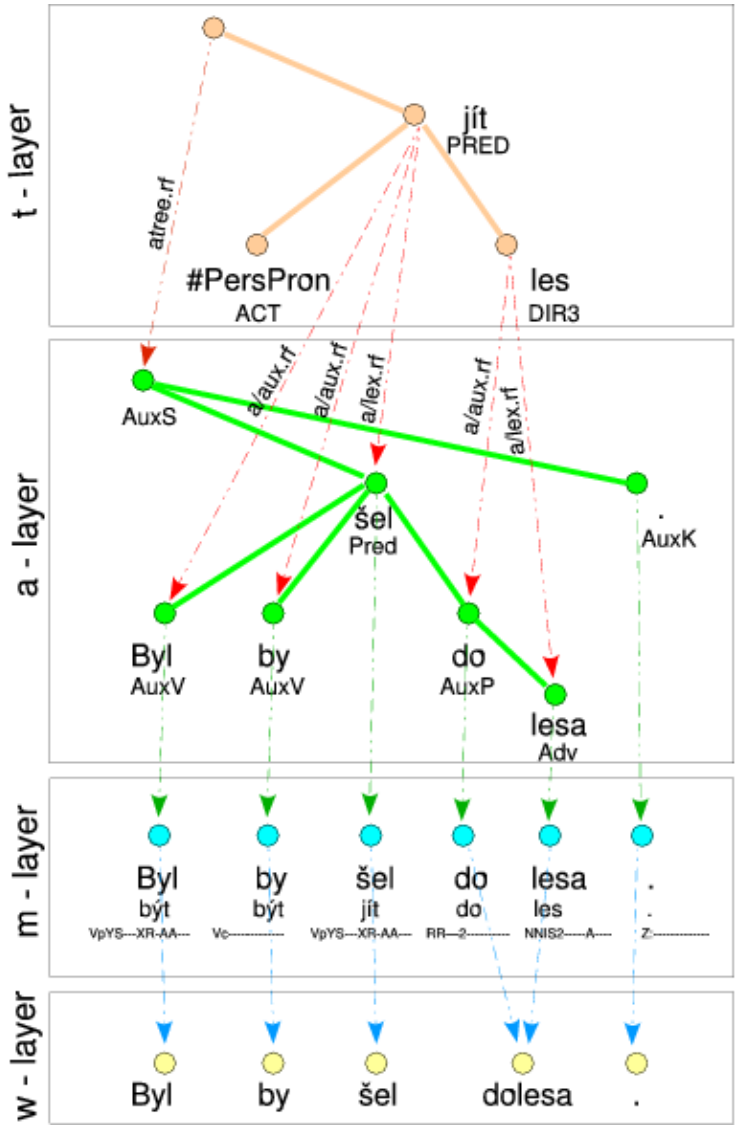
- First, technical reasons:
  - Want to make use of more than two NLP tools in your experiment? Be ready for endless data conversions, need for other people's source code tweaking, incompatibility of source code and model versions…
  - Unified software infrastructure might help us.

- Second, our long-term MT plan:
  - We believe that tectogrammar (deep syntax) as implemented in Prague Dependency Treebank might help to (1) **reduce data sparseness**, and (2) find and **employ structural similarities** revealed by tectogrammar even between typologically different languages.

Prague Dependency Treebank 2.0

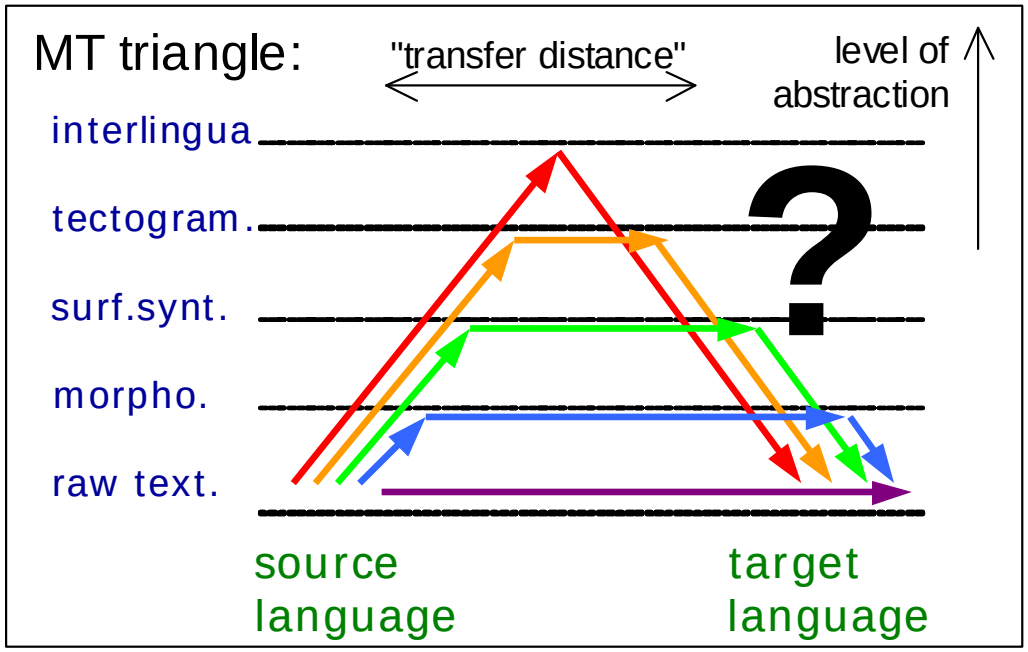
- three layers of annotation:
- tectogrammatical layer
  - deep-syntactic dependency tree
- analytical layer
  - surface-syntactic dependency tree
  - 1 word (or punct.) ~ 1 node
- morphological layer
  - sequence of tokens with their lemmas and morphological tags

[Ex: *He would have gone into forest*]



MT triangle in terms of PDT

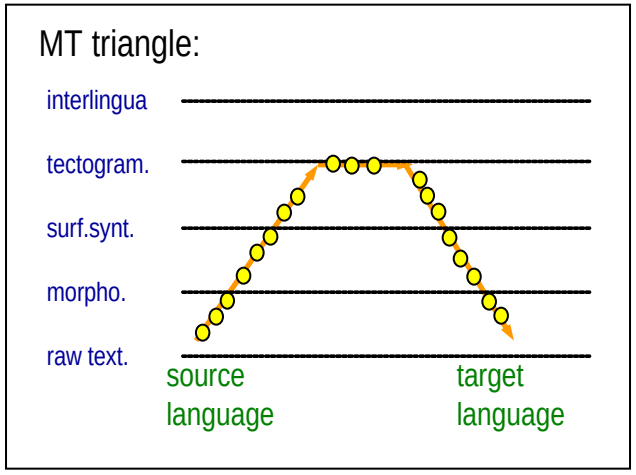
- Key question of MT: what is the optimal level of abstraction?



- Obvious trade-off: ease of transfer vs. additional analysis and synthesis costs (system complexity, errors...)

Design decisions

- **Linux + Perl**
- set of well-defined, **linguistically relevant layers** of language representation
- **neutral** w.r.t. chosen methodology ("rules vs. statistics")
- accent on modularity: translation **scenario** as a sequence of translation **blocks** (modules corresponding to individual NLP subtasks)
  - reusability
  - substitutability

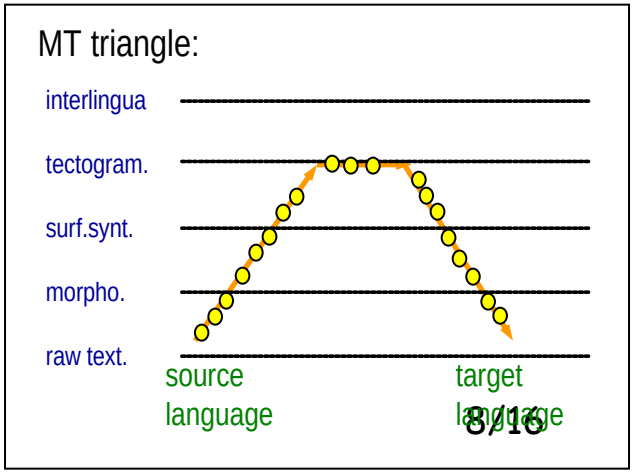


Design decisions (cont.)

- in-house **object-oriented architecture** as the backbone
  - all tools communicate via standardized OO Perl interface
  - avoiding the former practice of tools communicating via files in specialized formats
- easy **incorporation of external tools**
  - previously existing parsers, taggers, lemmatizers etc.
  - just provide them with a Perl "wrapper" with the prescribed interface

Hierarchy of processing units

- **block**
  - the smallest individually executable unit
  - with well-defined input and output
  - block parametrization possible (e.g. model size choice)
- **scenario**
  - sequence of blocks, applied one after another on given documents
- **application**
  - typically 3 steps:
    - 1. conversion from the input format
    - 2. applying the scenario on the data
    - 3. conversion into the output format



Blocks

- technically, Perl classes derived from a common ancestor class
- around 300 blocks in TectoMT now, for various purposes:
  - blocks for analysis/transfer/synthesis, e.g.
    `SEnglishW_to_SEnglishM::Lemmatize_mtree`
    `SEnglishP_to_SEnglishA::Mark_heads`
    `TCzechT_to_TCzechA::Vocalize_prepositions`
  - blocks for alignment, evaluation, feature extraction, etc.
- English-Czech tecto-based translation currently composes of roughly 80 blocks

Tools integrated as blocks

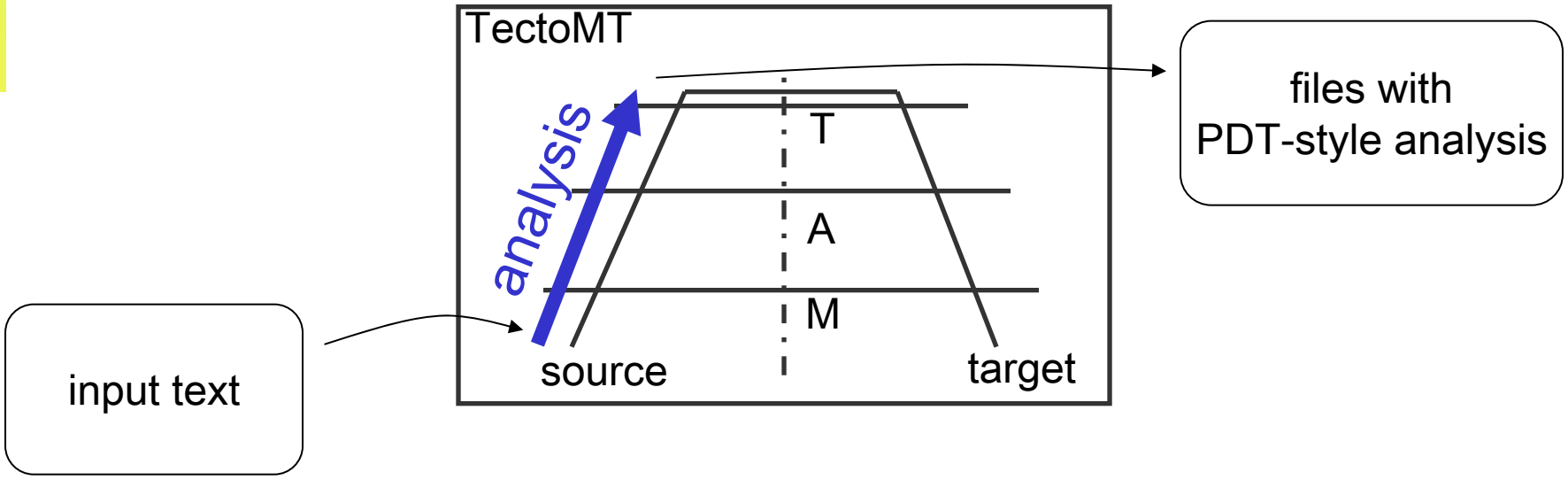
- already integrated tools:
  - taggers
    - Hajič's tagger, Raab&Spoustová Morče tagger, Rathnaparkhi MXPOST tagger, Brants's TnT tager, Schmid's Tree tagger, Coburn's Lingua::EN::Tagger
  - parsers
    - Collins' phrase structure parser, McDonalds dependency parser, ZŽ's dependency parser
  - named-entity recognizers
    - Stanford Named Entity Recognizer, Kravalová's SVM-based NE recognizer
  - etc.

Other TectoMT components

- numerous **file-format convertors** (e.g. from PDT, Penn treebank, Czeng corpus, WMT shared task data etc. to our xml format)
- TectoMT-customized Pajas' **tree editor TrEd** (visualization)
- tools for **parallelized processing** (Bojar)
- tools for testing (regular daily tests), documentation...

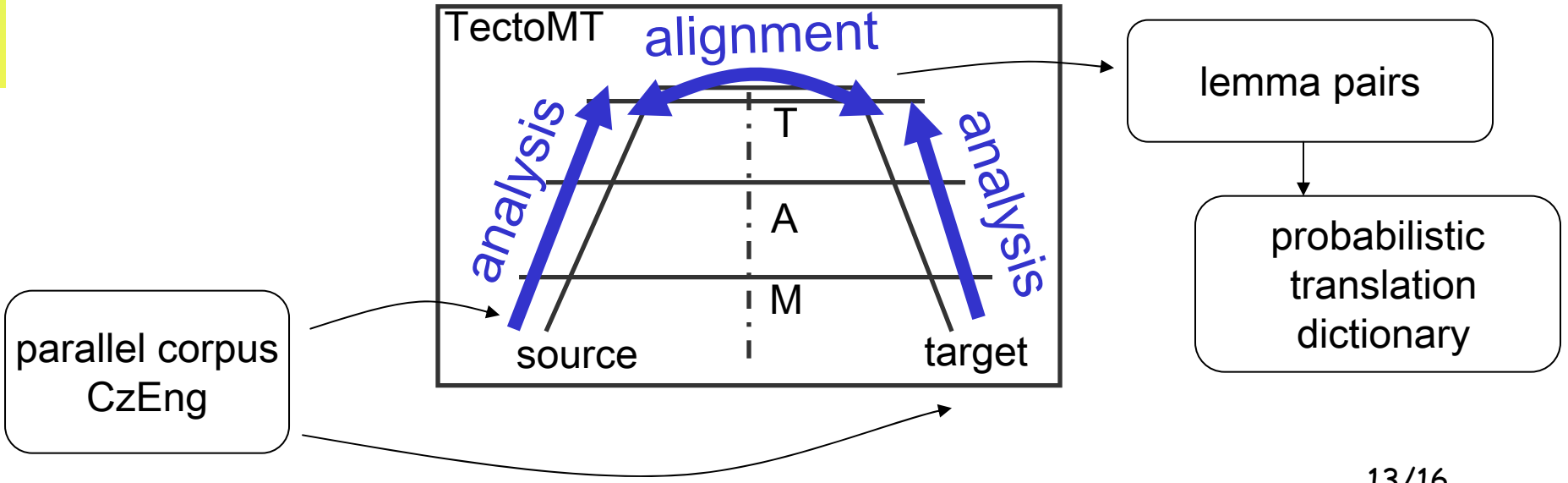
PDT-style layered analysis

- analyze a given Czech or English text up to morphological, analytical and tectogrammatical layer
- used currently e.g. in experiments with intonation generation, information extraction, man-machine dialog system



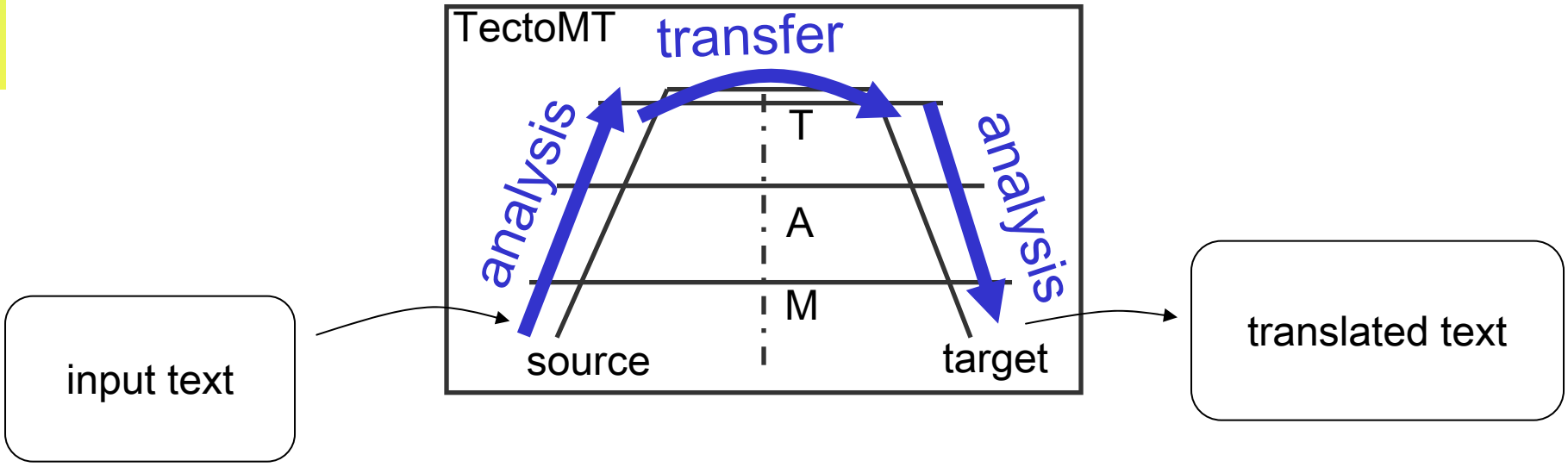
Transl. dictionary extraction

- using the lemma pairs from the aligned t-nodes from a huge parallel corpus, we build a probabilistic translation dictionary



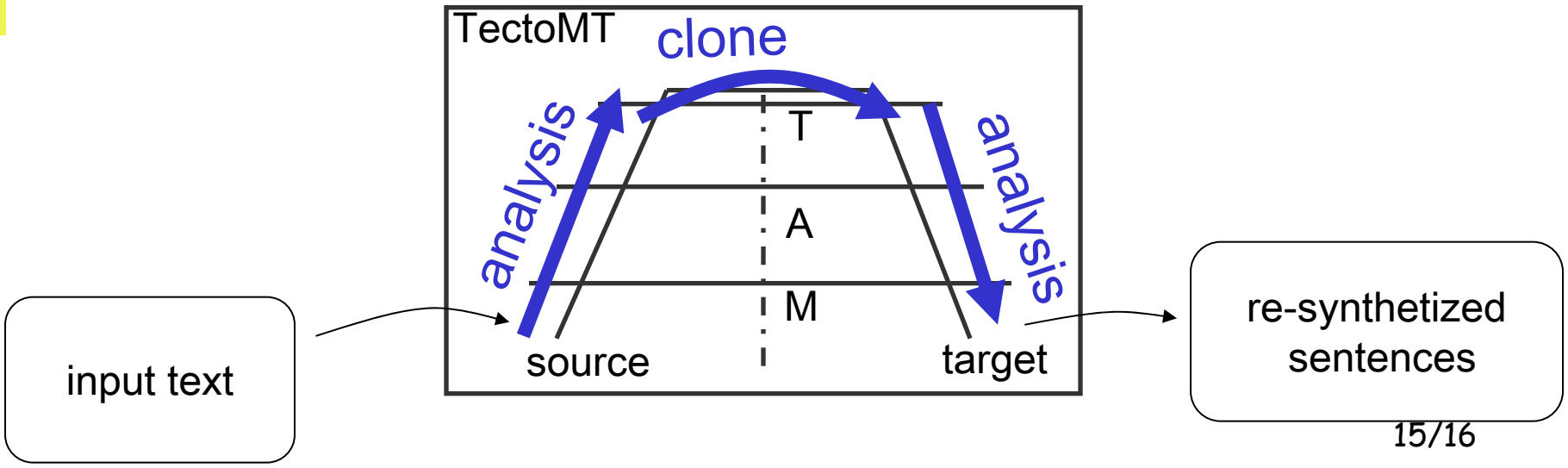
Translation with tecto-transfer

- analysis-transfer-synthesis translation from English to Czech and vice versa
- employed probabilistic dictionary from the previous slide



Sentence re-synthesis

- analysis-clone-synthesis scenario for
  - **postprocessing of other MT system's output** (to make it more grammatical)
  - **speech reconstruction** - postprocessing of STT's output (to make it more grammatical)



Final remarks

- TectoMT developed since 2005.
- TectoMT available under GPL since 2009.
- At the moment around 15 programmers using/contributing to TectoMT.
- Want to analyze Czech or English texts? Why don't you try doing it in your own TectoMT svn working copy?

- http://ufal.mff.cuni.cz/tectomt