Tree-based Translation with Tectogrammatical Representation Part 2: Treelet Decoding

Ondřej Bojar
bojar@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague

Overview of Part 2

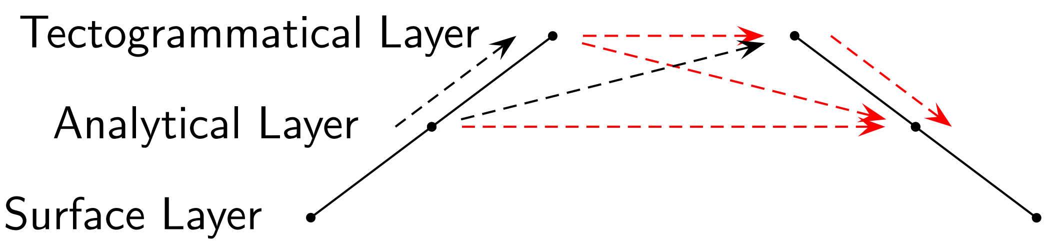
- Synchronous Tree-Substitution Grammars (STSG).
  - Illustrations, definitions,
  - Tree-to-tree alignments by heuristics or EM,
  - Beam-search decoding of STSG.
- Risks of data sparseness and back-off methods.
- Properties of the current version of my decoder.

Synchronous Tree-Substitution Grammars (STSG)

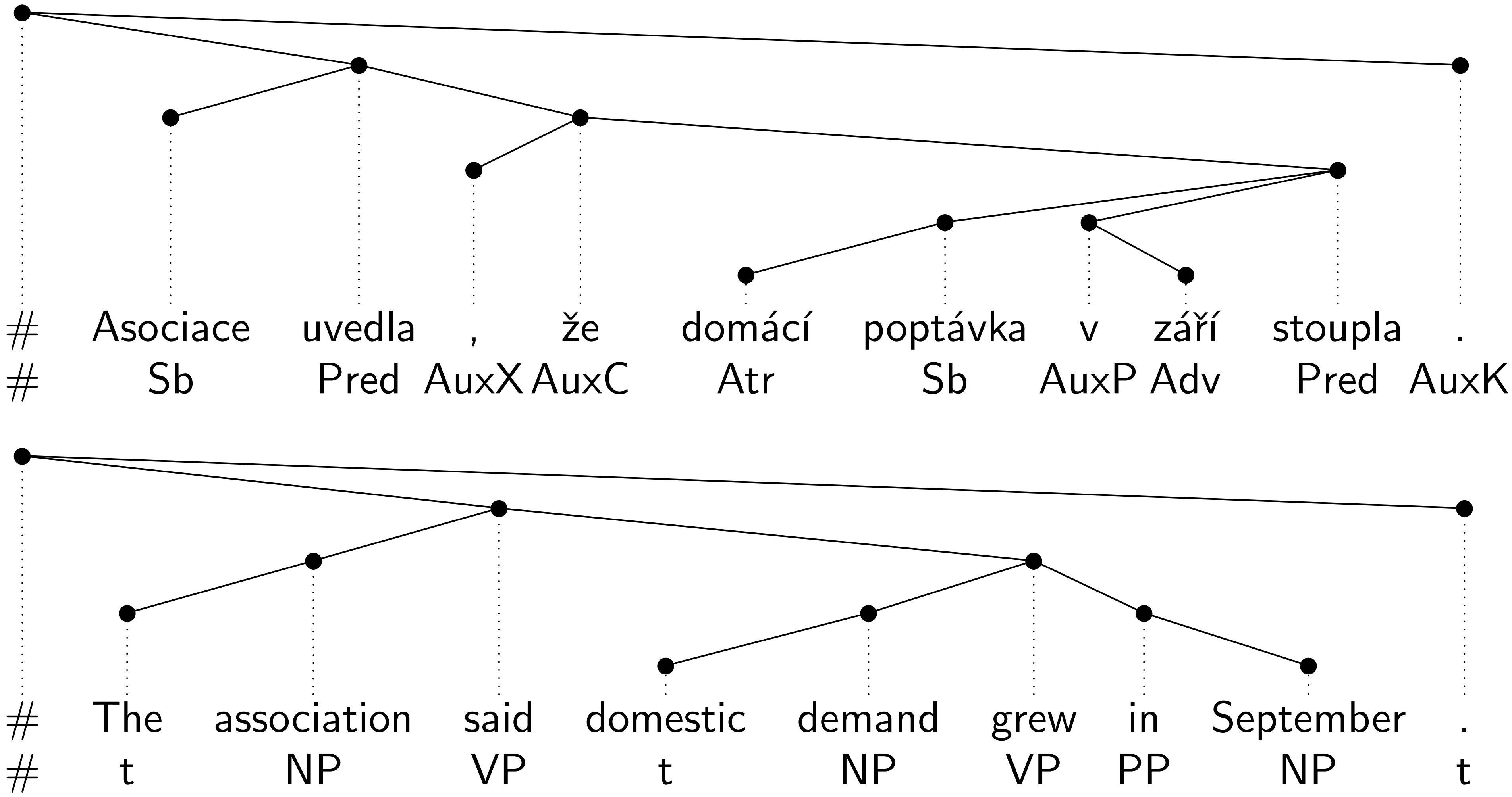
- Introduced by Hajič et al. (2002) and formalized by Eisner (2003) and Čmejrek (2006).
- Basic assumption when applied to MT: source and target sentences are structurally parallel.

  Not all training sentences are like that, because not all translations are literal enough.
- Generic model for non-isomorphic tree-to-tree transformation.

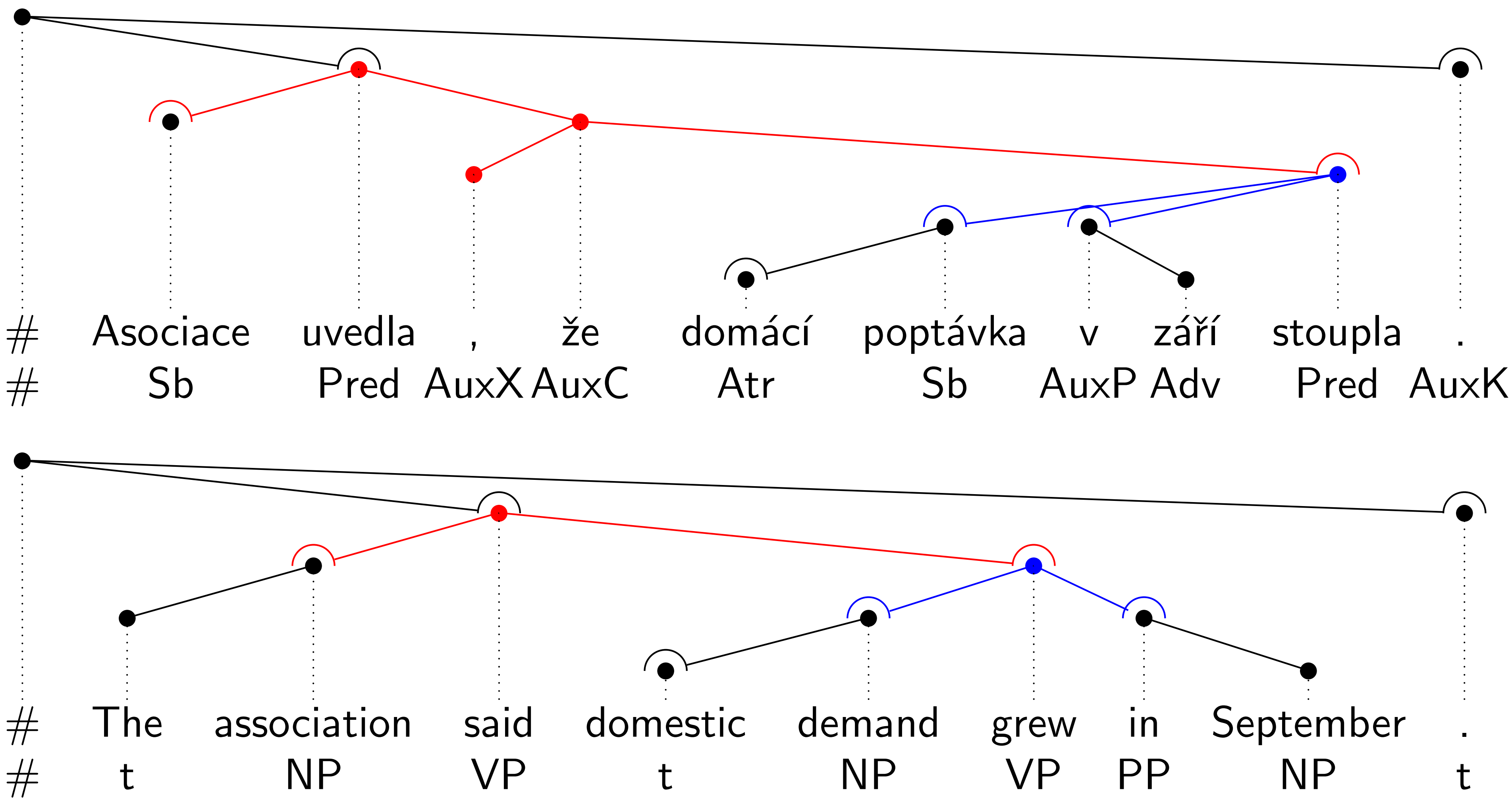
  Can be applied at or across various layers:



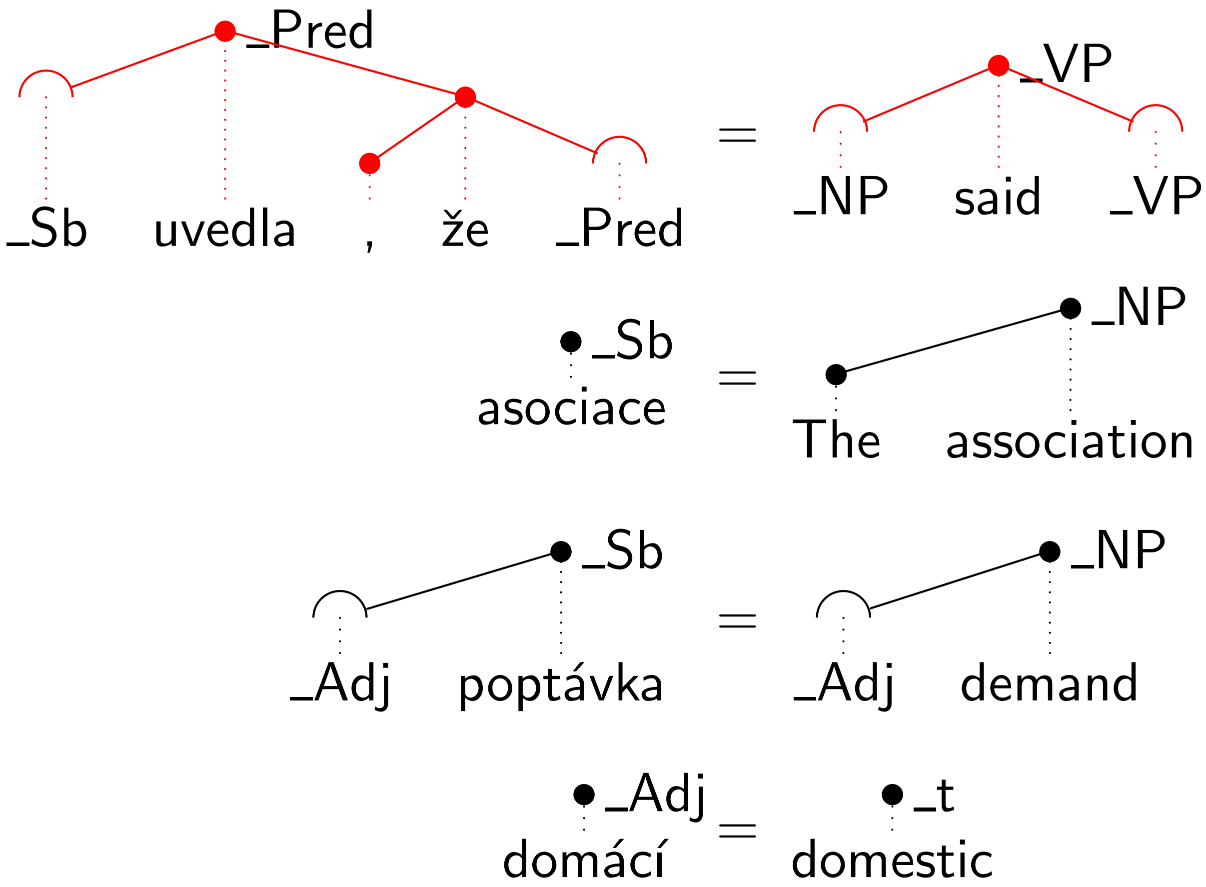
Idea: Observe a Pair of Dependency Trees



Idea: Decompose Trees into Treelets



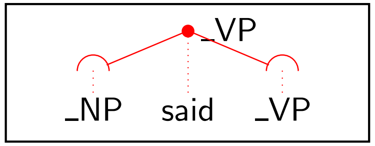
Idea: Collect Dictionary of Treelet Pairs



Little Trees Formally

Given a set of states Q and a set of word labels L, we define:

A LITTLE TREE or TREELET t is a tuple (V, V^i, E, q, l, s) where:



- V is a set of NODES,
- $V^i \subseteq V$ is a nonempty set of INTERNAL NODES. The complement $V^f = V \backslash V^i$ is called the set of FRONTIER NODES,
- $E \subseteq V^i \times V$ is a set of directed edges starting from internal nodes only and forming a directed acyclic graph,
- $q \in Q$ is the ROOT STATE,
- $l : V^i \rightarrow L$ is a function assigning labels to internal nodes,
- $s : V^f \rightarrow Q$ is a function assigning states to frontier nodes.

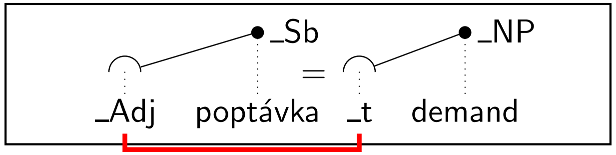
Optionally, we can keep track of local or global ordering of nodes in treelets.

I depart from Čmejrek (2006) in a few details, most notably I require at least one internal node in each little tree.

Treelet Pair Formally, Synch. Derivation

A TREELET PAIR $t_{1:2}$ is a tuple (t_1, t_2, m) where:

- t_1 and t_2 are little trees for source and target languages (L_1 and L_2) and states (Q_1 and Q_2),
- m is a 1-1 MAPPING of frontier nodes in t_1 and t_2.



  Unlike Čmejrek (2006), I require all frontier nodes mapped, i.e. equal number of left and right frontier nodes.

From a starting SYNCHRONOUS STATE $Start_{1:2} \in Q_1 \times Q_2$,
a SYNCHRONOUS DERIVATION δ constructs a pair of dependency trees by:

- attaching treelet pairs $t^0_{1:2}, \ldots, t^k_{1:2}$ at corresponding frontier nodes, and
- ensuring that the root states $q^0_{1:2}, \ldots, q^k_{1:2}$ of the attached treelets pairs $t^0_{1:2}, \ldots, t^k_{1:2}$ match the frontier states of the corresponding frontier nodes.

Can define probability of a derivation: $p(\delta) = p(t^0_{1:2}|Start_{1:2}) * \prod_{i=1}^{k} p(t^k_{1:2}|q^k_{1:2})$

Practical Issues

How big should the treelets be?

- The bigger, the better translation. $\times$ The bigger, the worse data sparseness.
- Currently, I consider all up to a certain size (e.g. 3 internals and 7 frontiers).

Given a pair of sentences (trees), how to learn treelet pairs?

- Heuristics similar to common phrase-extraction techniques:
  - Obtain node-to-node(s) alignments.

    Sometimes for free: Tectogrammatical layer contains links to analytical nodes.

    Or use GIZA++ word alignments as node-alignments.
  - Count all treelet pairs somehow compatible with word alignment.
- Expectation-maximization loop: Čmejrek (2006):
  - Assume all possible/reasonable decompositions and alignments equally likely.
  - Recalculate probabilities using corpus counts; iterate.

Decoding STSG

Given an input dependency tree, in all possible ways:

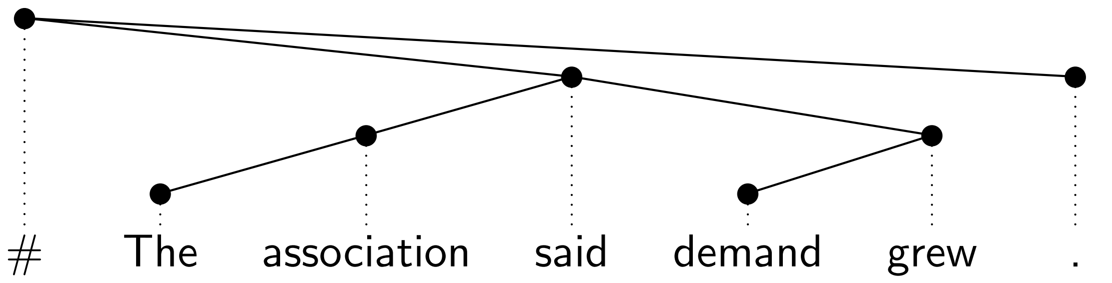
- Decompose it into translatable treelets,
- Replace treelets by their translations,
- Join output treelets and produce output final tree (or string).

Find target tree such that the synch. derivation δ is most likely.

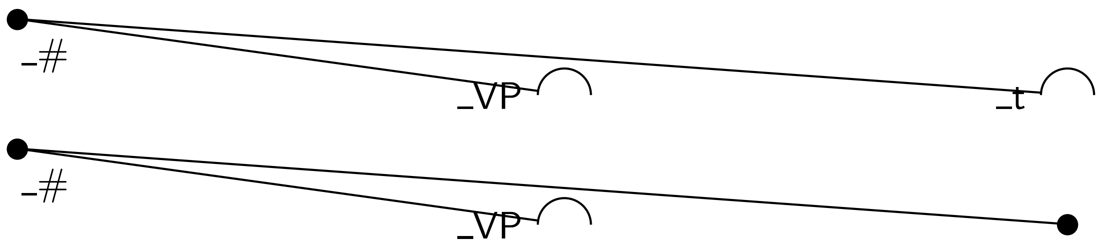
Implemented as top-down beam-search similar to Moses:

For input tree of k words, prepare translation options table:
    For each source node, record τ-best possible target treelets.
Create stacks $s_0, \ldots, s_k$ to hold partial hypotheses, stack s_i for hyps covering i input nodes.
Insert initial hypothesis into s_0.
for $i \in 0 \ldots k-1$
    foreach hypothesis $h \in s_i$
        Expand h by attaching one of possible translation options at a pair of pending frontiers,
            extending the set of covered words and adding output words.
        Insert the expanded h' (j words covered) to s_j, pruning s_j to at most σ hyps.
Output top-scoring h^* from s_k.

Translation Options Example



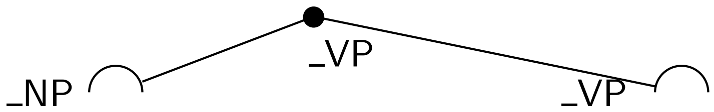
Sample translation options at root:



⇒ # _Pred _AuxK

⇒ # _Pred .

Sample translation options at 'said':

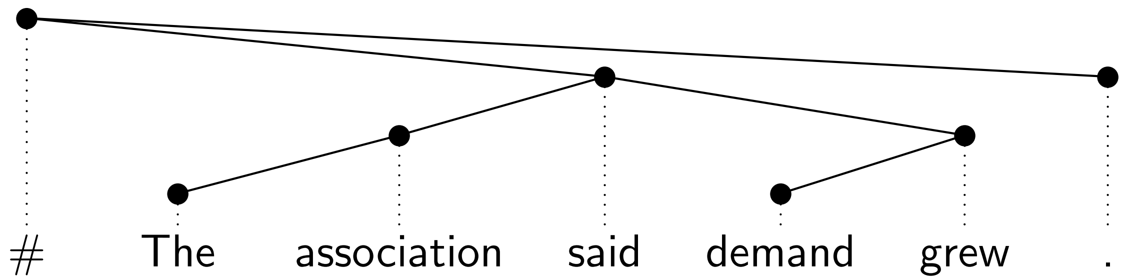


⇒ _Sb uvedla , že _Pred

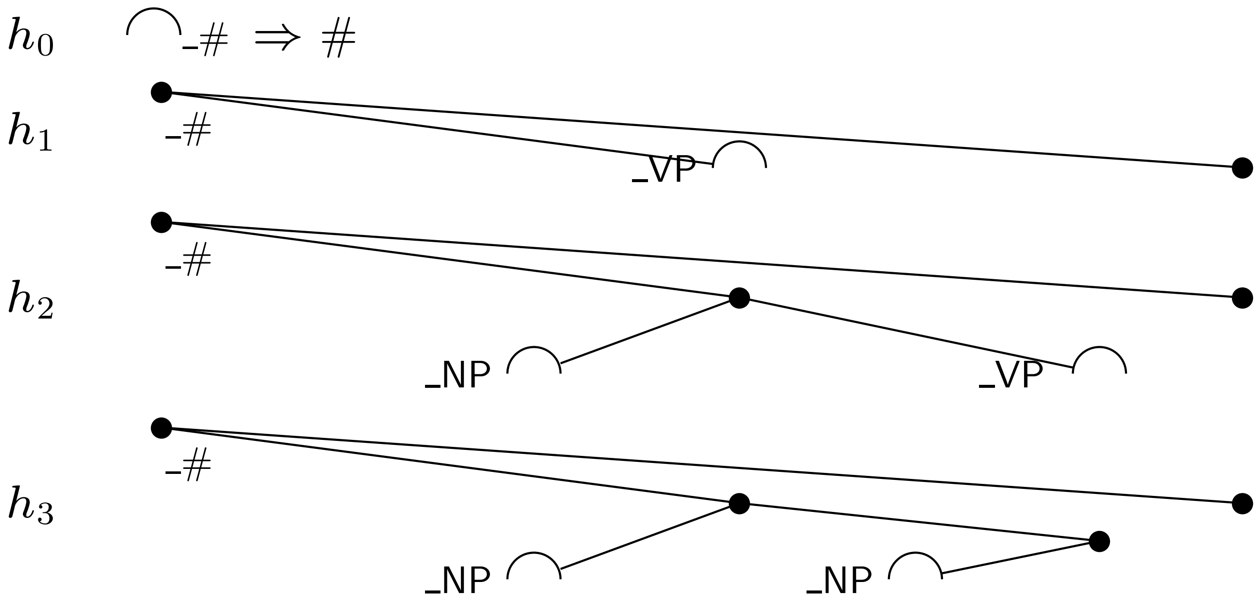
Sample translation options at '.':

⇒ .

Expanding Hypothesis Example



Sample Derivation:



$\Rightarrow$ # _Pred .

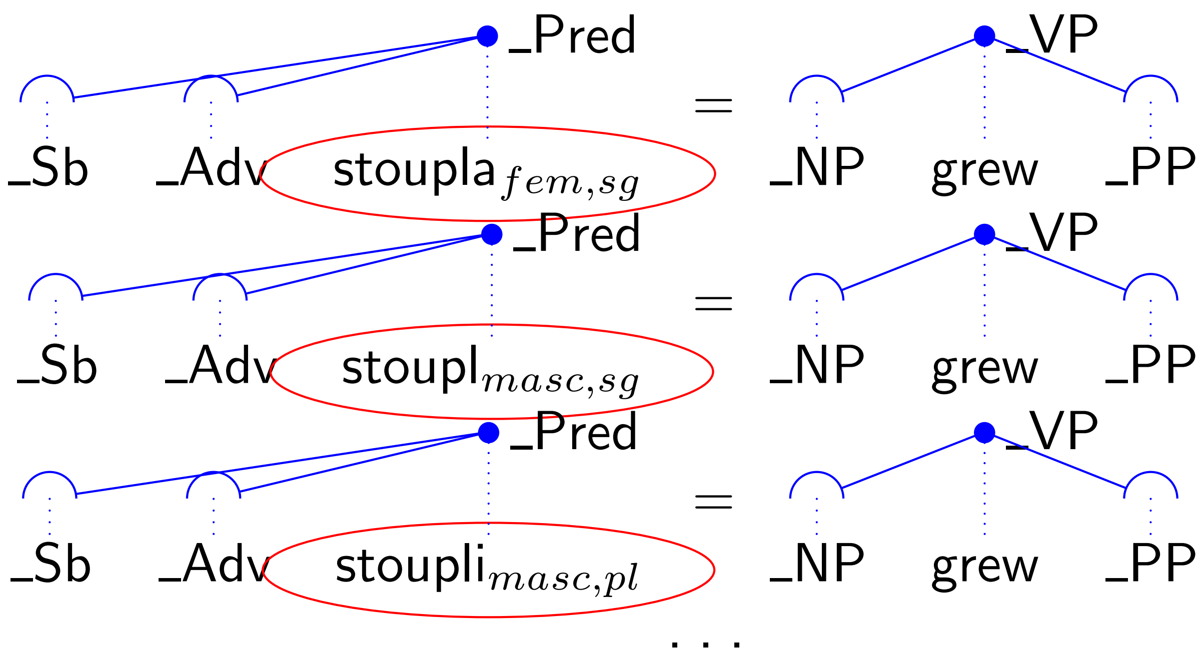
$\Rightarrow$ # _Sb uvedla , že _Pred .

$\Rightarrow$ # _Sb uvedla , že _Sb stoupla .

Risks of Data Sparseness (1)

Morphological richness:

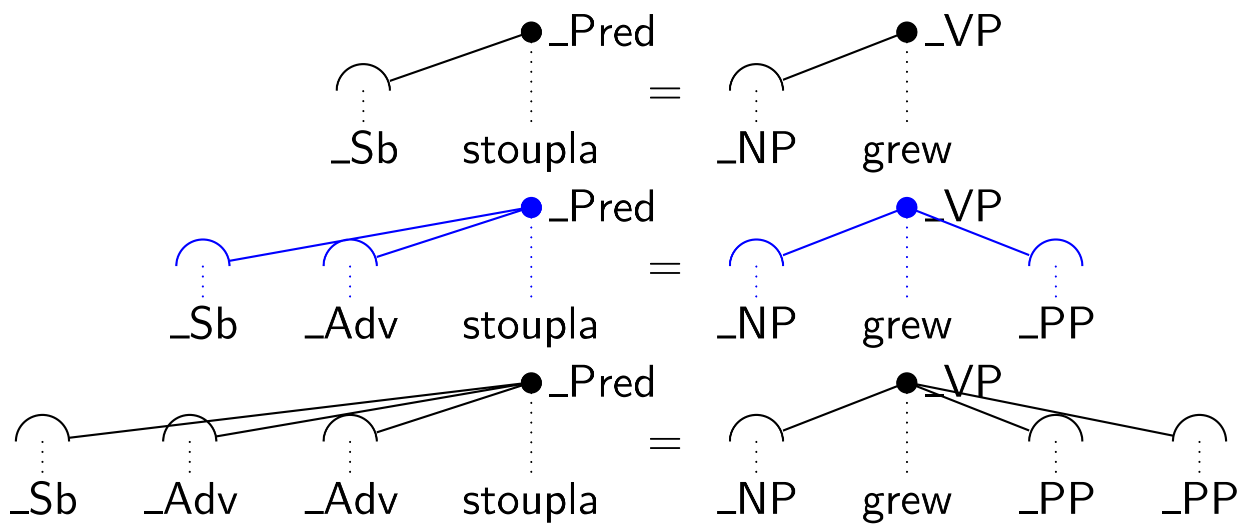
- not an issue at a higher layer, where nodes hold lemmas.



Risks of Data Sparseness (2)

Frontiers for additional adjuncts, state labels for root and frontiers:

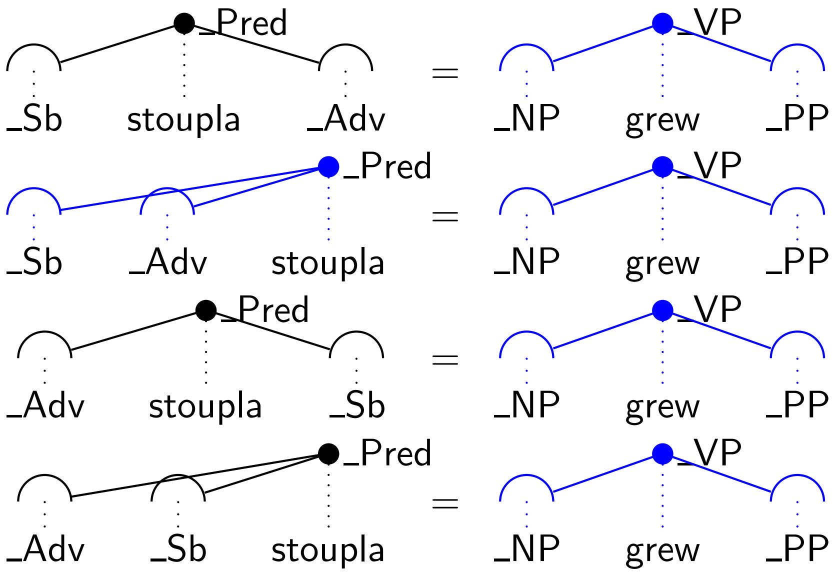
- Once a node is used as internal, all its children have to be included in the little tree as internals or frontiers. (There is no adjunction in STSG.)



Risks of Data Sparseness (3)

Ordering of nodes:

- Czech has a relatively free word order, many permutations possible.
- Not an issue if we decide to leave the tricky part for someone else, e.g. a tecto→analytical generator.



Back-off Schemes

Preserve all. Full-featured treelets are collected in training phase.
Required treelets often never seen in training data $\Rightarrow$ back-off needed.

Drop frontiers. Observed treelets reduced to internal nodes only.
Given a source treelet, internals translated by the dictionary, frontiers generated on the fly, labelled and positioned probabilistically.

Keep a word non-translated to handle unknown words.
Allowed only for single-internal treelets, frontiers mapped probabilistically.

Transfer numeric expression, showing possibility to include hand-coded rules.

Adjoin on the fly like Quirk, Menezes, and Cherry (2005); not implemented.

Modular approach to back-off schemes, config says:

- which methods to use
- in which order, or whether more should be attempted at simultaneously.

Current Experimental Setup

To allow for end-to-end BLEU evaluation, I mainly experiment with:

- analytical trees (treelets fully lexicalized with word forms, locally ordered),
- heuristic treelet dictionary extraction,
- target treelet structure disregarded (output linearized right away).

Features already supported:

- GDBM to store and access treelet tables (zero loading time).
- IrstLM to promote hypotheses containing frequent trigrams.
- MERT by Philipp Koehn (Och, 2003) or Smith and Eisner (2006).

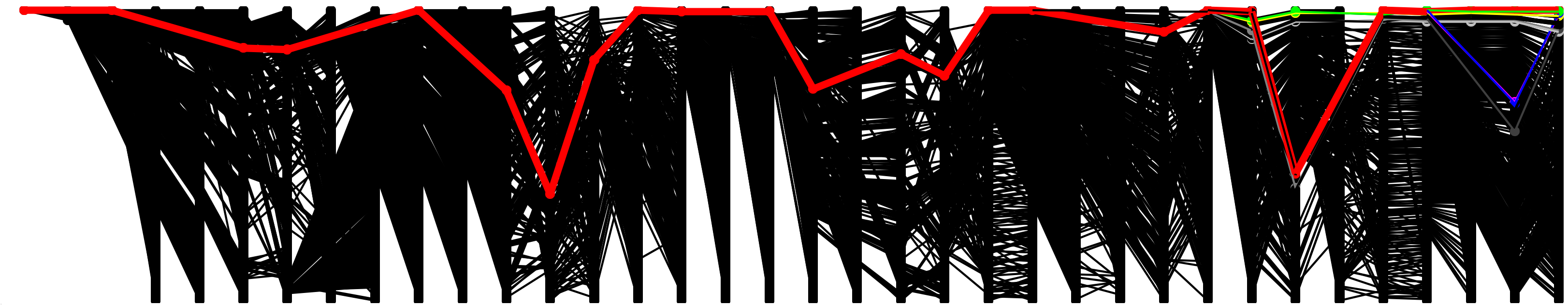
Future:

- Tectogrammatical transfer, chain of transfers.
- Impact of EM training, input parse quality.

Current Problems

- Search errors.

  Input sentence of 35 words, stack size 200. The final best hypothesis (red) ranked as the 126^{th} in stack 12.



- MERT won't work with many search errors.
- Bad parses mislead translation ⇒ plan to allow uncertain input.

⇒ Currently terribly beaten by Moses.
(English⟶Czech BLEU 7 or 8 instead of 13)

Summary

Bigger picture: MT model preserving dependency syntax:

- STSG can be used to model dependency tree-to-tree mapping.
- Linguistically motivated layers reduce sparseness.
  (STSG is applicable at or across various layers: t$\longrightarrow$t, a$\longrightarrow$a, t$_{cs}\longrightarrow$a$_{en}$, t$_{en}\longrightarrow$a$_{en}$.)
- Heuristics or EM to obtain treelet pairs.

"Smaller" picture:

- Czech-English data available at various layers of annotation.
- A preliminary version of an STSG decoder.
  Implemented in Mercury, a functional language compiled to C.
  Sharable with all interested.
  No public release yet, contact me directly. Eventually GPL'd.

Additional Useful Links

bojar@ufal.mff.cuni.cz For all interested in collaboration.

More Czech-English Data `http://ufal.mff.cuni.cz/czeng/`
Czech is a challenge for anyone!

Mercury `http://www.cs.mu.oz.au/research/mercury/`
Pure, functional, (higher order), statically type- and mode-checked
⇒ If it compiles, it runs.
Compiled to plain C
⇒ seamless integration with C/C++ components; efficient.

References

Čmejrek, Martin. 2006. *Using Dependency Tree Structure for Czech-English Machine Translation*. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.

Eisner, Jason. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, pages 205–208, Sapporo, July.

Hajič, Jan, Martin Čmejrek, Bonnie Dorr, Yuan Ding, Jason Eisner, Daniel Gildea, Terry Koo, Kristen Parton, Gerald Penn, Dragomir Radev, and Owen Rambow. 2002. Natural Language Generation in the Context of Machine Translation. Technical report. NLP WS'02 Final Report.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal smt. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279. Association for Computational Linguistics.

Smith, David A. and Jason Eisner. 2006. Minimum-risk annealing for training log-linear models. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL), Companion Volume*, pages 787–794, Sydney, July.