The Hot, Dusty, Corrugated Road to Deep Syntactic Machine Translation

Ondřej Bojar
obo@cuni.cz

October 26, 2007

Outline

(Hot)

- Phrase-Based vs. Deep Syntactic Machine Translation
- Tectogrammatical Representation of Czech and English

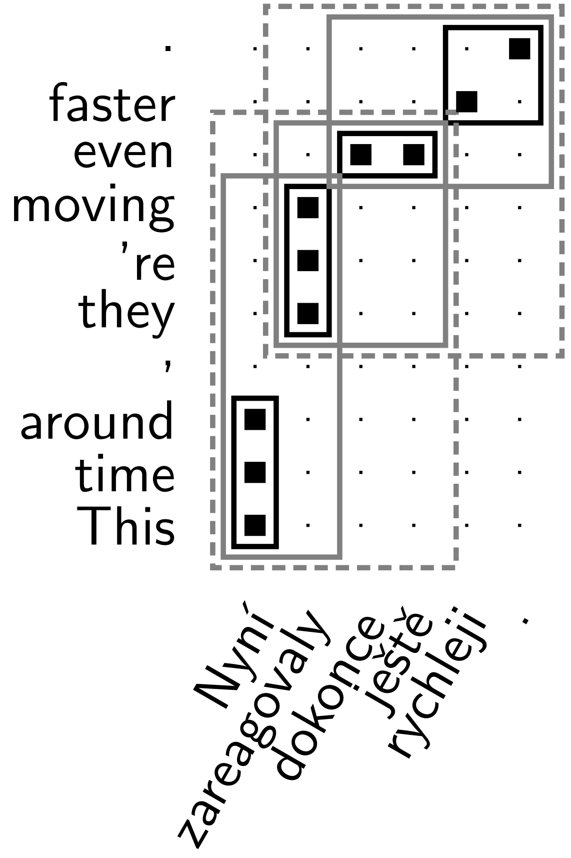
(Dusty)

- Tree-based Transfer

(Corrugated)

- Empirical Evaluation
- Error Analysis.

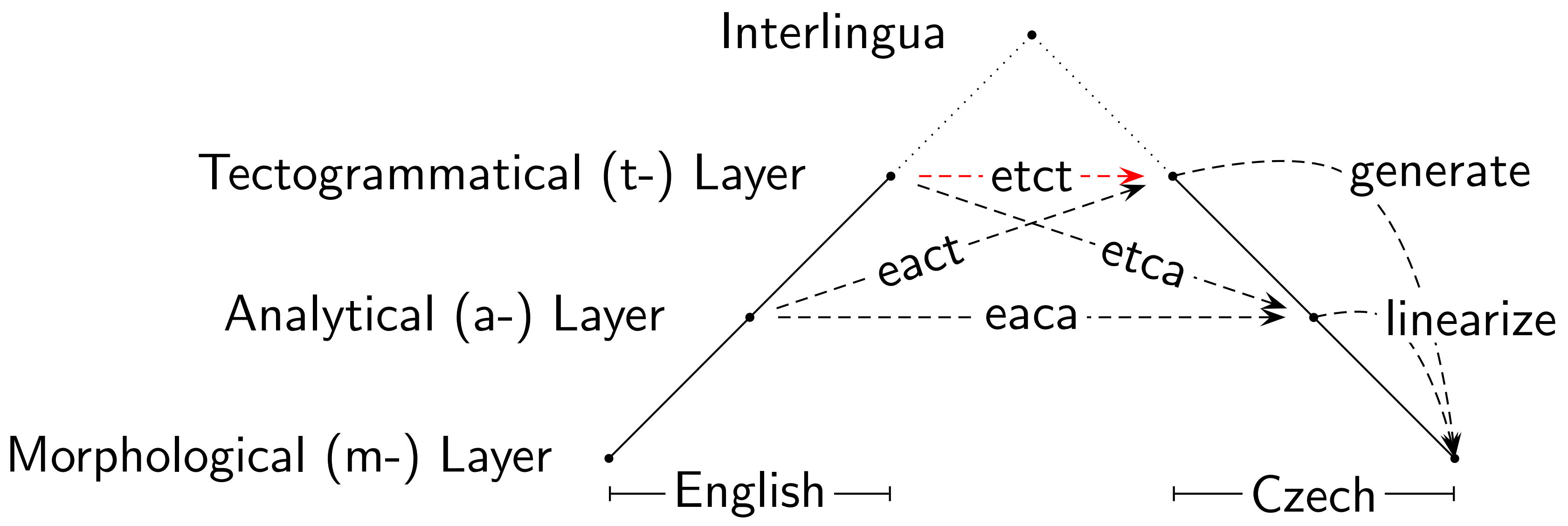
Baseline: Phrase-Based MT



This time around	=	Nyní
they 're moving	=	zareagovaly
even	=	dokonce ještě
. . .	=	. . .
This time around, they 're moving	=	Nyní zareagovaly
even faster	=	dokonce ještě rychleji
. . .	=	. . .

Phrase-based MT: choose such segmentation of input string and such phrase "replacements" to make the output sequence "coherent" (3-grams most probable).

Overview: Deep Syntatic Machine Translation



Tectogrammatics: Deep Syntax Culminating

Background: Prague Linguistic Circle (since 1926). Theory: Sgall (1967), Panevová (1980), Sgall, Hajičová, and Panevová (1986).

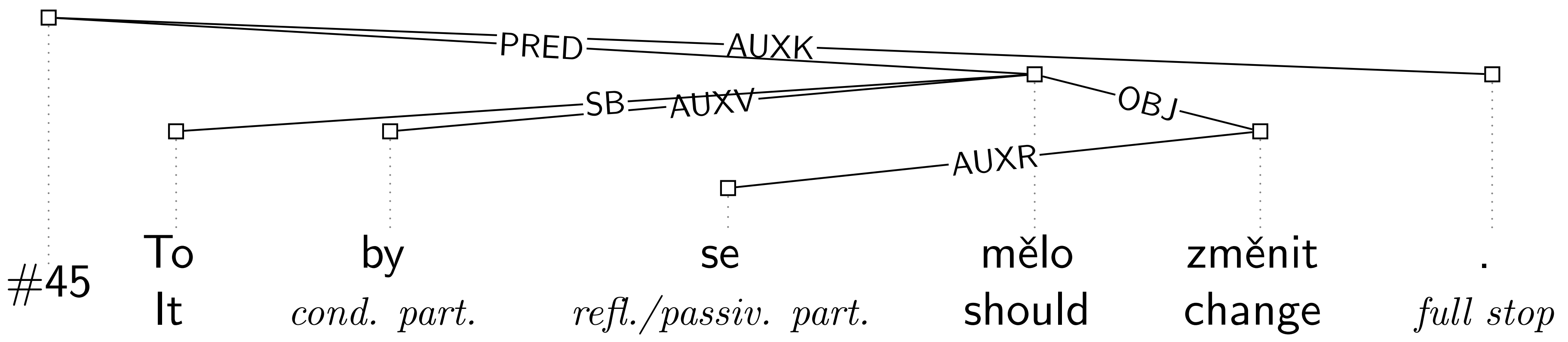
Materialized theory: Treebanks:

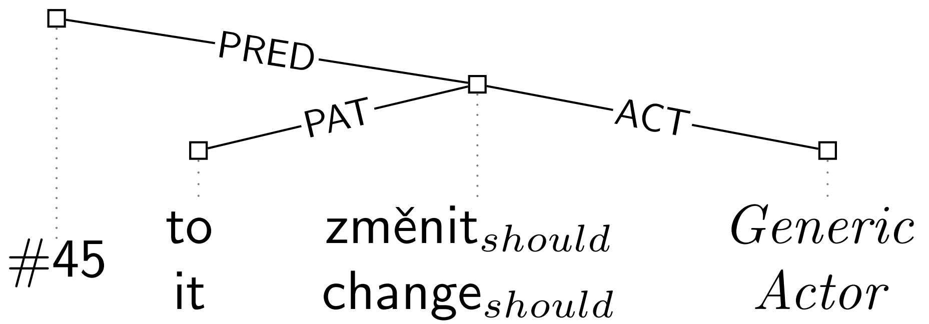
- Czech: PDT 1.0 (2001), PDT 2.0 (2006)
- Czech-English: PCEDT 1.0 (2004), PCEDT 2.0 (in progress)
- Arabic: PADT (2004)

Practice: Tools:

- parsing Czech to surface: McDonald et al. (2005)
- parsing Czech to deep: Klimeš (2006)
- parsing English to surface: well studied (+rules convert to dependency trees)
- parsing English to deep: heuristic rules (manual annotation in progress)

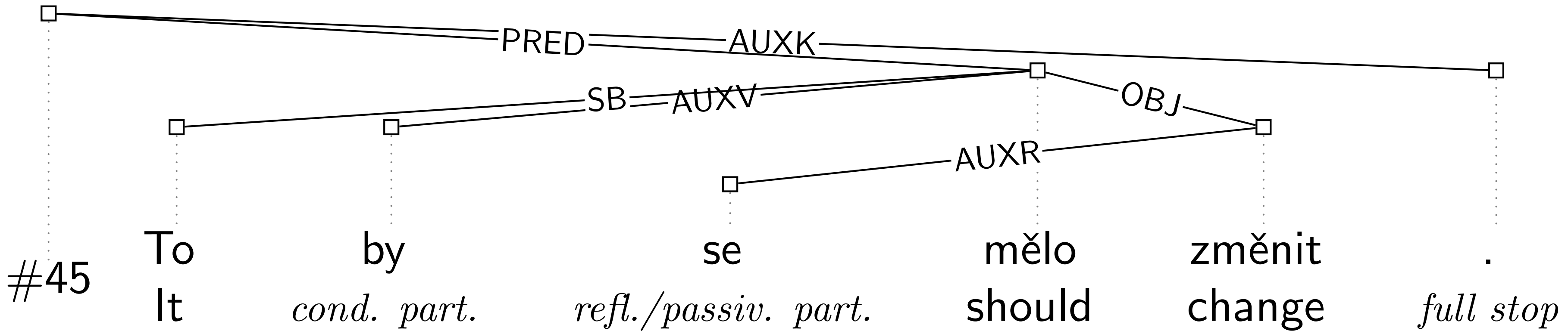
Analytic vs. Tectogrammatical

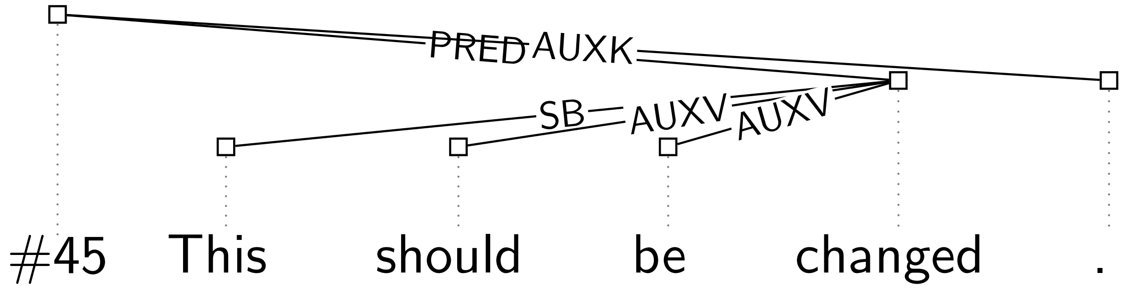




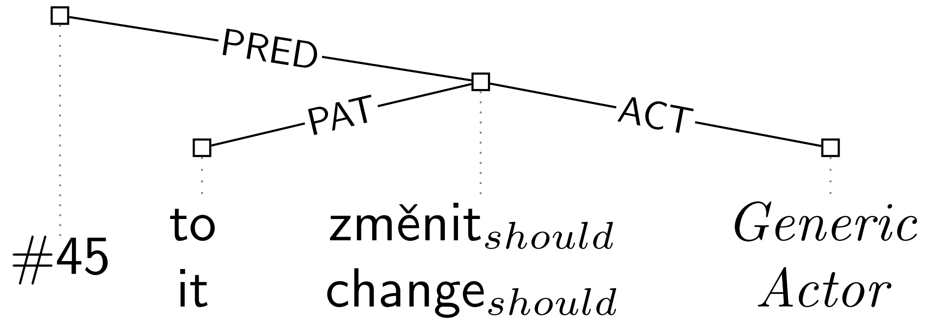
- hide auxiliary words, add nodes for "deleted" participants
- resolve e.g. active/passive voice, analytical verbs etc.
- "full" tecto resolves much more, e.g. topic-focus articulation or anaphora

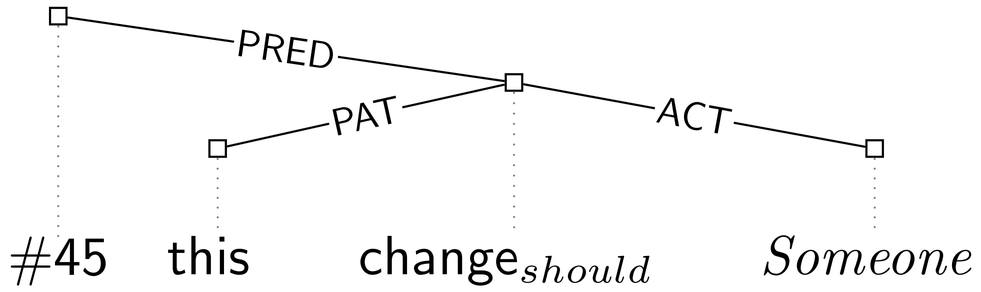
Czech and English Analytic





Czech and English Tectogrammatical





Predicate-argument structure: `change`$_{\texttt{should}}$`(ACT: someone, PAT: it)`

The Tectogrammatical Hope

Transfer at t-layer should be easier than direct translation:

be-easier$_{\text{should}}$(ACT: transfer(LOC: t-layer), CMP: translation(RSTR: direct))

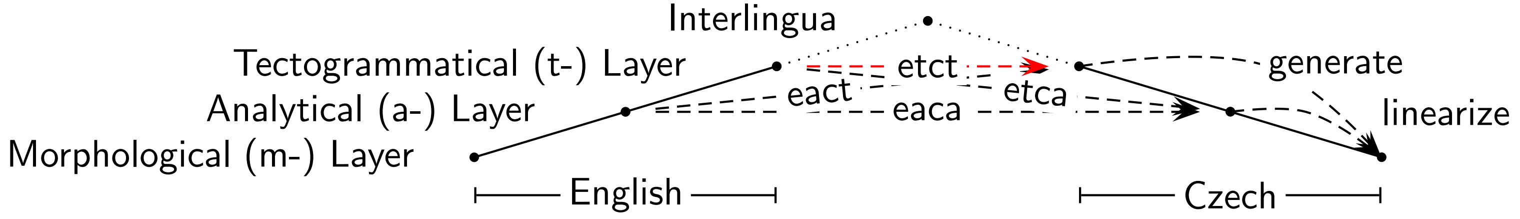
- Reduced structure size (auxiliary words disappear).
- Long-distance dependencies (non-projectivites) solved at t-layer.
- Word order ignored / interpreted as information structure (given/new).
- Reduced vocabulary size (Czech morphological complexity).
- Czech and English t-trees structurally more similar
  ⇒less parallel data might be sufficient (but more monolingual).
- Ready for fancy t-layer features: co-reference.

The complications:

- 47 pages documenting data format (PML, XML-based, sort of typed)
- 1200 pages documenting Czech t-structures
  "Not necessary" once you have a t-tree but useful understand or to blame the right people.

Tree-to-tree Transfer

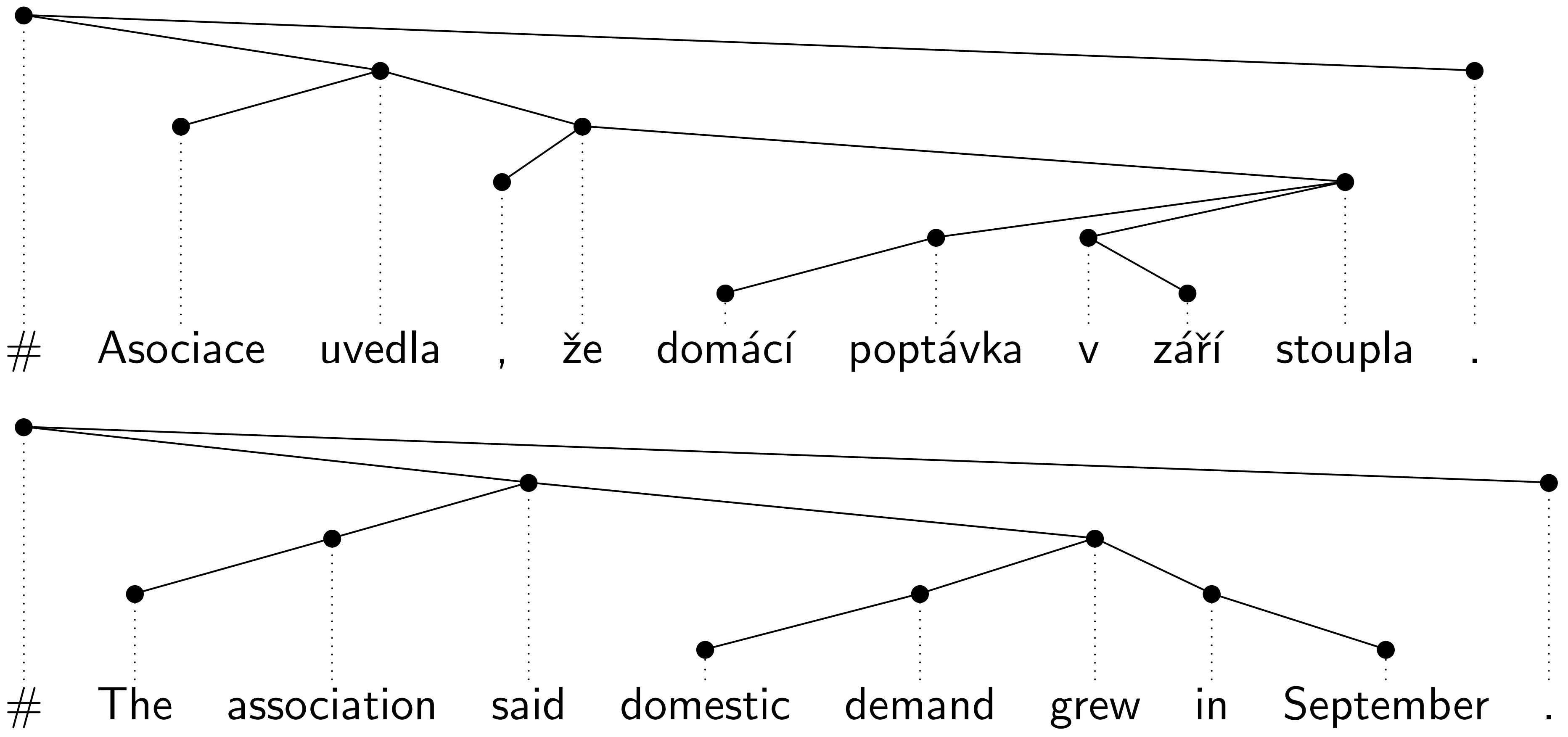
Synchronous Tree Substitution Grammar (Čmejrek, 2006).



Given an input dependency tree:

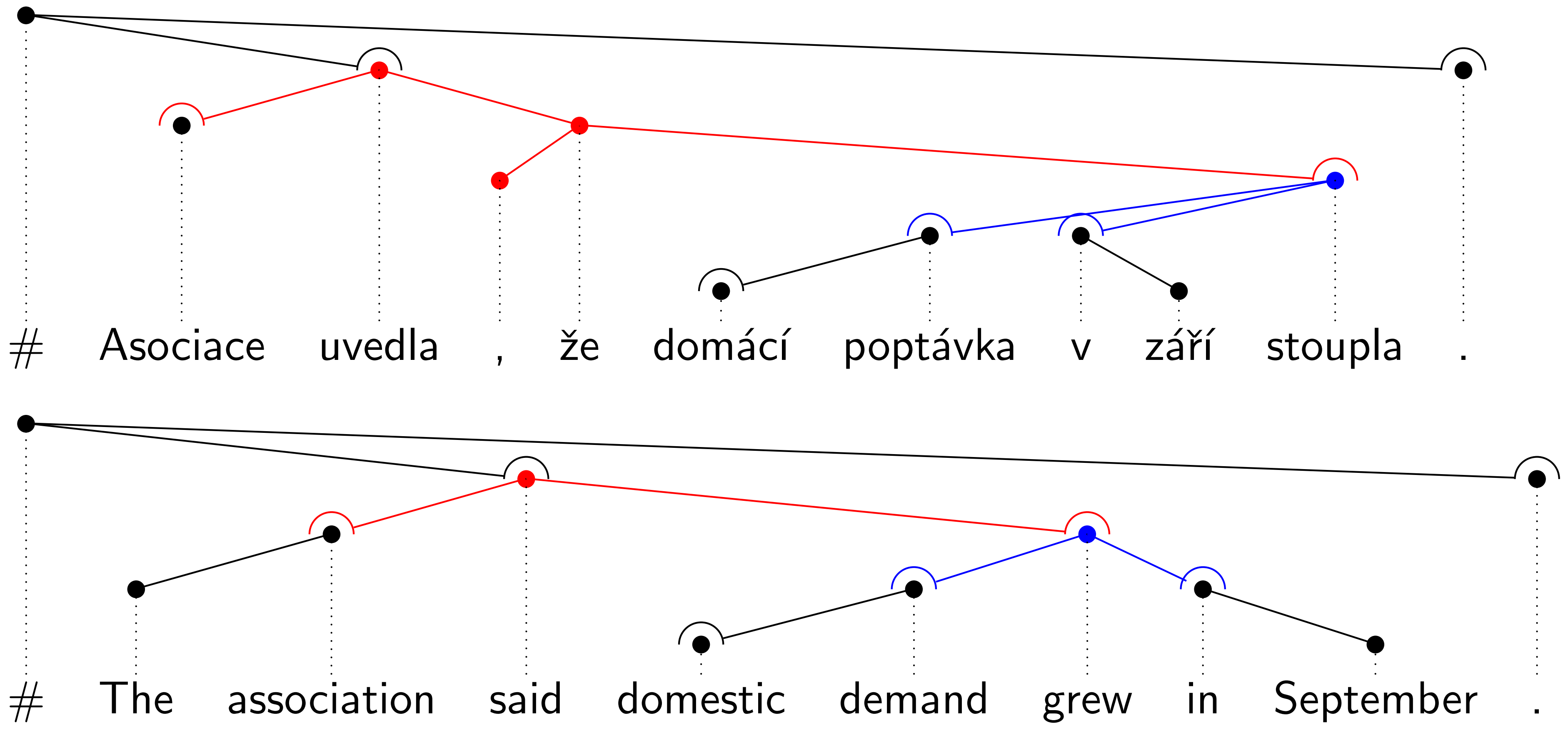
- decompose it into known treelets,
- replace treelets by their treelet translations,
- join output treelets and produce output final tree,
  - for a-tree, read off the sequence of words,
  - for t-tree, run rule-based Czech sentence generation (Ptáček, 2005) to get the sequence of words.

Idea: Observe a Pair of Dependency Trees



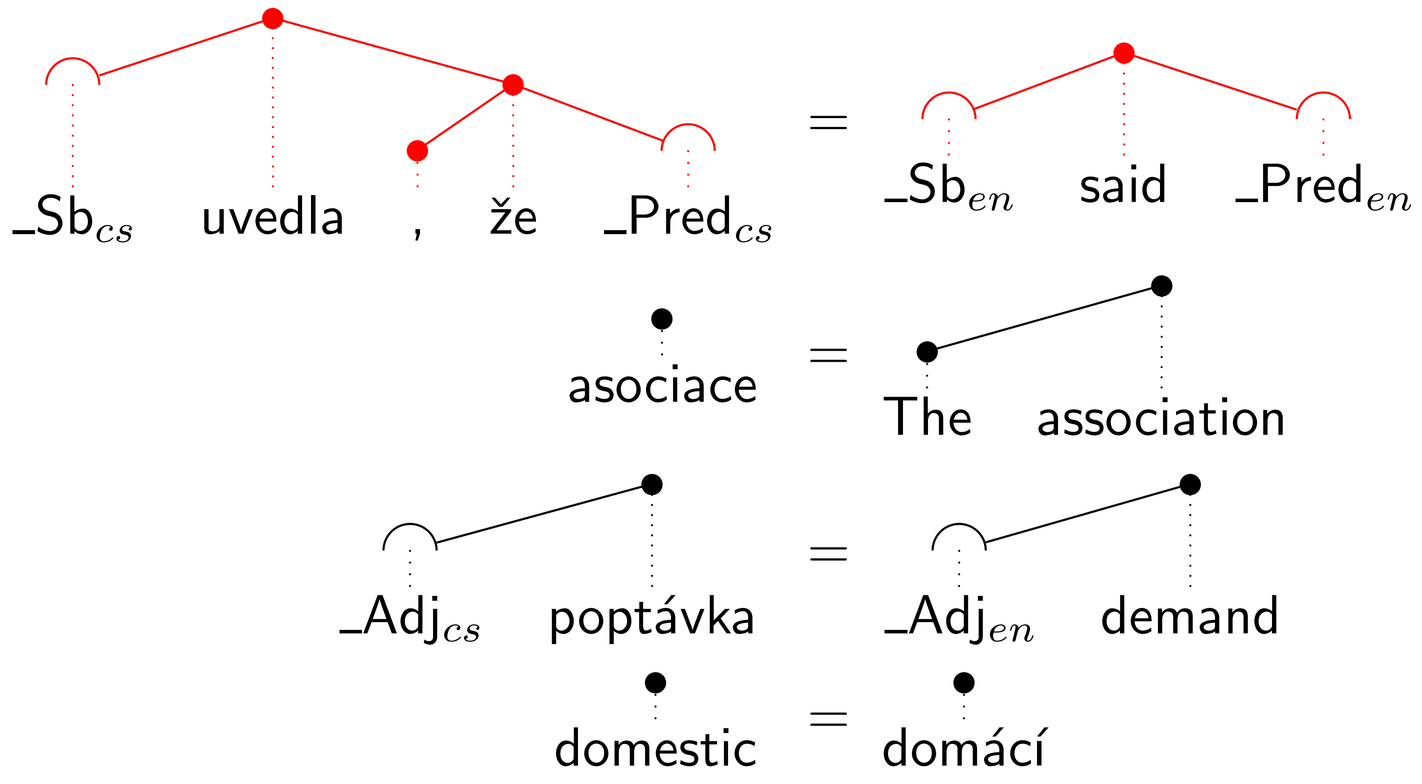
# Asociace uvedla , že domácí poptávka v září stoupla .
# The association said domestic demand grew in September .

Idea: Decompose Trees into Treelets



# Asociace uvedla , že domácí poptávka v září stoupla .
# The association said domestic demand grew in September .

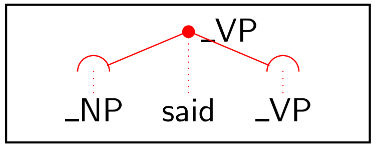
Idea: Collect Dictionary of Treelet Pairs



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 $V^f = V \setminus V^i$ 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.

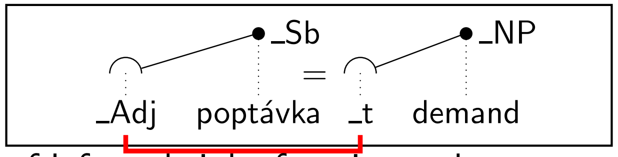
Optionally, we can keep track of local or global ordering of nodes in treelets.

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, Synchronous 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: $p(\delta) = p(t^0_{1:2} | Start_{1:2}) * \prod_{i=1}^{k} p(t^k_{1:2} | q^k_{1:2})$

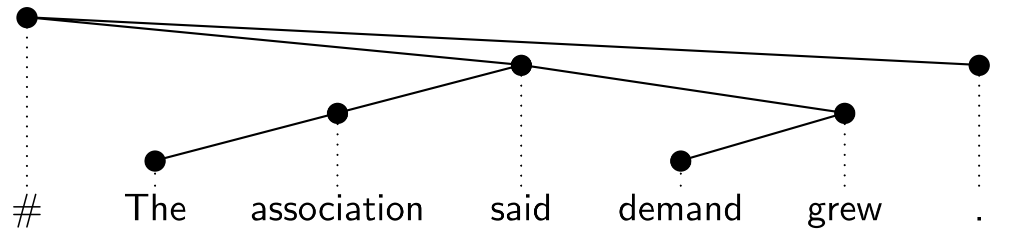
Decoding STSG

- Find target tree such that the synchronous derivation δ is most likely.
- Implemented as two-step top-down beam-search similar to Moses:

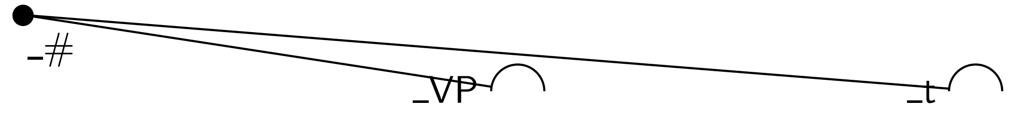
1. Prepare **translation options table**:
    - For every source node consider every subtree rooted at that node.
    - If the subtree matches the source treelet in a treelet pair, we've got a translation option.
    - Keep only best τ translation options at a node.

2. Gradually **expand partial hypotheses**:
    - Starting at root use translation options to cover source tree.
    - Keep only best σ partial hypotheses of a given size (input nodes covered).

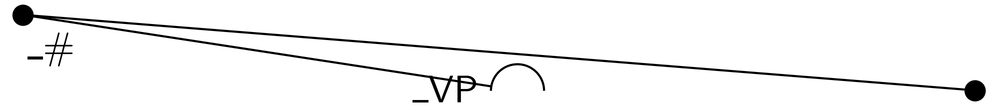
Translation Options Example



Sample translation options at root:

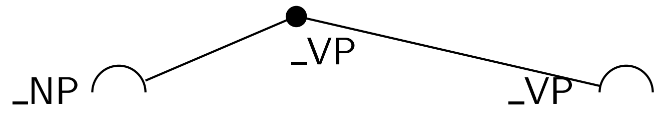


⇒ # _Pred _AuxK



⇒ # _Pred .

Sample translation options at 'said':

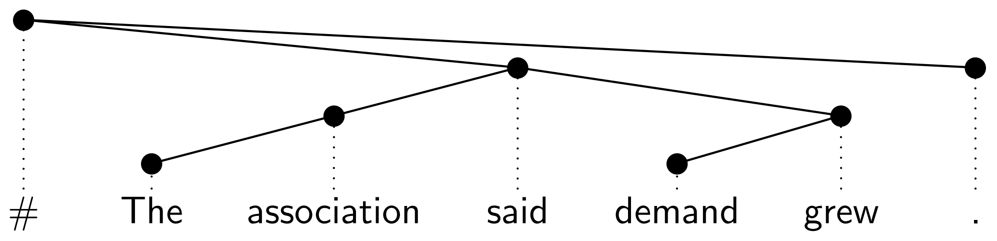


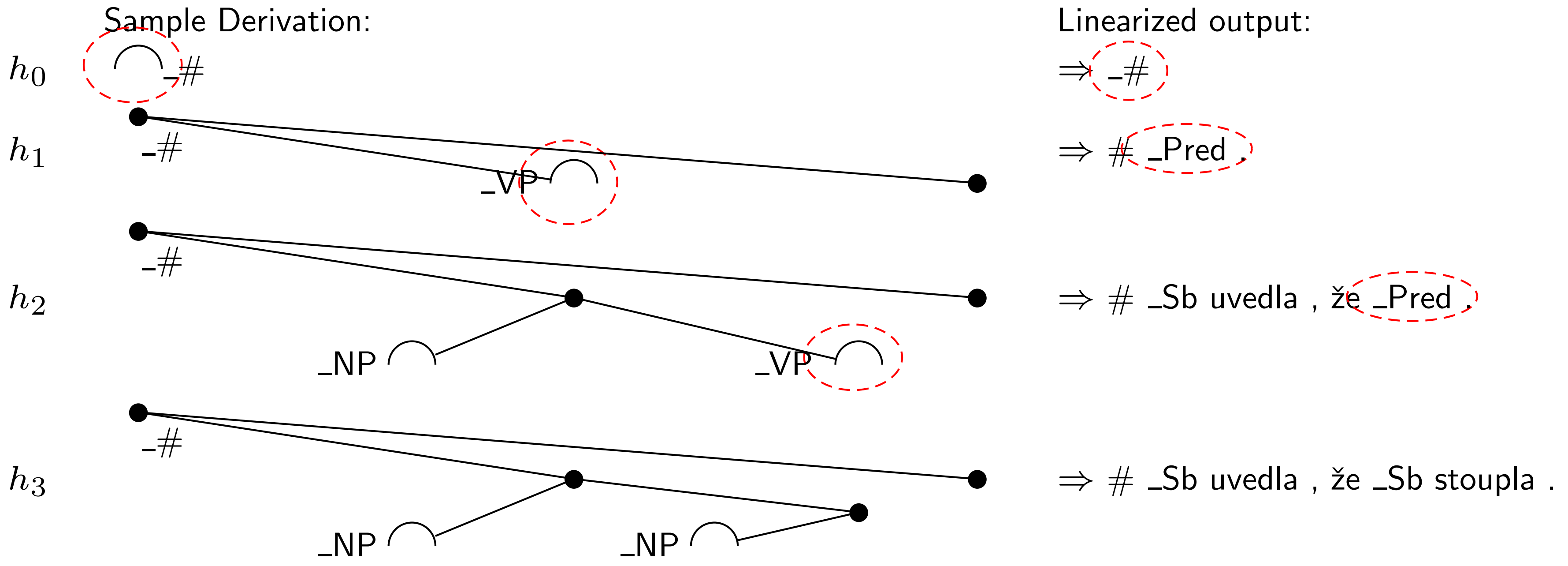
⇒ _Sb uvedla , že _Pred

Sample translation options at '.':

⇒ .

Hypothesis Expansion Example





Treelet Alignments: Heuristics

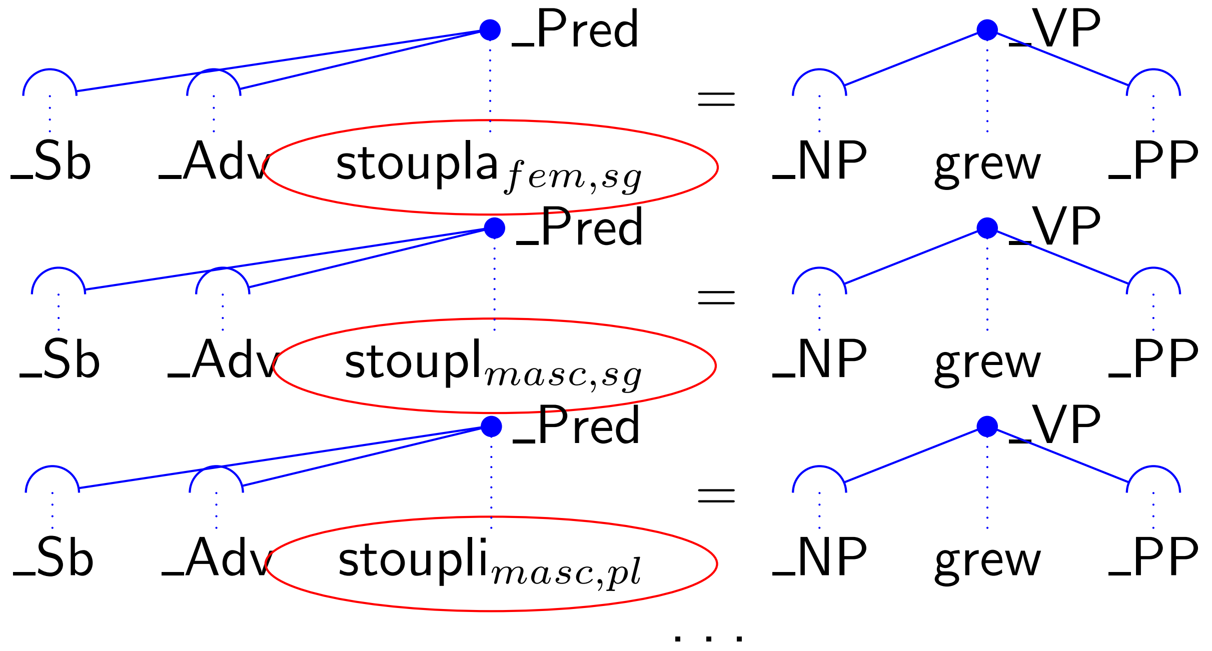
- Similar to common phrase-extraction techniques given word alignments.
- Basic units are little trees instead of word spans.

1. Obtain **node-to-node alignments** (GIZA++ on linearized trees).
2. Extract all treelet pairs satisfying these conditions:
   - no more than i internal nodes and f frontier nodes,
   - **compatible with node alignment**,
     e.g. no node-alignment link leads outside the treelet pair and frontiers are linked.
   - satisfying **STSG property**.
     All children of an internal node have to be included in the treelet (as frontiers or internals), ie. assume no adjunction operation was necessary to construct the full tree.
3. Estimate probabilities, e.g. $p(t_1, t_2 | \text{root state}_1, \text{root state}_2)$

Risks of Data Sparseness (1)

Morphological richness:

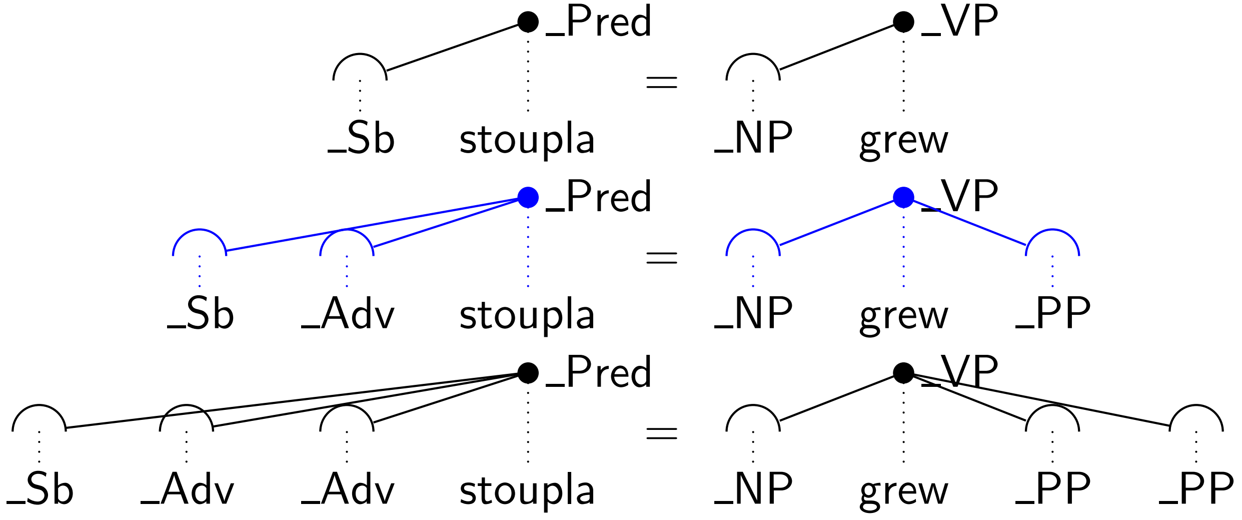
- not an issue at t-layer, where nodes hold t-lemmas.



Risks of Data Sparseness (2)

Number and states of frontiers for additional adjuncts:

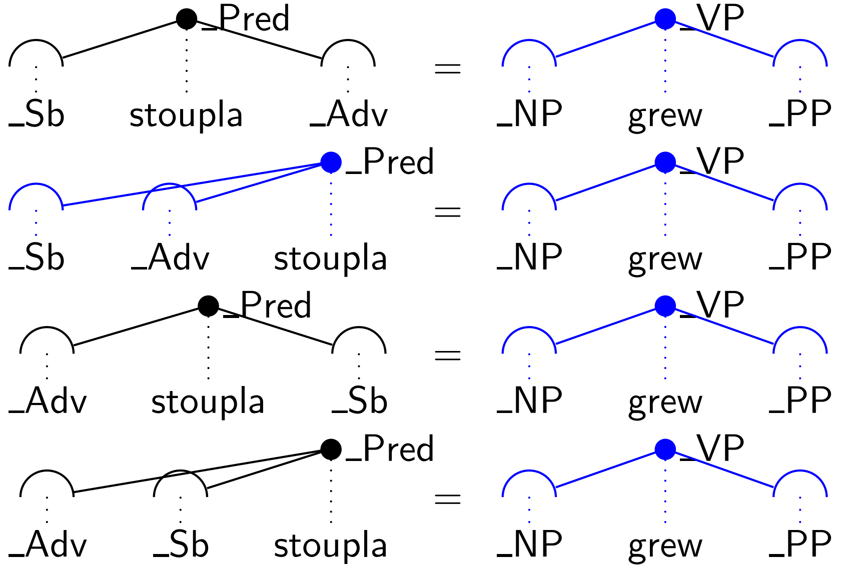
- STSG property: Once a node is used as internal, all its children have to be included in the little tree as internals or frontiers. (There is no adjunction.)



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 ⇒ 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.

Word for word. Useful for single-internal treelets only: The label of the root internal translated independently, frontiers generated on the fly, labelled probabilistically, order unchanged.

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.

Implementation Details

- STSG model extended to log-linear combination of features:

$$\text{best derivation } \hat{\delta} = \operatorname*{argmax}_{\delta \in \Delta(T_1)} \exp\Big(\sum_{m=1}^{M} \lambda_m h_m(\delta)\Big) \tag{1}$$

$$\text{instead of } \hat{\delta} = \operatorname*{argmax}_{\delta \in \Delta(T_1)} p(t^0_{1:2} | Start_{1:2}) * \prod_{i=1}^{k} p(t^k_{1:2} | q^k_{1:2}) \tag{2}$$

- Tinycdb (like GDBM) to store and access treelet dictionaries.
- Target tree structure can be disregarded (output linearized right away).
  - IrstLM to promote hypotheses containing frequent trigrams.
- Implemented in Mercury (Somogyi, Henderson, and Conway, 1995).
- Parallel computation on Sun Grid Engine cluster (160 CPUs in 40 machines).

Evaluation Metric: BLEU

BLEU ≈ ratio of 1- to 4-grams of hypothesis confirmed by a reference translation

System output (hypothesis):

n=1: For example , Fidelity ~~prepares~~ for case market plunge ads several months in advance .

n=2: For example , ~~Fidelity prepares for case market plunge ads several~~ months in advance .

Reference translations:

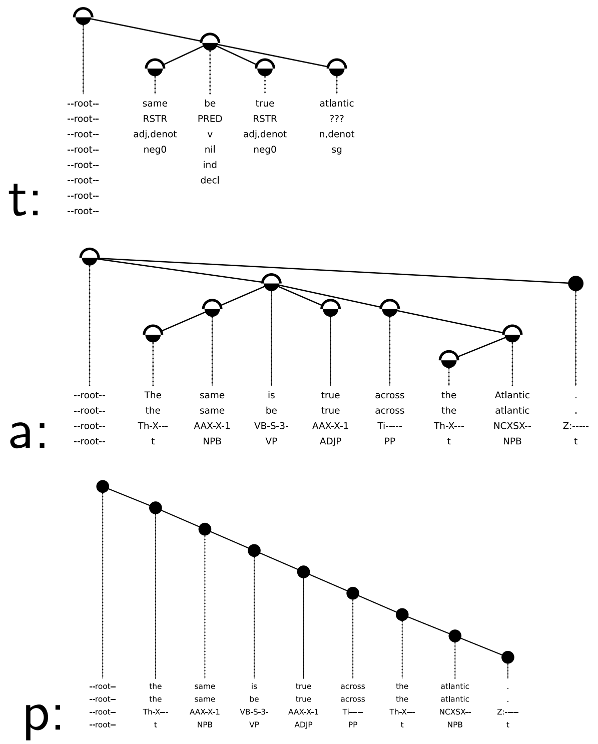
For example , Fidelity prepared advertisements for a potential market slump a few months in advance .
For example , Fidelity prepared ads some months in advance for a case where the market fell .
For instance Fidelity prepared ads for the event of a market plunge several months in advance .

- within the range 0-1, sometimes written as 0 to 100%
- humans: ˜60%, Google Chinese⟶English: ˜30%, Arabic⟶English: ˜50%.

Callison-Burch et al. (2007) show better-performing metrics, though target-language dependent.

Empirical Evaluation

	BLEU
eact	3.0±0.3
etct	5.0±0.5
etca	6.3±0.6
eaca	8.6±0.6
epcp	10.3±0.7
epcp with no state labels	11.0±0.7
Phrase-based (Moses)	
Vanilla	12.9±0.6
Factored	14.2±0.7


t:
--root-- same be true atlantic
RSTR PRED RSTR ???
adj.denot v adj.denot n.denot
neg0 nil neg0 sg
ind
decl
a:
--root-- The same is true across the Atlantic .
the same be true across the atlantic .
Th-X--- AAX-X-1 VB-S-3- AAX-X-1 Ti----- Th-X--- NCXSX-- Z:-----
t NPB VP ADJP PP t NPB t
p:
--root-- the same is true across the atlantic .
the same be true across the atlantic .
Th-X--- AAX-X-1 VB-S-3- AAX-X-1 Ti----- Th-X--- NCXSX-- Z:-----
t NPB VP ADJP PP t NPB t

ACL 2007 WMT shared task data, 55k training sentences, 964 test sentences.

Observations and Explanations

Why target t-layer performs worst?

- errors accumulate (noisy input parse, noisy transfer, noisy generation),
- increased data sparseness (we'll discuss this more),
- rule-based generation does not make use of n-gram language model,
  BLEU disfavours methods without language models.
- lack of tree-based language model.

Why a-transfer performs worse than p-transfer (no tree structure)?

- Incompatibilities of the parses and alignment prevent treelet extraction.

Why is Moses better than p-transfer?

- We don't allow any reordering of phrases.
- Moses has proper minimum-error rate training, we try a few settings.

One Source of Data Sparseness: Vocabulary Size

	BLEU	Sents	Vocabulary source	Vocabulary target	States source	States target
eact	3.02±0.34	56k	64k	128k	29	67
etct	5.02±0.47	57k	81k	133k	28	67
etca	6.28±0.57	56k	80k	85k	28	451
eaca	8.59±0.60	56k	60k	83k	29	450
epcp	10.34±0.66	56k	60k	83k	29	450
epcp with no state labels	11.03±0.68	56k	60k	83k	2	2

Why is t-layer vocabulary so much bigger:

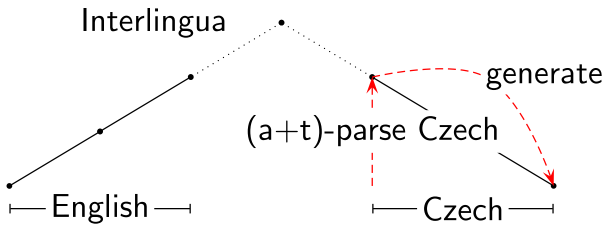
- t-nodes have about 25 attributes: t-lemma, functor, gender, person, tense, iterativeness, dispositional modality, . . .

What do we use as STSG states: (i.e. the thing checked when attaching treelets)

- t-layer: functors (ACT, PAT, PRED, . . .)
- a-layer: agreement info (N-accusative-prep'do', A-genitive, . . .)
- p-layer: agreement info or nothing

Upper Bound on MT Quality via t-layer

- Analyse Czech sentences to t-layer.
- Optionally ignore some node attributes.
- Generate Czech surface.
- Evaluate BLEU against input Czech sentences.

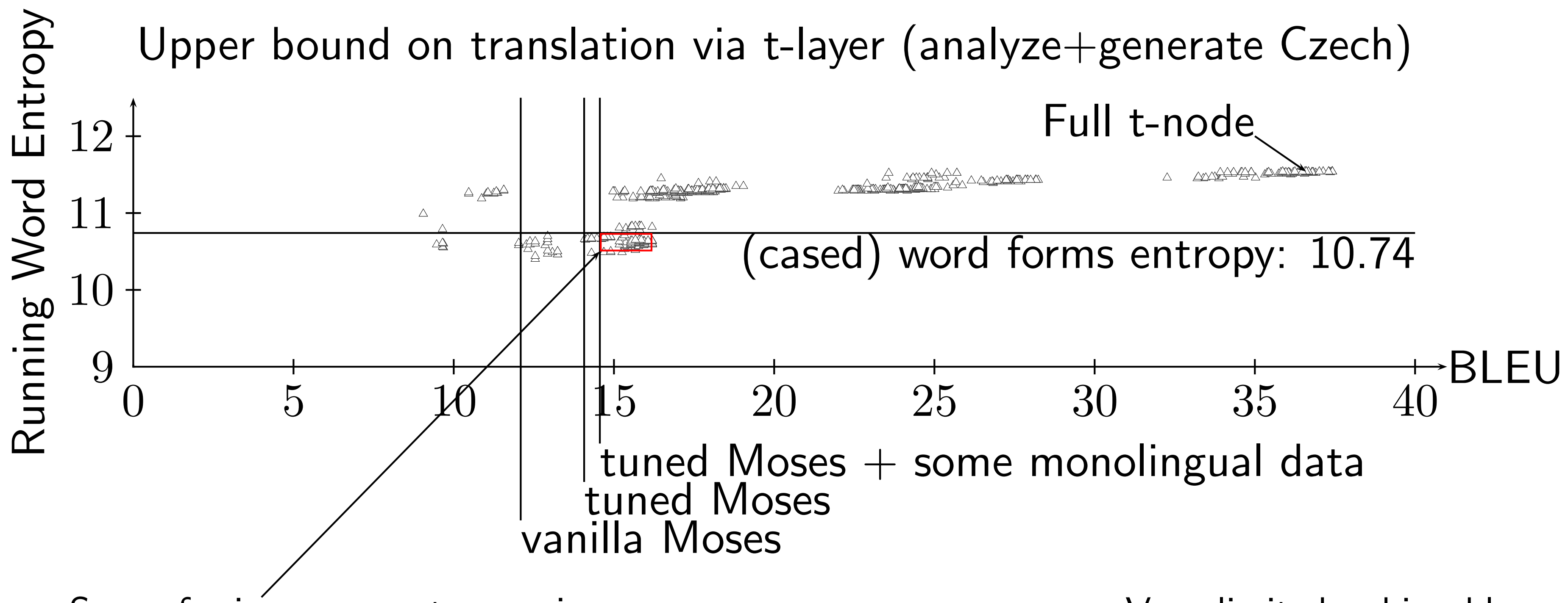


	BLEU
Full automatic t-layer, no attributes ignored	36.6±1.2
Ignore sentence mood (assume indicative)	36.6±1.2
Ignore verbal fine-grained info (resultativeness, . . .)	36.6±1.2
Ignore verbal tense, aspect, . . .	24.9±1.1
Ignore all grammatemes	5.3±0.5

⇒ Node attributes obviously very important.

⇒ Can we find a balance of small vocabulary and high achievable BLEU?

No More Fairy Tales on Vocabulary Reduction



Space for improvement assuming:

- t-nodes atomic (with a restricted set of attributes)
- we wish to stay below the entropy of plain text

⇒ Very limited achievable BLEU even if tranfer were absolutely perfect.

Consequences

- t-layer by itself increases complexity of node label choice.

⇒ cannot treat output nodes labels as atomic: `go.V.past.third.sg...`

The bare minimum:

- two factors to translate lexical item and grammatical features separately
- possibly check for internal output compatibility (more monolingual data)

English		Czech
t-lemma	$\longrightarrow$	t-lemma
other attributes	$\longrightarrow$	other attributes

Conflicting with the key concept of STSG: treelet shape (and size) alternations.

⇒ Open question: factored treelet translation.

Summary

- Theory, data and tools for deep syntactic analysis ready.
- STSG-like decoder implemented and evaluated in various settings.
- t-transfer without factoring has no chance for improvement.

References

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 136–158, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Klimeš, Václav. 2006. Analytical and Tectogrammatical Analysis of a Natural Language. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In Proceedings of HLT/EMNLP 2005, October.

Panevová, Jarmila. 1980. Formy a funkce ve stavbě české věty [Forms and functions in the structure of the Czech sentence]. Academia, Prague, Czech Republic.

Ptáček, Jan. 2005. Generování vět z tektogramatických stromů Pražského závislostního korpusu. Master's thesis, MFF, Charles University, Prague.

Sgall, Petr. 1967. Generativní popis jazyka a česká deklinace. Academia, Prague, Czech Republic.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. 1986. The Meaning of the Sentence and Its Semantic and Pragmatic Aspects. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands.

Somogyi, Zoltan, Fergus Henderson, and Thomas Conway. 1995. Mercury: an efficient purely declarative logic programming language. In Proceedings of the Australian Computer Science Conference, pages 499–512, Glenelg, Australia, February.