

Syntax in Pre-Neural Statistical MT

Ondřej Bojar

📅 April 9, 2020



EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research,
Development and Education

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics



unless otherwise stated

Outline

- Motivation for grammar in MT.
- Hierarchical Model.
- Proper syntax: Constituency vs. dependency trees.

Constituency Syntax:

- Context Free Grammars.
- MT as parsing.
 - Synchronous CFG, LM integration.
- Why real source/target parse trees make it harder.

Dependency Syntax:

- Surface syntax (STSG), problems.
- Deep syntax (t-layer); TectoMT, HMTM.

Motivation

Simple phrase-based models:

- Don't check for grammatical coherence.
⇒ 3-grams fluent, overall word salad.
- Don't allow gaps in phrases:
I do not know... \leftrightarrow Je ne sais pas...
“do not” \leftrightarrow “ne pas”
- Reordering models have little explicit knowledge:
 - No movement of longer blocks.
 - No grammar constraints possible.

Hierarchical Model

Hierarchical Phrase-Based Model

Hierarchical model (Chiang, 2005): rough approximation of syntax.

Hiero addresses only the gaps in phrases:

- Gaps don't have labels \Rightarrow any phrase fits.

“do not X” \leftrightarrow “ne X pas”

“do not cat” \leftrightarrow “ne chat pas”

\Rightarrow Not really a grammar, but not an issue for correct input.

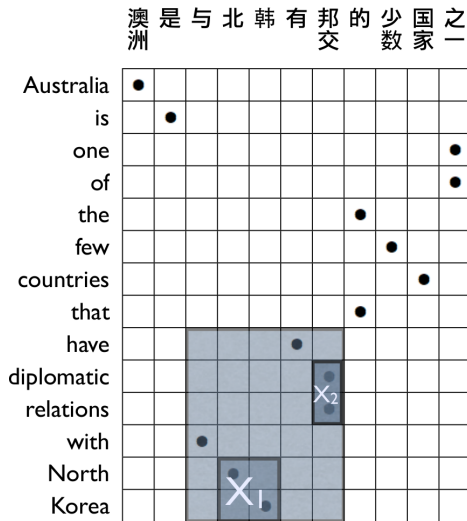
- Phrase extraction similar to phrase-based:

1. Extract all (non-gappy) phrases consistent with alignment.

2. If a subphrase is also extracted, make a synchronous gap.

\Rightarrow Much higher number of rules extracted.

Hierarchical Phrase Extraction

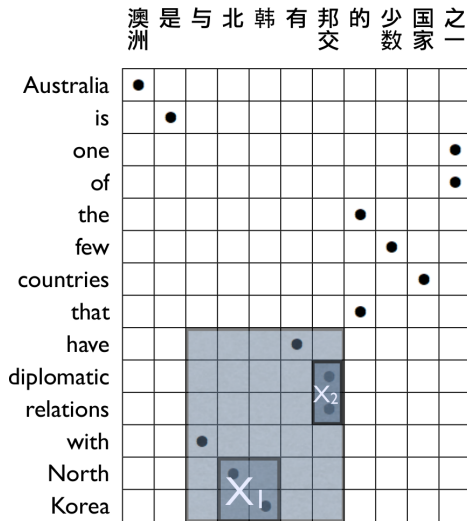


(与北韩有邦交,
have diplomatic
relations with
North Korea)

(邦交, diplomatic
relations)

(北韩, North Korea)

Hierarchical Phrase Extraction



(与北韩有邦交,
have diplomatic
relations with
North Korea)

(邦交, diplomatic
relations)

(北韩, North Korea)

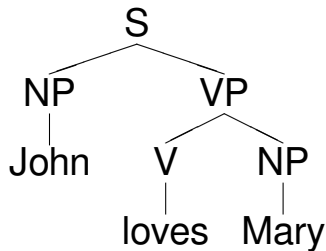
$X \rightarrow$ 与 X_1 有 X_2 ,
have X_2 with X_1

Challenges of Hierarchical Model

- Very high number of extractable rules.
 - Limited by ad-hoc constraints:
 - Initial phrases ≤ 10 words.
 - Rules ≤ 6 symbols.
 - At least one aligned terminal required.
 - At most two non-terminals, non-adjacent.
- Spurious ambiguity.
 - = many ways to obtain the same output.
 - Pollutes n-best lists.
- LM is a non-local feature.
 - Causes serious state splitting \Rightarrow Search space much larger.
- ... Plus hierarchical model is not a syntactic model.
 - With a special treatment of unaligned words, a regular PBMT system can reach most of hierarchical hypotheses (Auli et al., 2009).

Proper Syntax

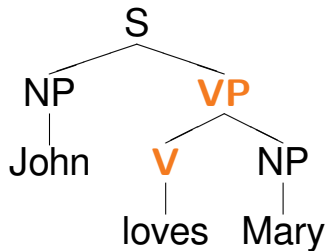
Constituency vs. Dependency Trees



Constituency trees = syntactic structure of a sentence as “bracketting”:

$$(_S (_N^P \text{ John}) (_V^P (_V \text{ loves}) (_N^P \text{ Mary})))$$

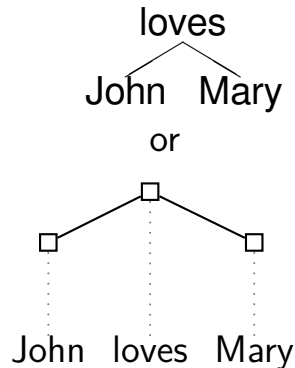
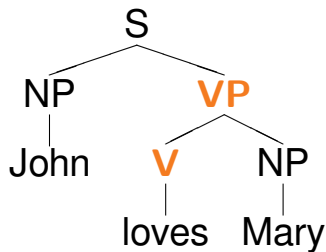
Constituency vs. Dependency Trees



Constituency trees = syntactic structure of a sentence as “bracketting”:

$$(_S (_N^P \text{ John}) (_V^P (_V \text{ loves}) (_N^P \text{ Mary})))$$

Constituency vs. Dependency Trees



Constituency trees = syntactic structure of a sentence as “bracketting”:

$$(_S (_N^P \text{ John}) (_V^P (_V \text{ loves}) (_N^P \text{ Mary}))))$$

Constituency Syntax

See MT Talk #10:

http://youtu.be/y_9SEdG1u3U

Context Free Grammar

CONTEXT-FREE GRAMMAR (CFG) describes infinite set of valid trees using a finite set of rules, e.g.:

$$S \rightarrow NP VP$$

PROBABILISTIC CFG assign weights to rules, e.g.:

$$VP \rightarrow \begin{cases} V & 0.1 \\ V NP & 0.5 \\ V NP NP & 0.4 \end{cases} \quad (1)$$

Generative model for probabilistic CFG:

$$p(\text{tree } T | \text{sentence } s) = \prod_{X \rightarrow \alpha \in T} p(\alpha | X) \quad (2)$$

Parsing

PARSING is finding the most probable constituency tree:

$$\hat{T} = \underset{T \in \{\text{trees of sentence } s\}}{\operatorname{argmax}} p(T|s) \quad (3)$$


CKY (CYK) algorithm for $O(n^3)$ parsing. (“dynamic programming”):

length: 7

7	S						
6		VP					
5							
4	S						
3		VP			PP		
2	S		NP			NP	
1	NP	V, VP	Det	N	P	Det	N
	₀ she ₁	₁ eats ₂	₂ a ₃	₃ fish ₄	₄ with ₅	₆ a ₆	₆ fork ₇

Synchronous CFG

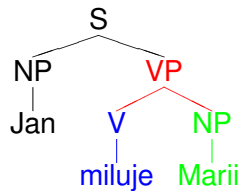
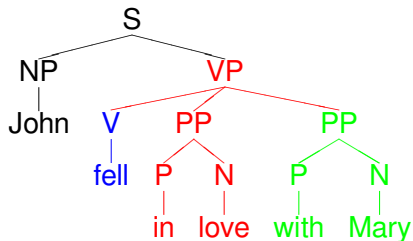
- SYNCHRONOUS CFG capture the “double generation”.
 - Nonterminals must map 1-1.

$X \rightarrow X_0$ 的 X_1  X_1 in X_0

Synchronous CFG

- SYNCHRONOUS CFG capture the “double generation”.
 - Nonterminals must map 1-1.

$X \rightarrow X_0$ 的 X_1 **I** X_1 in X_0

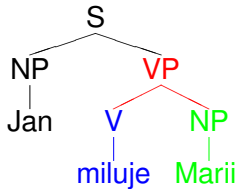
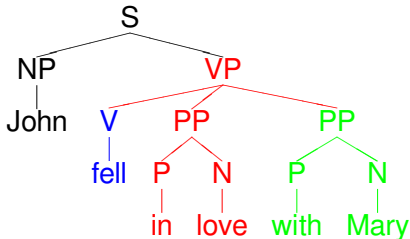


Synchronous CFG

- SYNCHRONOUS CFG capture the “double generation”.
 - Nonterminals must map 1-1.

$X \rightarrow X_0$ 的 X_1 **I** X_1 in X_0

- SYNCHRONOUS TREE SUBSTITUTION GRAMMARS (STSG)
 - Whole subtrees are attached \Rightarrow structural changes ok.

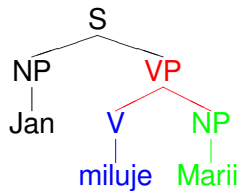
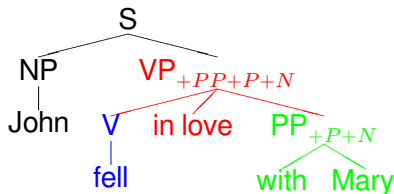
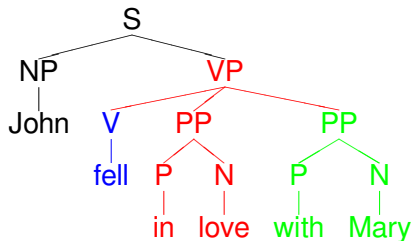


Synchronous CFG

- SYNCHRONOUS CFG capture the “double generation”.
 - Nonterminals must map 1-1.

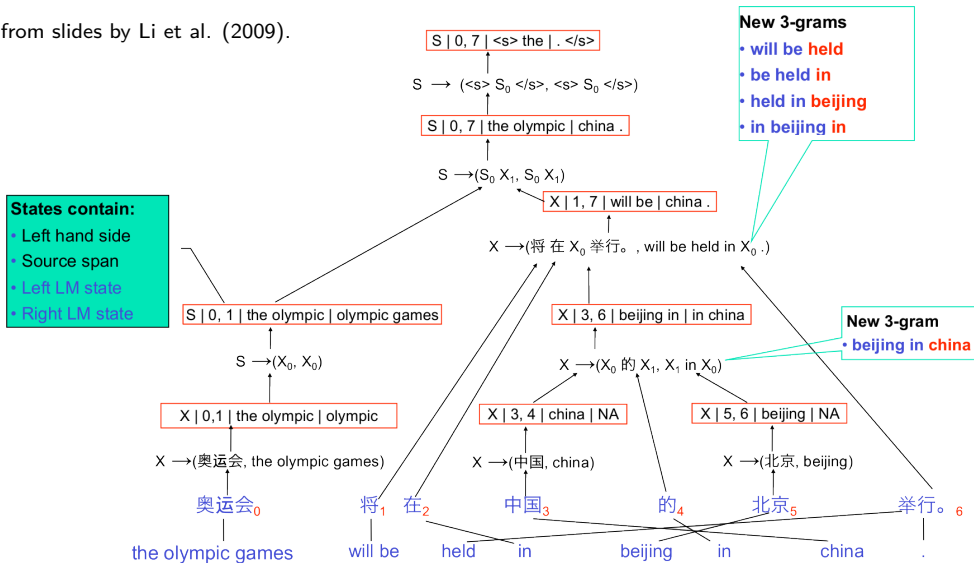
$$X \rightarrow X_0 \text{ 的 } X_1 \text{ } X_1 \text{ in } X_0$$

- SYNCHRONOUS TREE SUBSTITUTION GRAMMARS (STSG)
 - Whole subtrees are attached \Rightarrow structural changes ok.
 - In fact equivalent to SCFG.

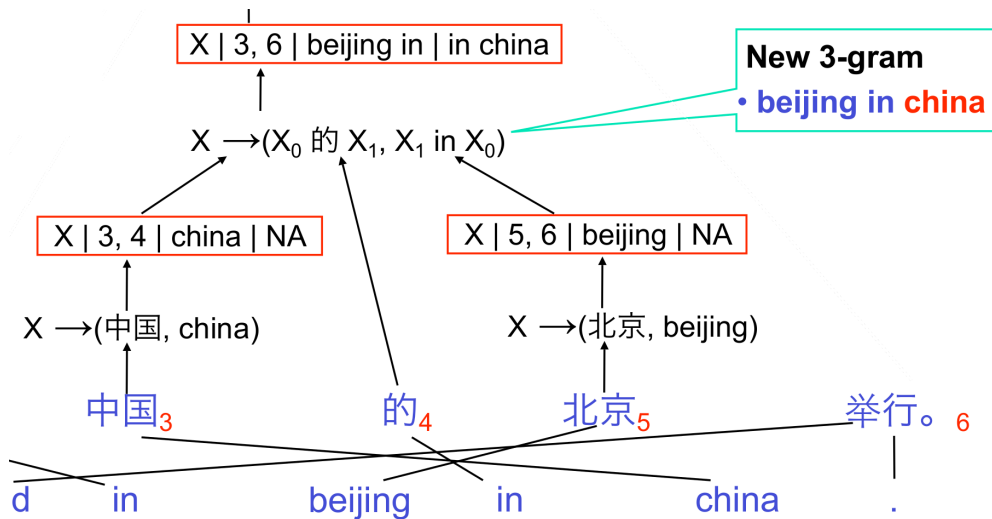


MT as Parsing: While Parsing, Translate

Picture from slides by Li et al. (2009).



State Splitting for LM



Proper Syntax is Hard

See slides by Chiang (2010):

- The source and target trees constrain too much.
 - ⇒ Too few rules extracted.
 - ⇒ Coverage lost.
- Labelled non-terminals do not always match.
 - ⇒ Some valid translations not admissible.

Relaxation of the hard constraints needed:

- Allow breaking trees into smaller fragments (e.g. binarization).
- Allow attachment of non-matching non-terminals.

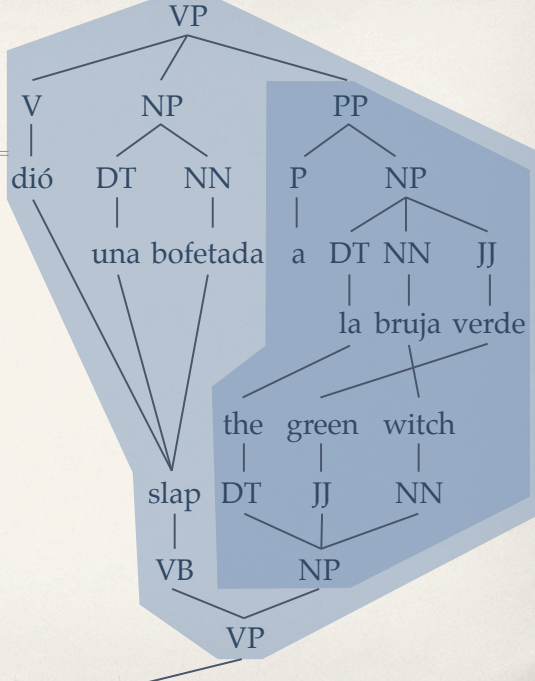
STSG extraction

1. Phrases

- * respect word alignments
- * are syntactic constituents on *both* sides

2. Phrase pairs form rules

3. Subtract phrases to form rules



STSG

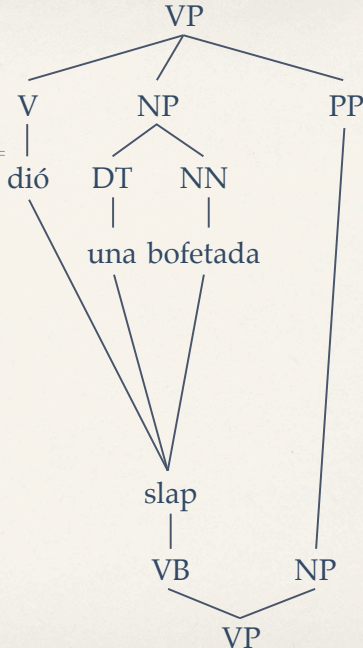
extraction

1. Phrases

- * respect word alignments
- * are syntactic constituents on *both* sides

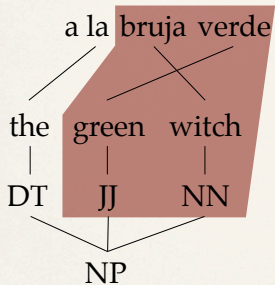
2. Phrase pairs form rules

3. Subtract phrases to form rules

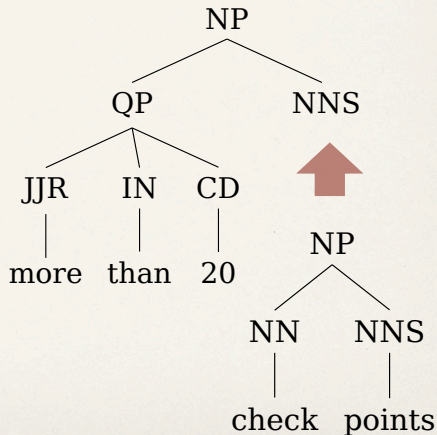


Why is tree-to-tree hard?

too few rules

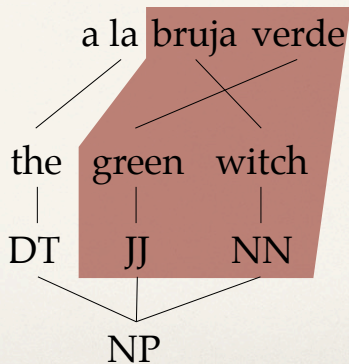


too few derivations

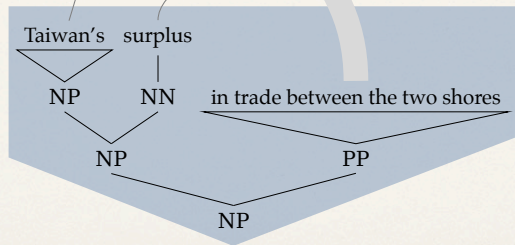
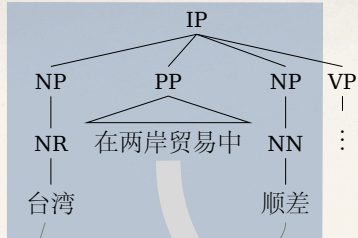


Why is tree-to-tree hard?

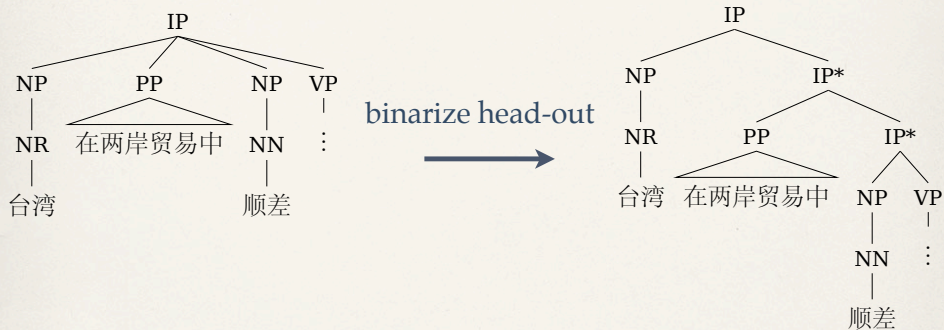
too few rules



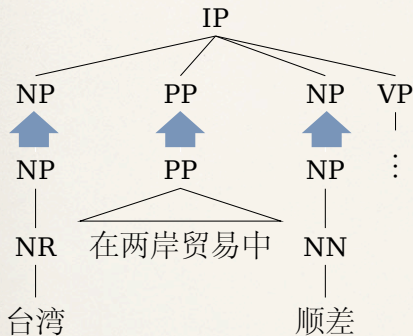




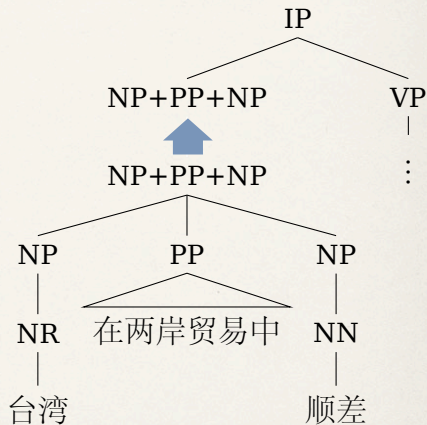
Extracting more rules



Extracting more rules



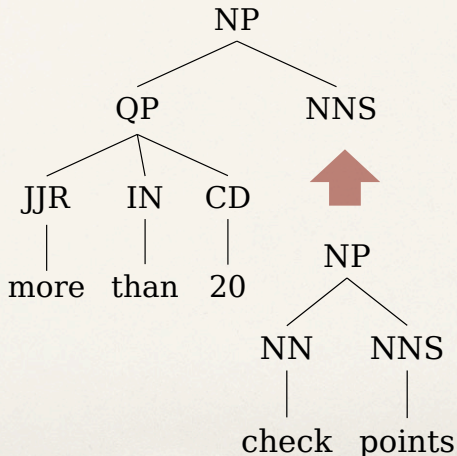
Tree-Sequence Substitution Grammar
(M. Zhang et al., 2008)



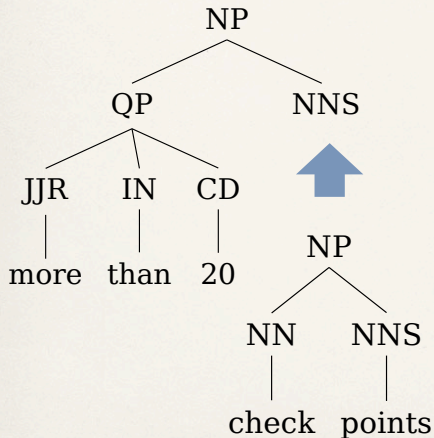
Syntax-Augmented Machine
Translation (Venugopal & Zollmann)

Why is tree-to-tree hard?

too few derivations



Allow more derivations



- ✦ STSG: allow only matching substitutions
- ✦ Hiero-like: allow any substitutions
- ✦ Let the model learn to choose:
 - ✦ matching substitutions
 - ✦ mismatching substitutions
 - ✦ monotone phrase-based

Summary of Constituency Syntax in MT

- CFG capture syntax of some natural languages.
- Translating while chart parsing.
 - SCFG/STSG parse input and construct syntactically-parallel output.
- Hierarchical model: a simplification.
 - Only a single nonterminal used.
- LM integrated by state splitting.
- When real source and/or target parse trees are used:
 - Tricks/binarization necessary to extract non-isomorphic trees.
 - Fuzzy matching must be allowed during decoding.

Dependency Syntax in MT

See MT Talk #11:

<http://youtu.be/xauhVtfXbDQ>

Constituency vs. Dependency Again

Constituency trees (CFG) represent **only bracketing**:

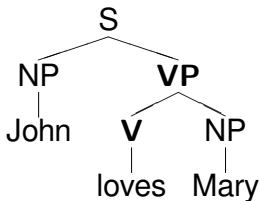
= which **adjacent** constituents are glued tighter to each other.

Dependency trees represent which words depend on which.

+ usually, some agreement/conditioning happens along the edge.

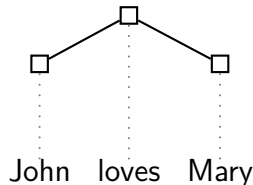
Constituency

John (loves Mary)
John _{VP}(loves Mary)



Dependency

loves
John Mary



What Dependency Trees Tell Us

Input: The **grass** around your house should be **cut** soon.

Google: **Trávu** kolem vašeho domu by se měl **snížit** brzy.

Long-distance between *grass* and *cut*:

- Can “pump” many words in between \Rightarrow phrase limit exceeded.

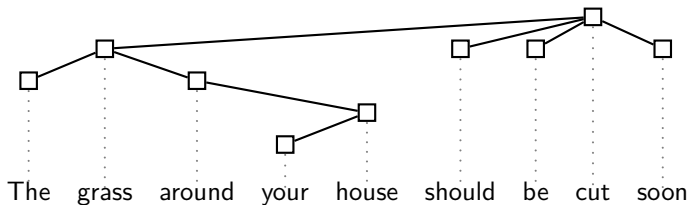
Two errors caused by independent translation of *grass* and *cut*:

- Bad lexical choice for *cut* = *sekat*/*snížit*/*krájet*/*řezat*/...
- Bad case of *tráva*.
 - Depends on the chosen active/passive form:

active \Rightarrow accusative	passive \Rightarrow nominative
trávu ... by ste se měl posekat	tráva ... by se měla posekat
	tráva ... by měla být posekána

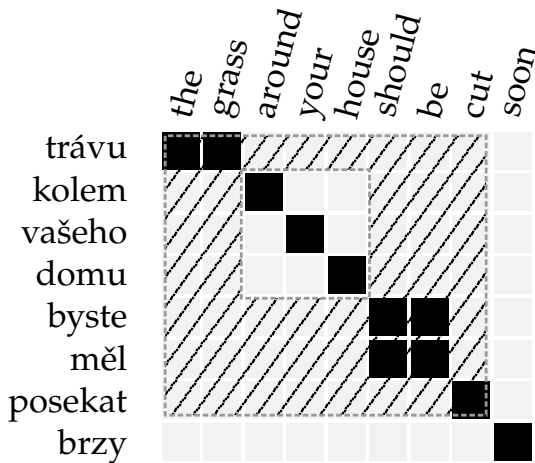
Examples by Zdeněk Žabokrtský, Karel Oliva and others.

Tree vs. Linear Context



- Tree context (neighbours in the dependency tree):
 - is better at predicting lexical choice than n -grams.
 - often equals linear context:
Czech manual trees: 50% of edges link neighbours,
80% of edges fit in a 4-gram.
- Phrase-based MT is a very good approximation.

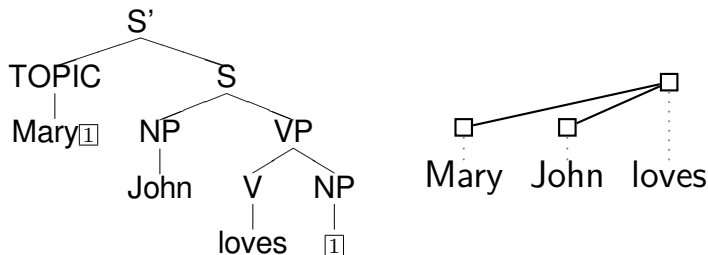
Hiero Can Cover Long-Distance Dependency



the grass X_1 should be cut = trávu X_1 byste měl posekat

“Crossing Brackets”

- Constituent outside its father’s span causes “crossing brackets.”
 - Linguists use “traces” ($\boxed{1}$) to represent this.
- Sometimes, this is not visible in the dependency tree:
 - There is no “history of bracketing”.
 - See Holan et al. (1998) for dependency trees including derivation history.

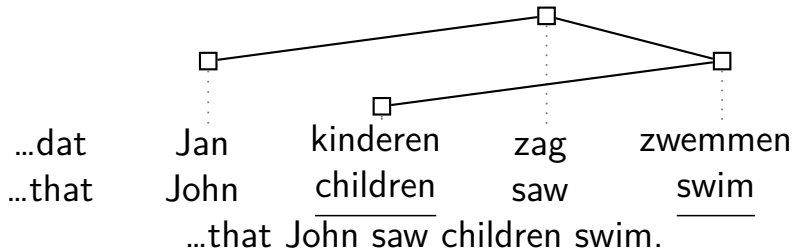


Despite this shortcoming, CFGs are popular and “the” formal grammar for many. Possibly due to the charm of the father of linguistics, or due to the abundance of dependency formalisms with no clear winner (Nivre, 2005).

Non-Projectivity

= a gap in a subtree span, filled by a node higher in the tree.

Ex. Dutch “cross-serial” dependencies, a non-projective tree with one gap caused by *saw* within the span of *swim*.



- 0 gaps \Rightarrow projective tree \Rightarrow can be represented in a CFG.
- ≤ 1 gap & “well-nested” \Rightarrow mildly context sensitive (TAG).

See Kuhlmann and Möhl (2007) and Holan et al. (1998).

Why Non-Projectivity Matters?

- CFGs cannot handle non-projective constructions:

Think in Dutch that **John grass** saw **being-cut**!

Why Non-Projectivity Matters?

- CFGs cannot handle non-projective constructions:

Think in Dutch that **John grass saw being-cut!**

- No way to glue these crossing dependencies together:

- Lexical choice:

$X \rightarrow \langle \text{grass } X \text{ being-cut, } \text{trávu } X \text{ sekát} \rangle$

- Agreement in gender:

$X \rightarrow \langle \text{John } X \text{ saw, Jan } X \text{ viděl} \rangle$

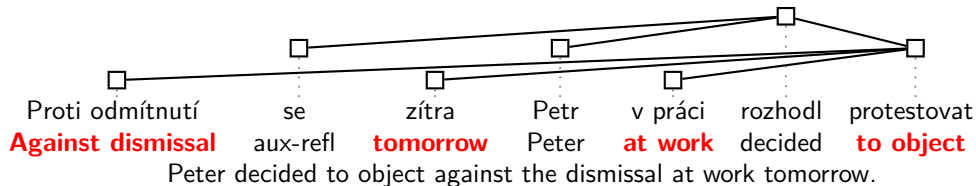
$X \rightarrow \langle \text{Mary } X \text{ saw, Marie } X \text{ viděla} \rangle$

- Phrasal chunks can memorize fixed sequences containing:
 - the non-projective construction
 - and all the words in between! (\Rightarrow extreme sparseness)

Is Non-Projectivity Severe?

In principle:

- Czech allows long gaps as well as many gaps in a subtree.



In treebank data:

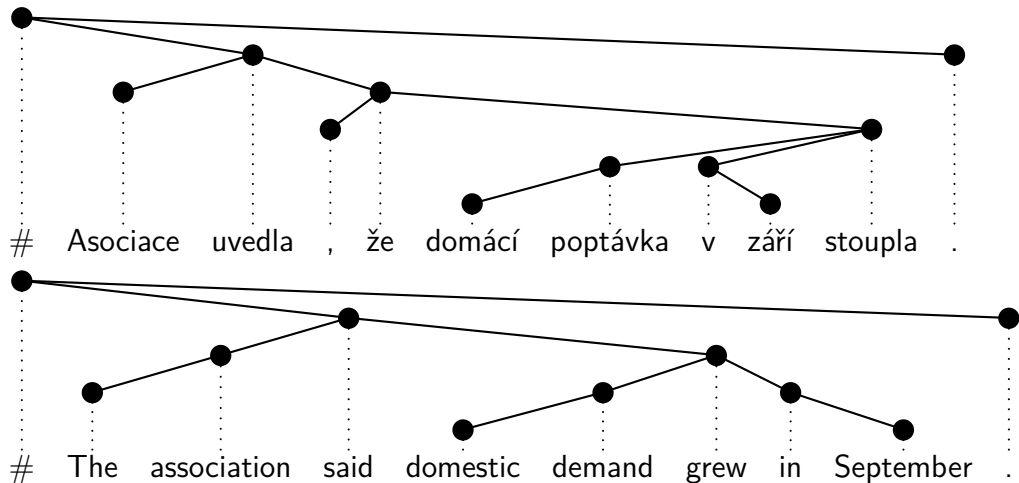
- ⊖ 23% of Czech sentences contain a non-projectivity.
- ⊕ 99.5% of Czech sentences are well nested with ≤ 1 gap.

Summary of Dependency Trees

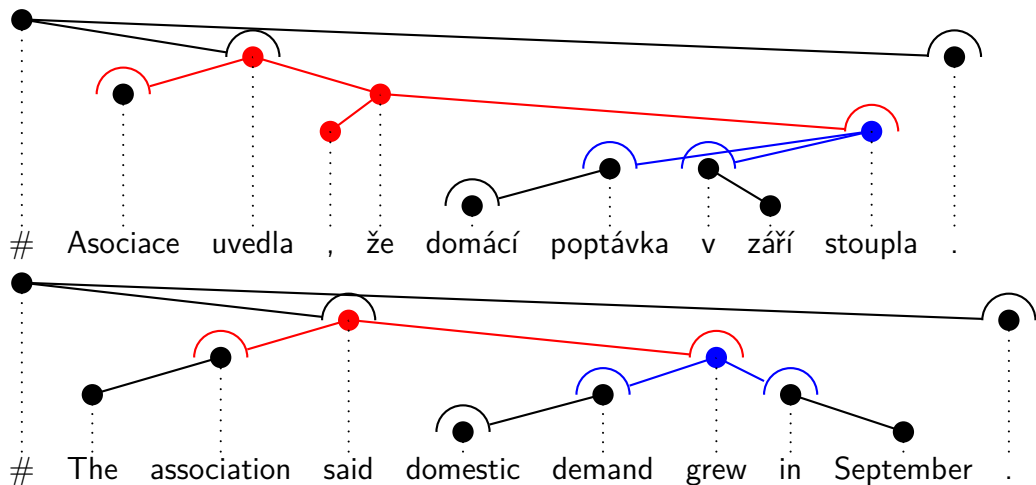
- More appropriate for Czech (frequent non-projectivity).
- Exhibit less divergence across languages (Fox, 2002).
- Dependency context more relevant than adjacency context.

So let's look if we can apply them in MT.

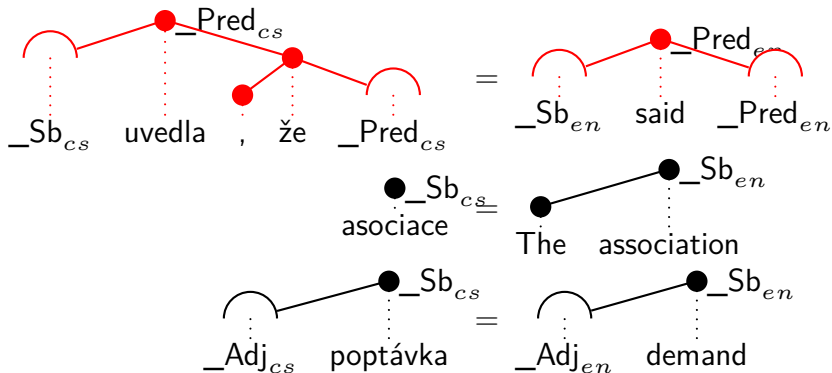
Idea: Observe a Pair of Trees...



...Decompose into Treelets...



...Collect Dict. of Treelet Pairs

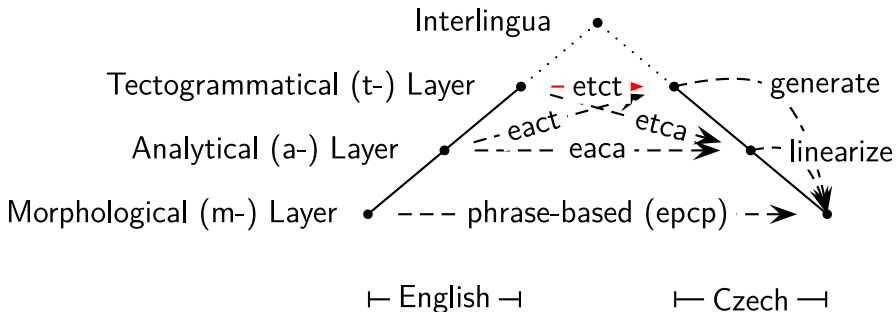


...Synchronous Tree Substitution Grammar,

e.g. Čmejrek (2006).

Transfer at Various Layers

- My thesis main goal: Transfer at deep-syntactic layer.
- Implementation to be applicable anywhere with dependency trees.



Deep Syntax

See MT Talk #14:

<http://youtu.be/1JwCW2mFk2M>

Going Deeper

- Motivation for tectogrammatical layer.
- T-Layer in STSG:
 - Complexity of t-layer attributes.
 - Factorization inevitable, but how to factorize?
 - Empirical evaluation.
- TectoMT Transfer.
 - Hidden Markov Tree Model.

Tectogrammatics: Deep Syntax Culminating

Background: Prague Linguistic Circle (since 1926).

Theory: Sgall (1967), Panevová (1980), Sgall et al. (1986).

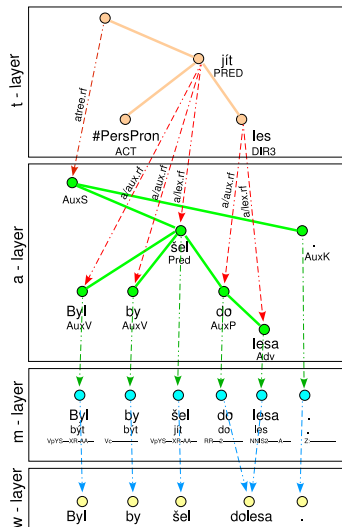
Materialized theory — Treebanks:

- Czech: PDT 1.0 (2001), PDT 2.0 (2006)
- Czech-English: PCEDT 1.0 (2004), PCEDT 2.0 (2012)
- Arabic: PADT (2004)

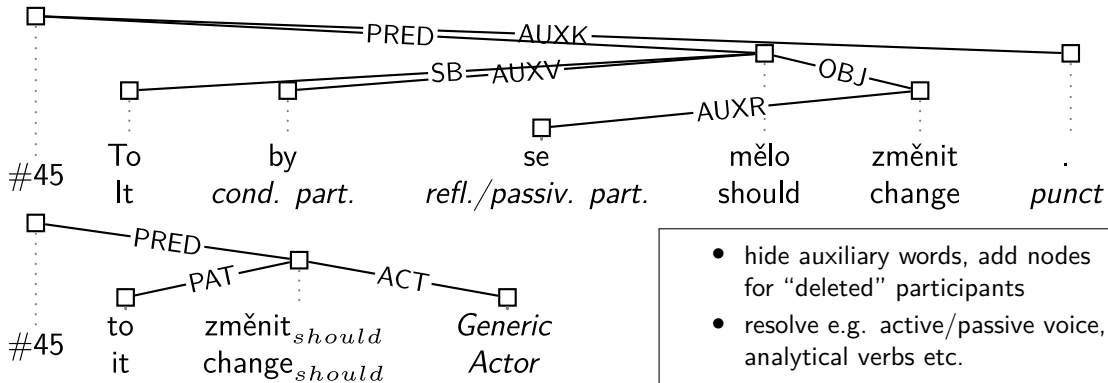
Practice — Tools:

- 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)
- generating Czech surface from t-layer: Ptáček and Žabokrtský (2006)

Layers in PDT

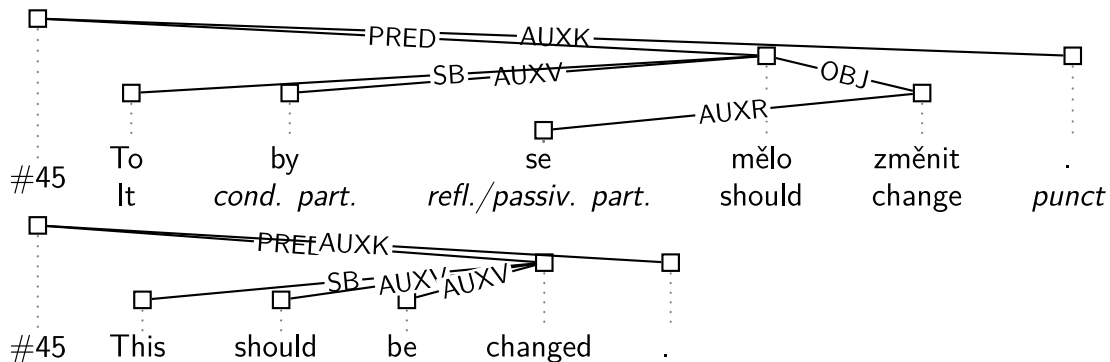


Analytical vs. Tectogrammatical

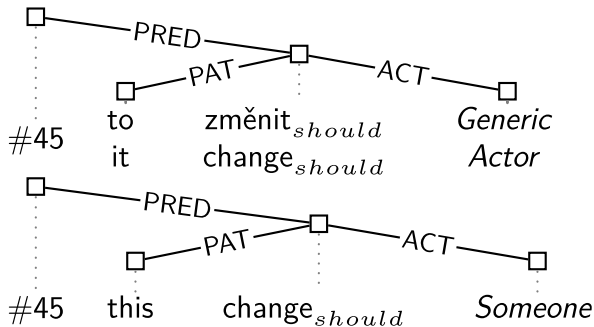


- hide auxiliary words, add nodes for “deleted” participants
- resolve e.g. active/passive voice, analytical verbs etc.
- “full” t-layer resolves much more, e.g. topic-focus articulation or anaphora

Czech and English A-Layer



Czech and English T-Layer



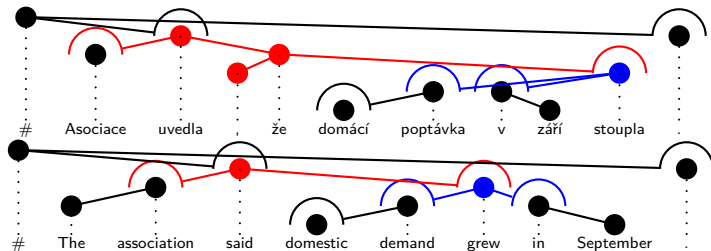
Predicate-argument structure: $\text{change}_{\text{should}}(\text{ACT: someone, PAT: it})$

The Tectogrammatical Hope

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

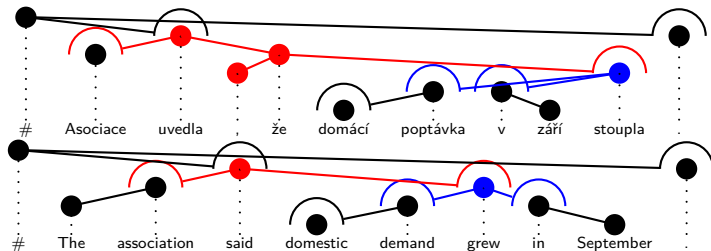
- Reduced structure size (auxiliary words disappear).
- Long-distance dependencies (non-projectivites) solved at t-layer.
- Word order ignored / interpreted as information structure (given/new).
- Reduced vocabulary size (Czech morphological complexity).
- Czech and English t-trees structurally more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

Reminder: STSG



1. Decompose input tree into treelets.
2. Replace treelets with their translations.
3. Join output treelets.

Reminder: STSG



1. Decompose input tree into treelets.
2. Replace treelets with their translations.
3. Join output treelets.

Real t-nodes have 25 attributes! \Rightarrow Can't treat them as atomic.

Structure vs. Attributes

Factorization = introduction of independence assumptions.

- STSG factorizes along structure (input into treelets).
- T-layer requires factorization along attributes.

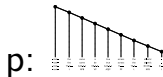
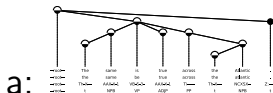
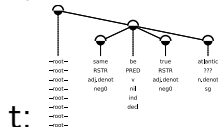
Which should go first?

- Treelets of attributes?
 - Similar to phrases of factors, synchronous approach.
 - Can easily fill up stacks with treelets differing too little.
- Layers of trees?
 - Would be hard to ensure matching tree structure.
 - Rather use a few attributes to construct structure and postpone the choice of others until the tree is finished.

Transfer at Various Layers

Layers \ Language Models	no LM	<i>n</i> -gram/ <i>binode</i>
epcp, no factors	8.65 ± 0.55	10.90 ± 0.63
eaca, no factors	6.59 ± 0.52	8.75 ± 0.61
etct 2009; 43k	-	7.39 ± 0.52
etca, no factors	-	6.30 ± 0.57
etct factored, preserving structure	5.31 ± 0.53	5.61 ± 0.50
eact, source factored, output atomic	-	3.03 ± 0.32
etct, no factors, all attributes	1.61 ± 0.33	2.56 ± 0.35
etct, no factors, just t-lemmas	0.67 ± 0.19	-

etct 2009: strall + wfwind. LM rescoring. Formemes (not functors) as frontier labels.
Improved node-to-node alignment (Mareček et al., 2008). New generation pipeline.

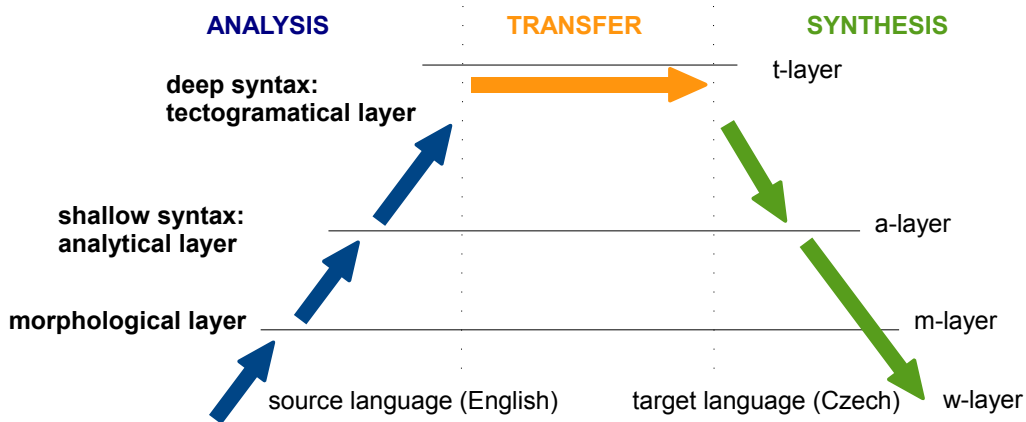


(WMT07 DevTest)

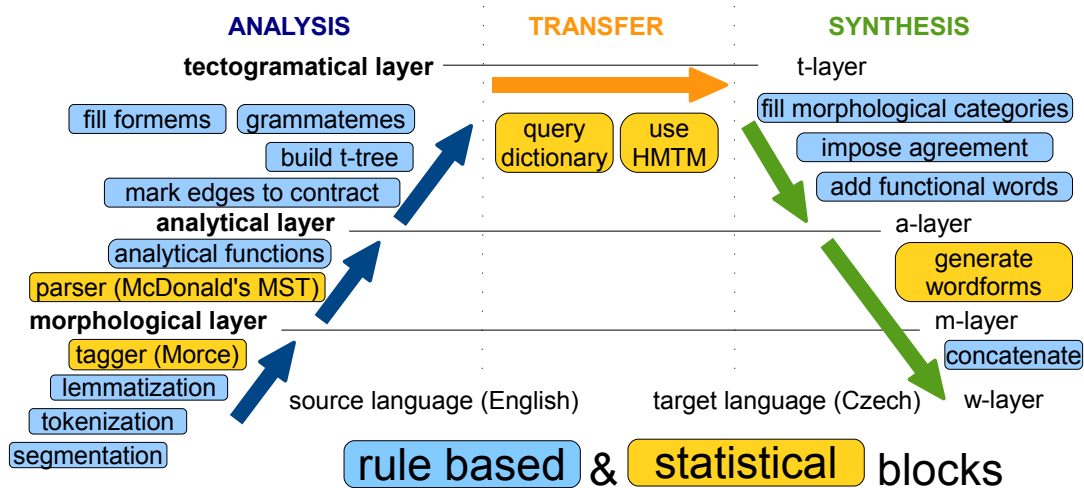
Reasons of STSG Bad Performance

- **Cumulation of Errors** in annotation pipeline.
- **Data Loss** due to incompatible structures:
 - Error in parses or word-alignment prevents treelet pair extraction.
- **Combinatorial Explosion** of factored output:
 - Abundance of t-node attribute combinations
⇒ e.g. lexically different translation options pushed off the stack
⇒ n -bestlist varies in unimportant attributes.
- **Too Strong Independence Assumptions:**
 - Should never analyze and factorize phrases seen often enough.
- **Complex models hard to tune:**
 - More models ⇒ Minimum error rate training has harder time.

“TectoMT Transfer” (1/3)



“TectoMT Transfer” (2/3)



“TectoMT Transfer” (3/3)

Slides 6–28 by Martin Popel (2009):

- Illustration of TectoMT transfer.
- Hidden Markov Tree Model (HMTM).

Demo Translation – Analysis



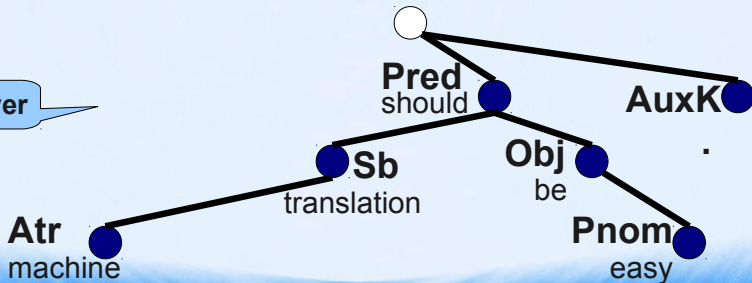
raw text

Machine translation should be easy.

m-layer

machine translation should be easy .
NN NN MD VB JJ .

a-layer



Demo Translation – Analysis



raw text

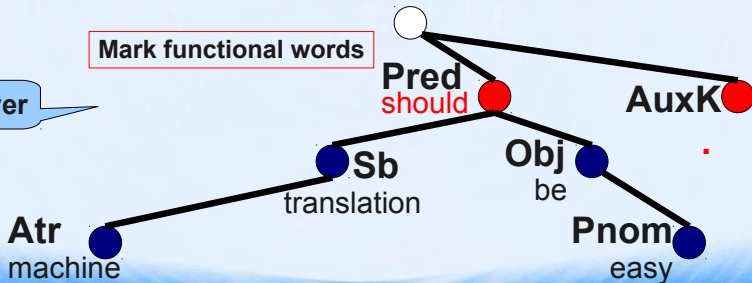
Machine translation should be easy.

m-layer

machine translation should be easy .
NN NN MD VB JJ .

a-layer

Mark functional words



Demo Translation – Analysis



raw text

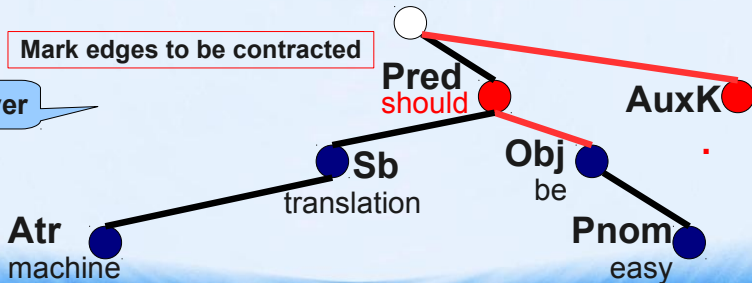
Machine translation should be easy.

m-layer

machine translation should be easy .
NN NN MD VB JJ .

Mark edges to be contracted

a-layer



Demo Translation – Analysis



raw text

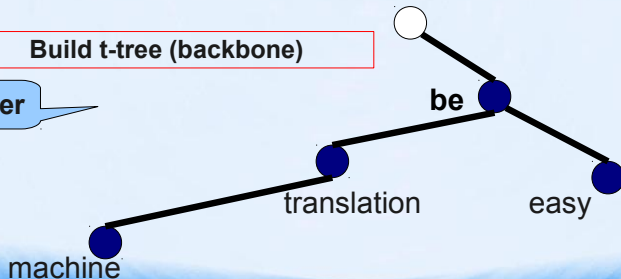
Machine translation should be easy.

m-layer

●	●	●	●	●	●
machine	translation	should	be	easy	.
NN	NN	MD	VB	JJ	.

Build t-tree (backbone)

t-layer



Demo Translation – Analysis



raw text

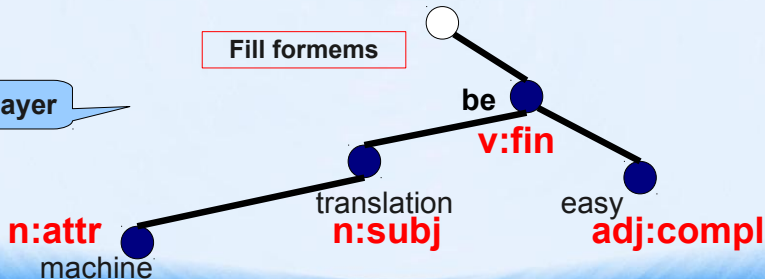
Machine translation should be easy.

m-layer

●	●	●	●	●	●
machine	translation	should	be	easy	.
NN	NN	MD	VB	JJ	.

t-layer

Fill formems



Demo Translation – Analysis



raw text

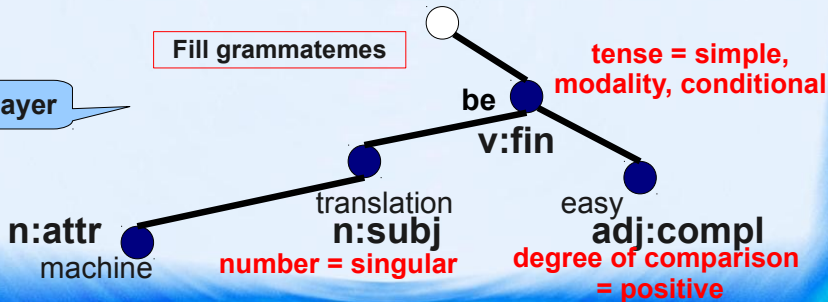
Machine translation should be easy.

m-layer

machine translation should be easy .
NN NN MD VB JJ .

t-layer

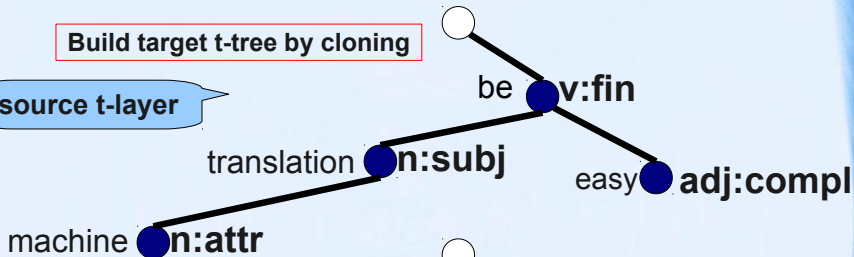
Fill grammatememes



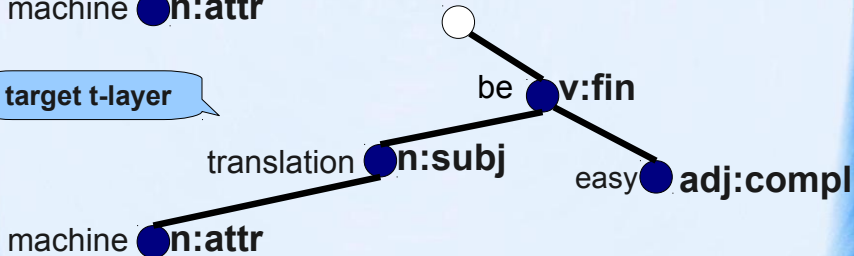
Demo Translation – Transfer

Build target t-tree by cloning

source t-layer



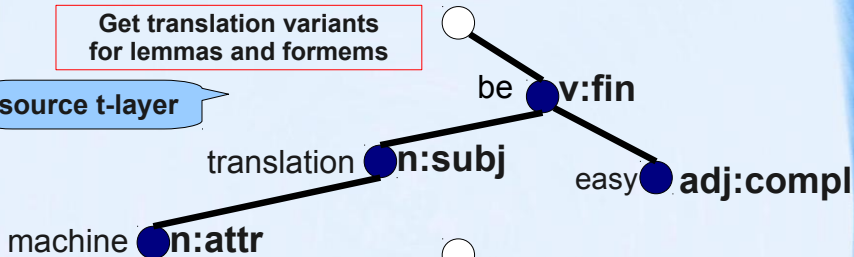
target t-layer



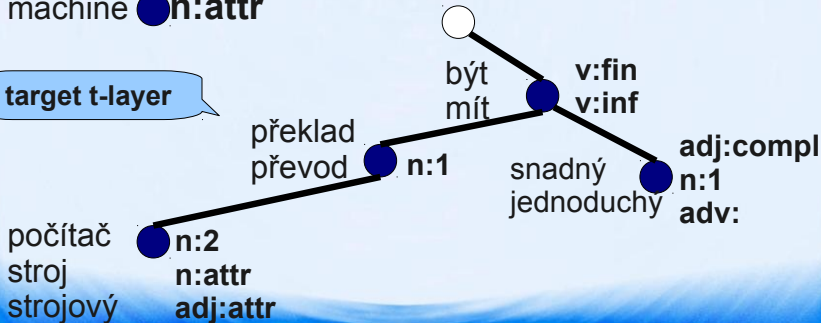
Demo Translation – Transfer

Get translation variants
for lemmas and formems

source t-layer



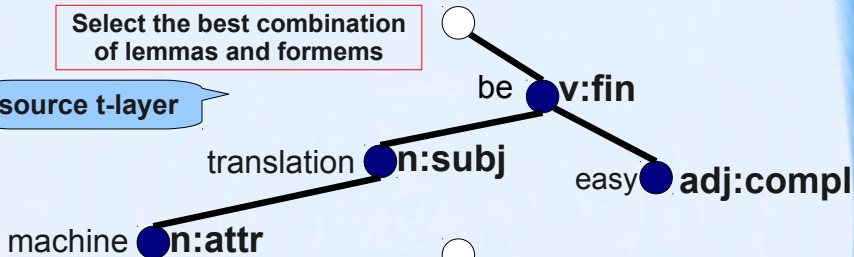
target t-layer



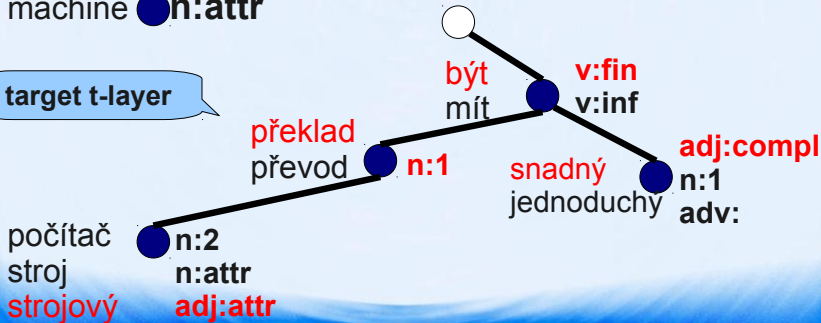
Demo Translation – Transfer

Select the best combination
of lemmas and formems

source t-layer



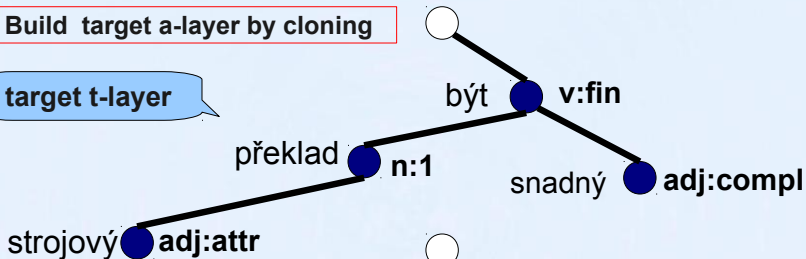
target t-layer



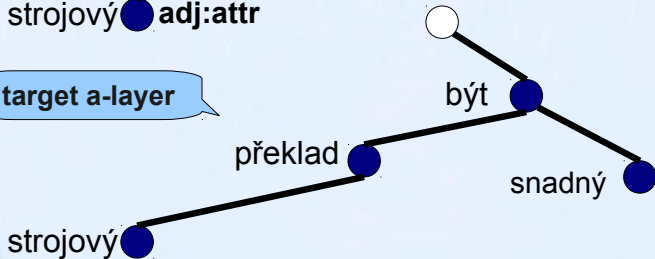
Demo Translation – Synthesis

Build target a-layer by cloning

target t-layer



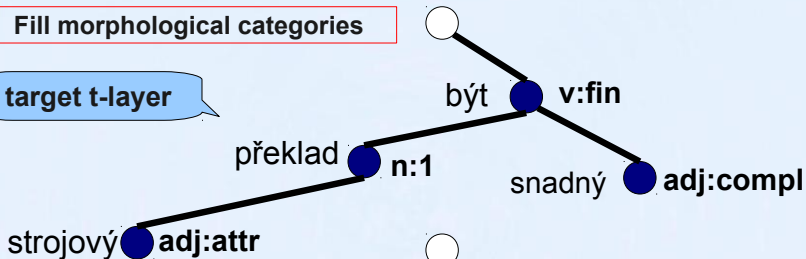
target a-layer



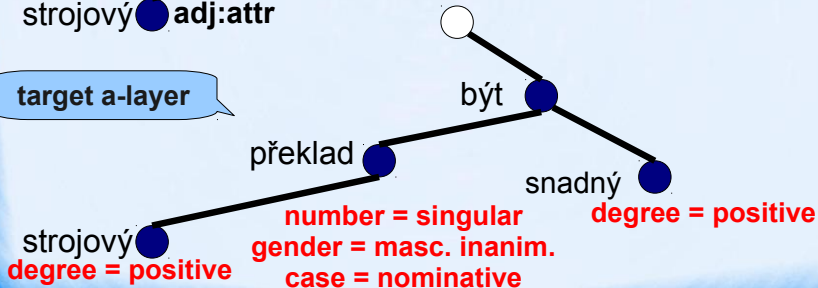
Demo Translation – Synthesis

Fill morphological categories

target t-layer



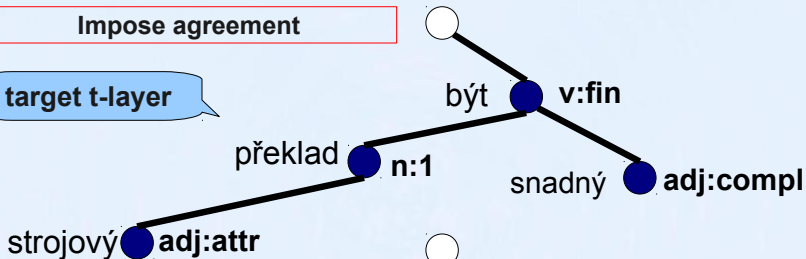
target a-layer



Demo Translation – Synthesis

Impose agreement

target t-layer



target a-layer

number = singular
case = nominative
gender = masc. inanim.

strojový
degree = positive

překlad
number = singular
gender = masc. inanim.
case = nominative

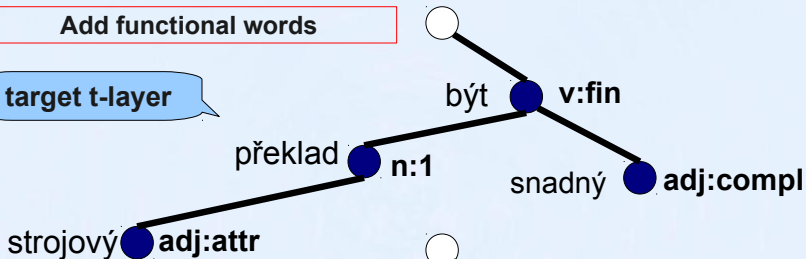
být
number = singular
gender = masc. inanim.

snadný
degree = positive
number = singular
case = nominative
gender = masc. inanim.

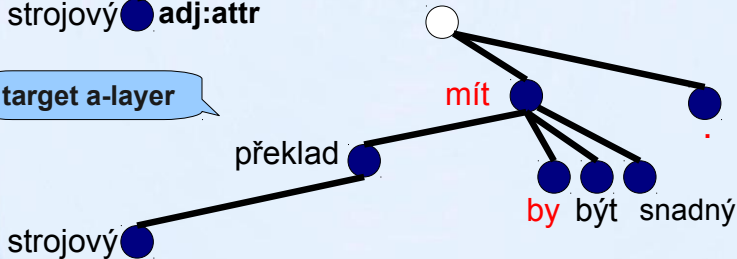
Demo Translation – Synthesis

Add functional words

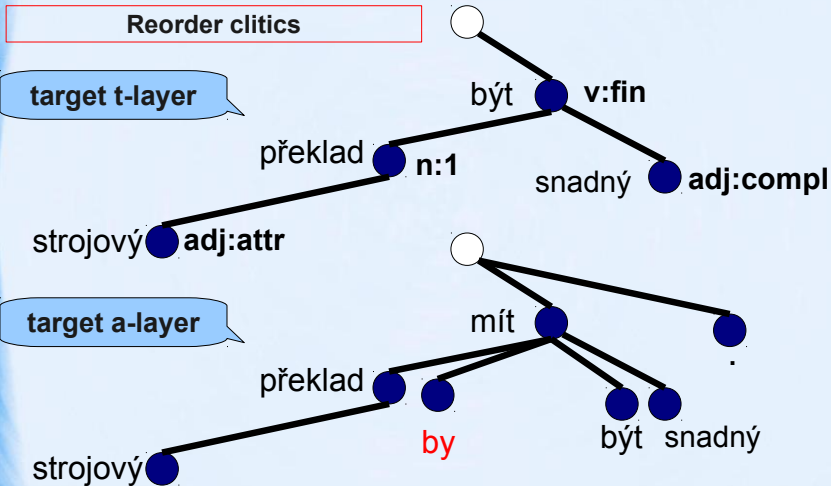
target t-layer



target a-layer



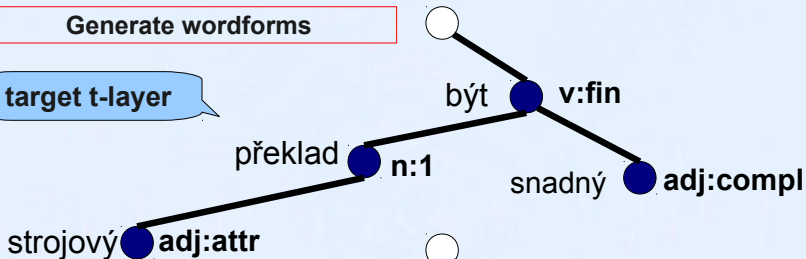
Demo Translation – Synthesis



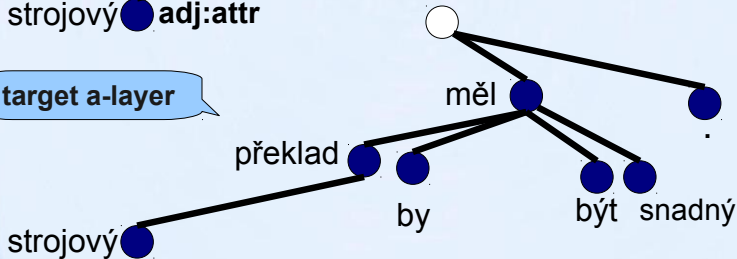
Demo Translation – Synthesis

Generate wordforms

target t-layer



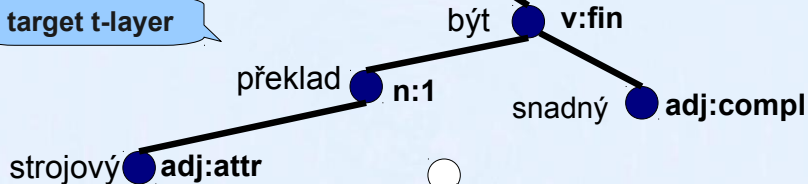
target a-layer



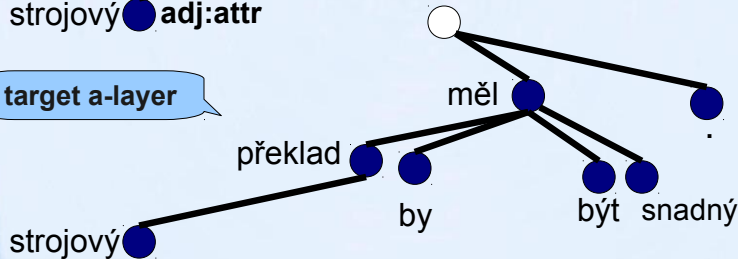
Demo Translation – Synthesis

Concatenate tokens for output

target t-layer



target a-layer



Strojový překlad by měl být snadný.

HMTM – Motivation

Select the best label for each node

source t-layer

translation
n:subj

machine **on:attr**

be ov:fin

easy●adj:compl

target t-layer

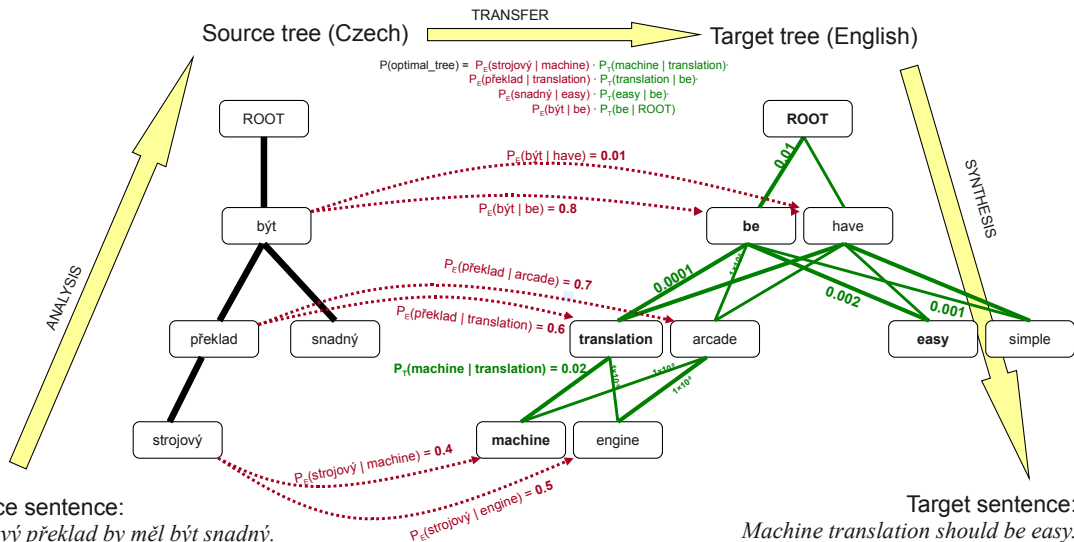
překlad|n:1,
převod|n:1

počítač|n:2, ●
počítač|n:attr,
strojový|adj:attr, ...

být|v:fin, být|v:inf,
mít|v:fin, mít|v:inf

snadný|adj:compl,
jednoduchý|adj:compl, ...

Hidden Markov Tree Model



Summary

- Dependency trees linguistically more promising.
 - Tree context vs. linear context. Non-projectivity. T-layer.
- STSG to transfer dependency trees:
 - Severe issues of sparseness, i.a. due to missing adjunction.
- TectoMT system with HMTM transfer.

Rich annotation **hurts** if not **backed-off**.

- Due to sparser data (incompatible trees).
- Due to cummulation of errors.
- Due to too strong independence assumptions.
- Due to harder optimization problem for MERT.

References

- Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. A systematic analysis of translation model search spaces. In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 224–232, Athens, Greece, March. Association for Computational Linguistics.
- David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- David Chiang. 2010. Learning to translate with source and target syntax. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.
- Martin Čmejrek. 2006. Using Dependency Tree Structure for Czech-English Machine Translation. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.
- Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, pages 304–311. Association for Computational Linguistics.
- Tomáš Holan, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 1998. Two Useful Measures of Word Order Complexity. In A. Polguere and S. Kahane, editors, Proceedings of the Coling '98 Workshop: Processing of Dependency-Based Grammars, Montreal. University of Montreal.
- Václav Klimeš. 2006. Analytical and Tectogrammatical Analysis of a Natural Language. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.
- Marco Kuhlmann and Mathias Möhl. 2007. Mildly context-sensitive dependency languages. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 160–167, Prague, Czech Republic, June. Association for Computational Linguistics.