Tree Automata in Parsing and Machine Translation

Andreas Maletti

Institute for Natural Language Processing
University of Stuttgart

Highlights — September 9, 2016

Parsing

Parsing

- determining the syntactic structure of a sentence
- subject to a given theory of syntax encoded in the training data

Parsing

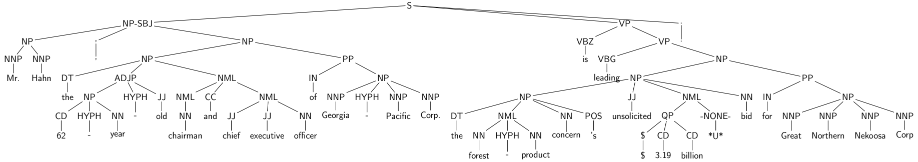
Parsing

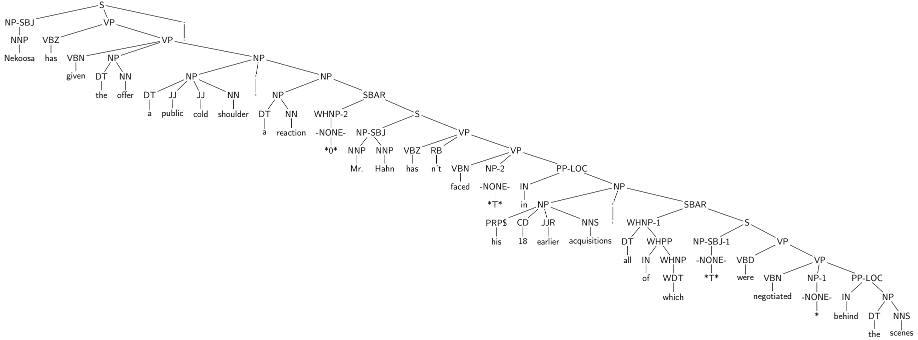
- determining the syntactic structure of a sentence
- subject to a given theory of syntax encoded in the training data

Theories

- Constituent syntax
- Dependency syntax
- ...

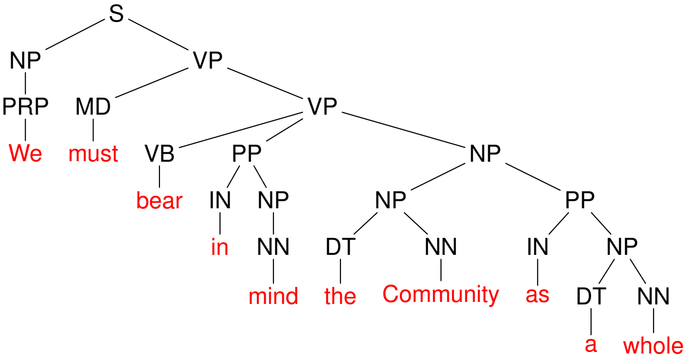
Penn tree bank (50,000 trees for English)





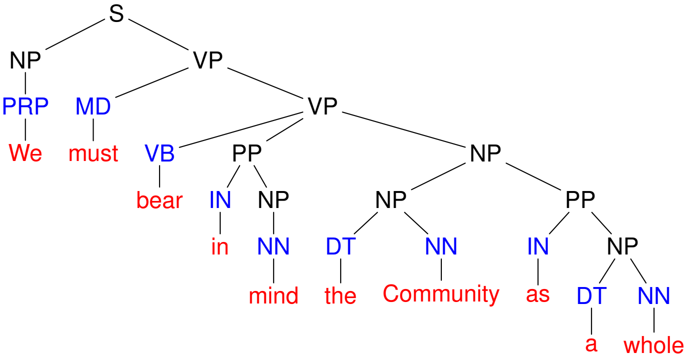
Constituent Parsing

Example: We must bear in mind the Community as a whole



Constituent Parsing

Example: We must bear in mind the Community as a whole



POS-tag: part-of-speech tag, “class” of a word

Constituent Parsing

Short history

… Chomskyan approach (perfect analysis, poor coverage)

- hand-crafted CFG, TAG refined via POS tags
- corrections and selection by human annotators

90s Statistical approach (cheap, automatically trained)

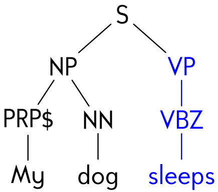
- Penn and WSJ tree bank (1M and 30M words)
- automatically obtained weighted CFG

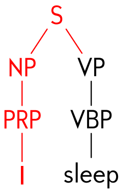
00s Subcategorization

- manual, e.g., Collins (1999), Stanford (2003)
- manual and reranking, e.g. BLLIP (2005)
- automatic, e.g. Berkeley (2007)

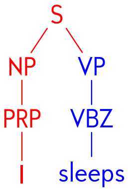
Subcategorization

These CFG derivations

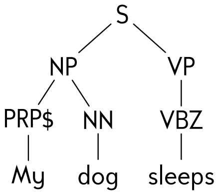




also admit



Subcategorization

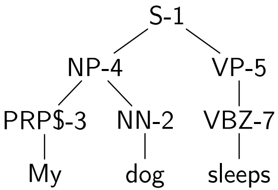


Tags:

- official tags often conservative
  - English: ≈ 50 tags
  - German: $\gg 200$ tags

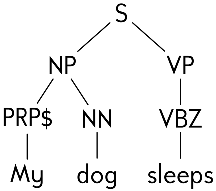
ADJA-Sup-Dat-Sg-Fem

Subcategorization



Tags:

- official tags often conservative
  - English: $\approx$ 50 tags
  - German: $\gg$ 200 tags   ADJA-Sup-Dat-Sg-Fem
- all modern parsers use refined tags $\rightarrow$ subcategorization



Tags:

- official tags often conservative
  - English: $\approx$ 50 tags
  - German: $\gg$ 200 tags   ADJA-Sup-Dat-Sg-Fem
- all modern parsers use refined tags $\to$ subcategorization
- but return parse over official tags $\to$ relabeling

Definition

Tree automaton (Q, Σ, I, R)

- finite set Q of states
- finite set Σ of terminals
- initial states $I \subseteq Q$
- finite set R of rules of the form $q \to \sigma(q_1, \ldots, q_k)$

$(\sigma \in \Sigma,\ k \geq 0,\ q, q_1, \ldots, q_k \in Q)$

Example rules

$q_4 \to \mathrm{VP}(q_5, q_2, q_3)$ $\qquad$ $q_0 \to \mathrm{S}(q_1, q_4)$ $\qquad$ $q_0 \to \mathrm{S}(q_6, q_2)$

Tree Automaton

Derivation semantics

$\xi \Rightarrow \zeta$ for sentential forms $\xi, \zeta \in T_\Sigma(Q)$
if there exist leaf position w in ξ and rule $q \to r$ in R

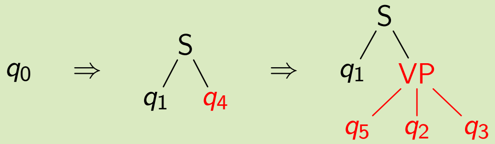
$$\xi = \xi[q]_w \qquad\qquad \zeta = \xi[r]_w$$

Recognized tree language

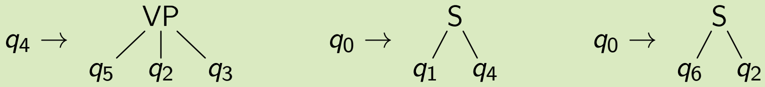
$$\{t \in T_\Sigma \mid \exists q_0 \in I \colon q_0 \Rightarrow^* t\}$$

Tree Automaton

Example derivation



Example rules



Constituent Parsing

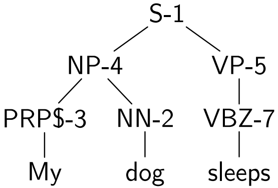
Grammar with subcategorization (grammar with relabeling)

- a grammar G generating $L(G) \subseteq T_\Sigma(W)$ (subcategorized trees)
- a (total) mapping $\rho\colon \Sigma \to \Delta$ (functional relabeling)

Constituent Parsing

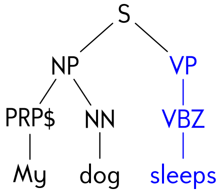
Grammar with subcategorization (grammar with relabeling)

- a grammar G generating $L(G) \subseteq T_\Sigma(W)$ (subcategorized trees)
- a (total) mapping $\rho \colon \Sigma \to \Delta$ (functional relabeling)



Grammar with subcategorization (grammar with relabeling)

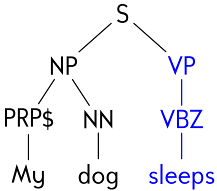
- a grammar G generating $L(G) \subseteq T_\Sigma(W)$ (subcategorized trees)
- a (total) mapping $\rho\colon \Sigma \to \Delta$ (functional relabeling)



Constituent Parsing

Grammar with subcategorization (grammar with relabeling)

- a grammar G generating $L(G) \subseteq T_\Sigma(W)$ (subcategorized trees)
- a (total) mapping $\rho \colon \Sigma \to \Delta$ (functional relabeling)



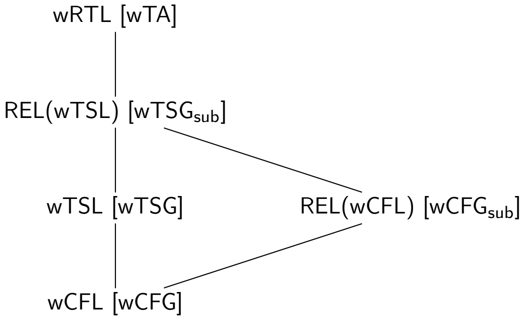
Semantics

$$L(G, \rho) = \rho(L(G)) = \{\rho(t) \mid t \in L(G)\}$$

Language class: REL($\mathcal{L}$) for language class $\mathcal{L}$

Theorem

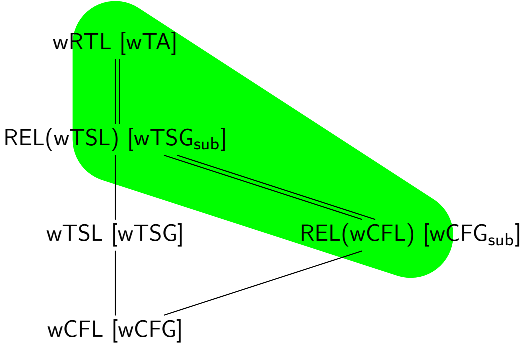
$\mathrm{REL}(\mathrm{wCFL}) = \mathrm{REL}(\mathrm{wTSL}) = \mathrm{wRTL}$



Constituent Parsing

Theorem

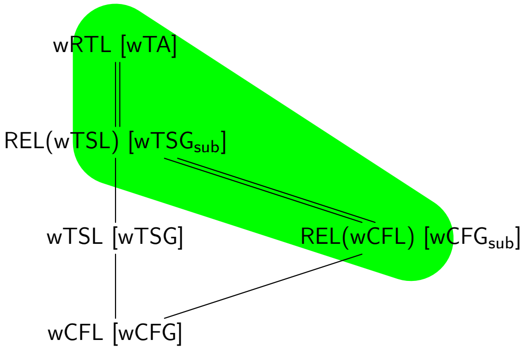
$\mathrm{REL}(\mathrm{wCFL}) = \mathrm{REL}(\mathrm{wTSL}) = \mathrm{wRTL}$



Constituent Parsing

Theorem

$\mathrm{REL}(\mathrm{wCFL}) = \mathrm{REL}(\mathrm{wTSL}) = \mathrm{wRTL}$



Hence: subcategorization ≈ finite-state

Constituent Parsing

Comparison

1. rule of subcategorized grammar:

$$\text{S-1} \to \text{ADJP-2 S-1} \qquad\qquad \text{weight: } 0.003545$$

with relabeling $\rho(\text{S-1}) = \text{S}, \ldots$

Constituent Parsing

Comparison

1. rule of subcategorized grammar:

$$\text{S-1} \rightarrow \text{ADJP-2 S-1} \qquad \text{weight: } 0.003545$$

   with relabeling $\rho(\text{S-1}) = \text{S}, \ldots$

2. corresponding rule of tree automaton

$$\text{S-1} \rightarrow \text{S}\big(\text{ADJP-2},\ \text{S-1}\big) \qquad \text{weight: } 0.003545$$

Constituent Parsing

State-of-the-art models

- CFG with subcategorization (CFG_{sub})
- tree substitution grammars with subcategorization (TSG_{sub})
  [Shindo et al., 2012]
- (both as expressive as weighted tree automata)
- other models

Constituent Parsing

State-of-the-art models

- CFG with subcategorization (CFG_{sub})
- tree substitution grammars with subcategorization (TSG_{sub})
  [Shindo et al., 2012]
- (both as expressive as weighted tree automata)
- other models

Experiment [Shindo et al., 2012]

grammar model	F_1
wCFG	72.6
wTSG [Cohn et al., 2010]	84.7
$wCFG_{sub}$ [Petrov, 2010]	91.8
$wTSG_{sub}$ [Shindo et al., 2012]	92.4

Constituent Parsing

Advances in NLP

- best learning algorithms from positive data
  (state splitting & EM)
- fastest evaluators of weighted tree automata
  (coarse-to-fine parsing)
- fastest *n*-best derivation extraction
- ...

Tree Automaton

Literature

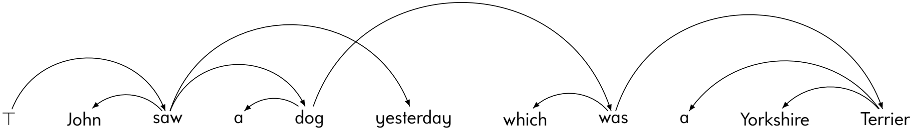
- Thatcher & Wright (1968):
  Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic.
  Math. Systems Theory 2(1): 57–81

- Rounds (1970): *Mappings and Grammars on Trees.*
  Mathematical Systems Theory 4(3): 257–287

- Gécseg & Steinby (1984): *Tree Automata.*
  Akadémiai Kiadó 2nd edition (2015): `arXiv:1509.06233`

- Gécseg & Steinby (1997): *Tree Languages.*
  In *Handbook of Formal Languages* 3: 1–68, Springer

Dependency Parsing



Problems

Given edge-weighted directed graph, extract “best” edge cover

- (general) [Edmonds, 1965]
- that is a tree [Chu-Liu & Edmonds, 1965–1967]
- that is projective tree [Eisner, 1996]
- that is acyclic NP-hard [Guruswami et al., 2011]
- that is a tree with page-number 2 [Gómez-Rodríguez & Nivre, 2013]
- that has page-number $k \geq 2$ NP-hard [Kuhlmann & Jonsson, 2015]
- that is a tree with page-number $k \geq 3$ open

Combinatory Categorial Grammars

$$
\begin{array}{c}
c \quad c \quad d \quad d \quad e \quad e \\[6pt]
\dfrac{\dfrac{\dfrac{C \qquad \dfrac{\dfrac{C \quad D/E/D\backslash C}{D/E/D} \qquad D/E\backslash C}{D/E/E\backslash C}}{D/E/E} \qquad E}{D/E} \qquad E}{D}
\end{array}
$$

The lexicon generates string language $\mathcal{L}$ with $\mathcal{L} \cap c^+ d^+ e^+ = \{c^i d^i e^i \mid i \geq 1\}$ for goal item D

$$
\begin{aligned}
L(c) &= \{C\} \\
L(d) &= \{D/E\backslash C, D/E/D\backslash C\} \\
L(e) &= \{E\}
\end{aligned}
$$

Combinatory Categorial Grammars

Problems

Under a suitable relabeling, characterize the set of valid proof trees

- for just applications $\rightarrow$ sub-regular tree languages
- for compositions of order 1 $\rightarrow$ open (probably still regular)
- for compositions of order $k \geq 2$ $\rightarrow$ open
- for arbitrary compositions $\rightarrow$ context-free tree language

Machine Translation

Review translation [by Google Translate]

1. The room it is not narrowly was a simple, bathtub was also attached.
2. Wi-fi, TV and I was available.
3. Church looked When morning awake open the curtain.
4. When looking at often, wives, went out and is invited to try to go [...].
5. But was a little cold, morning walks was good.

Machine Translation

Review translation [by Google Translate]

1. The room it is not narrowly was a simple, bathtub was also attached.
2. Wi-fi, TV and I was available.
3. Church looked When morning awake open the curtain.
4. When looking at often, wives, went out and is invited to try to go [...].
5. But was a little cold, morning walks was good.

Original [Japanese — © tripadvisor]

1. 部屋もシンプルでしたが狭くなく、バスタブもついていました。
2. Wi-fi、テレビも利用出来ました。
3. 朝起きてカーテンを開けると教会が見えました。
4. しばし眺めていると、妻たちは、[...]るから行こうとさそわれ出かけました。
5. ちょっと寒かったけれど、朝の散策はグッドでしたよ。

Short History

Timeline

1960 — **Dark age**

- rule-based systems (e.g., SYSTRAN)
- Chomskyan approach (perfect translation, poor coverage)

1991 — **Reformation**

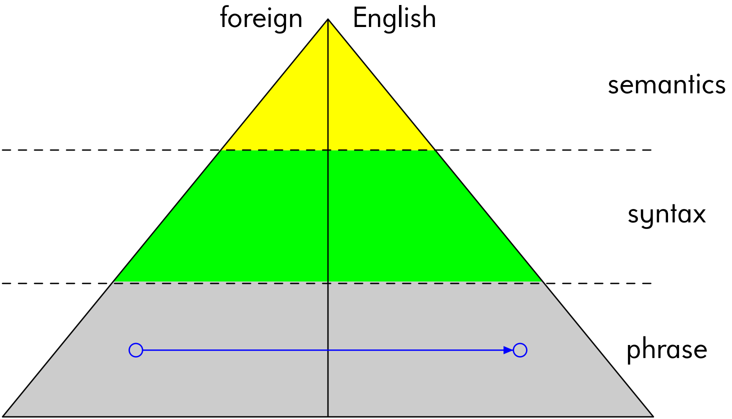
- phrase-based and syntax-based systems
- statistical approach (cheap, automatically trained)

2016 — **Potential future**

- semantics-based systems (e.g., FrameNet-based)
- semi-supervised, statistical approach
- basic understanding of (translated) text

Machine Translation

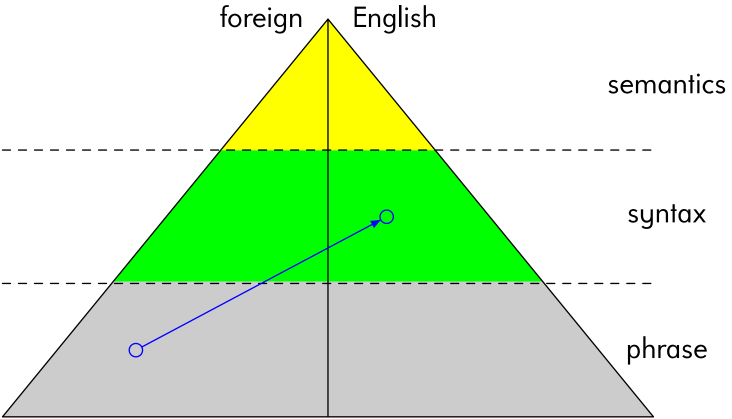
Vauquois triangle:



Translation model: string-to-string

Machine Translation

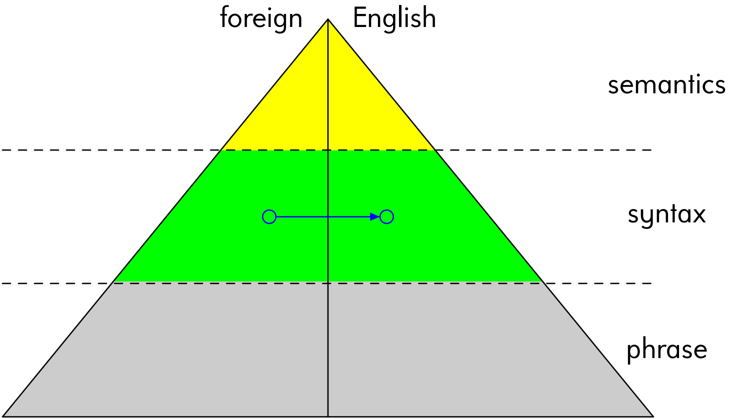
Vauquois triangle:



Translation model: string-to-tree

Machine Translation

Vauquois triangle:



Translation model: tree-to-tree

Machine Translation

Training data

- parallel corpus
- word alignments
- parse trees (for syntax-based systems)

Machine Translation

Training data

- parallel corpus
- word alignments
- parse trees (for syntax-based systems)

Parallel corpus

linguistic resource containing (sentence-by-sentence) example translations

Machine Translation

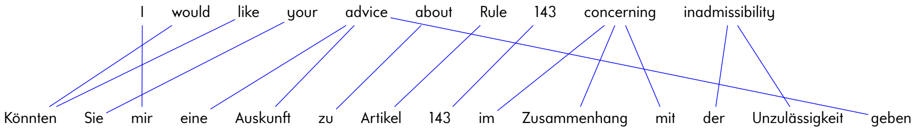
parallel corpus, word alignments, parse tree

I would like your advice about Rule 143 concerning inadmissibility

Könnten Sie mir eine Auskunft zu Artikel 143 im Zusammenhang mit der Unzulässigkeit geben

Machine Translation

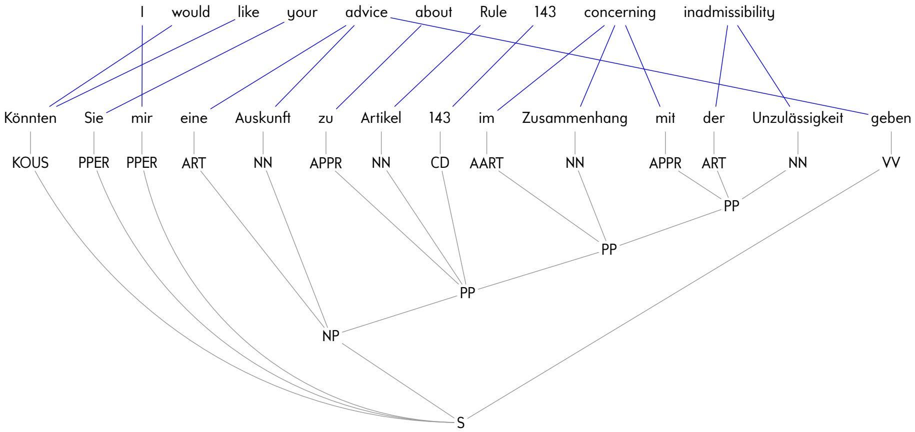
parallel corpus, word alignments, parse tree



via GIZA++ [Och, Ney: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 2003]

Machine Translation

parallel corpus, word alignments, parse tree

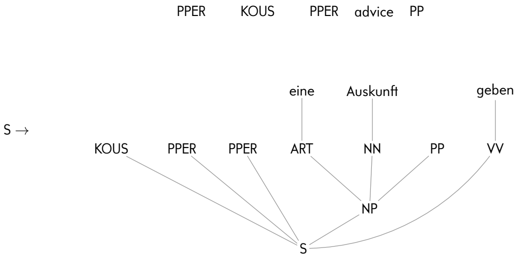


via Berkeley parser [Petrov, Barrett, Thibaux, Klein: Learning accurate, compact, and interpretable tree annotation. *Proc. ACL*, 2006]

Weighted Synchronous Grammars

Synchronous tree substitution grammar: productions $N \to (r, r_1)$

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand side r_1 of tree substitution grammar production

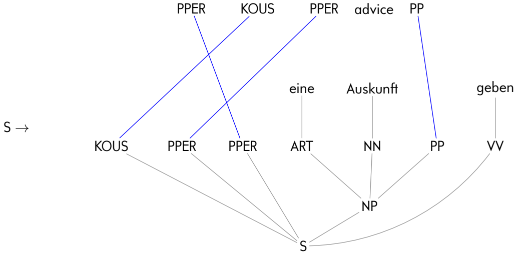


variant of [M., Graehl, Hopkins, Knight: The power of extended top-down tree transducers. *SIAM Journal on Computing* 39(2), 2009]

Weighted Synchronous Grammars

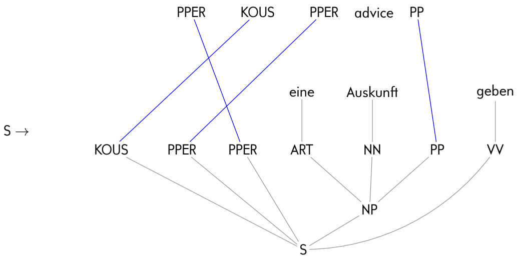
Synchronous tree substitution grammar: productions $N \to (r, r_1)$

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand side r_1 of tree substitution grammar production
- (bijective) synchronization of nonterminals



variant of [M., Graehl, Hopkins, Knight: The power of extended top-down tree transducers. *SIAM Journal on Computing* 39(2), 2009]

Synchronous Grammars

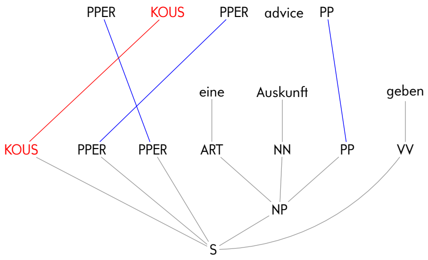


Production application

1. Selection of synchronous nonterminals

Synchronous Grammars

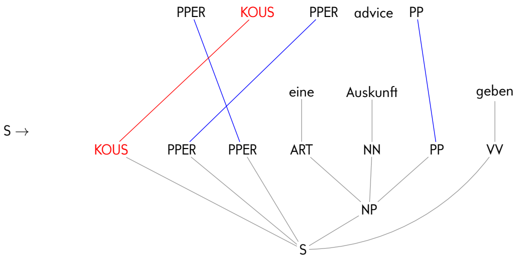
S →



Production application

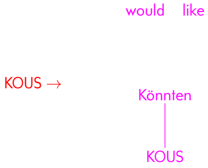
1. Selection of synchronous nonterminals

Synchronous Grammars

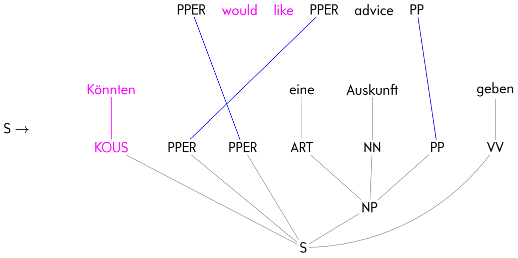


Production application

1. Selection of synchronous nonterminals
2. Selection of suitable production

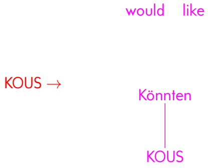


Synchronous Grammars

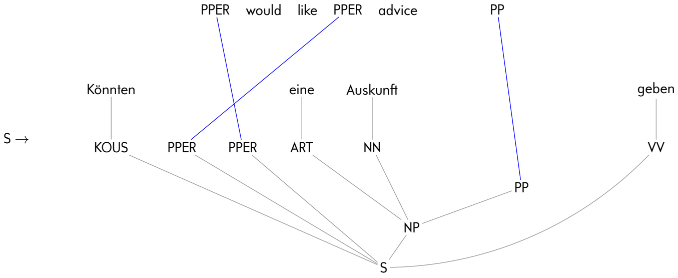


Production application

1. Selection of synchronous nonterminals
2. Selection of suitable production
3. Replacement on both sides



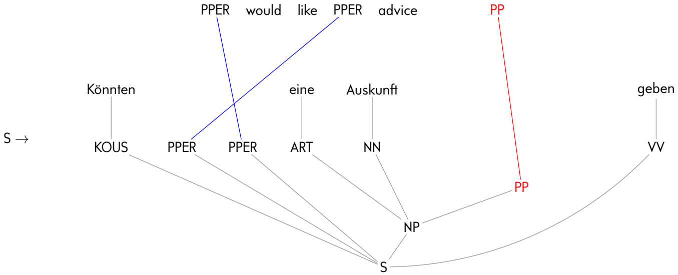
Synchronous Grammars



Production application

1. synchronous nonterminals

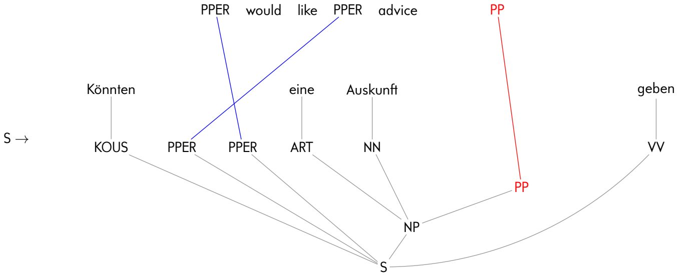
Synchronous Grammars



Production application

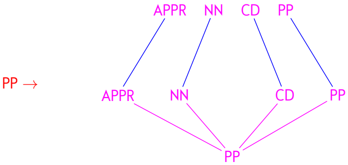
1. synchronous nonterminals

Synchronous Grammars

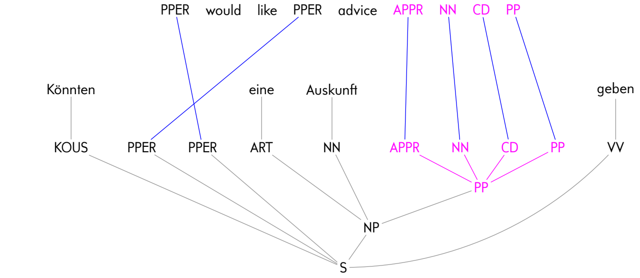


Production application

1. synchronous nonterminals
2. suitable production



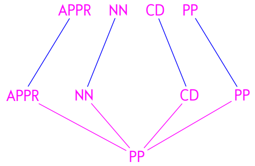
S →



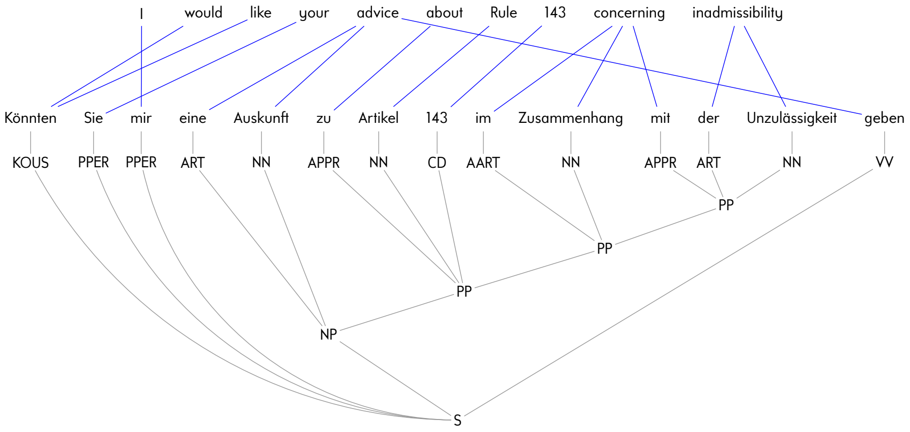
Production application

1. synchronous nonterminals
2. suitable production
3. replacement

PP →



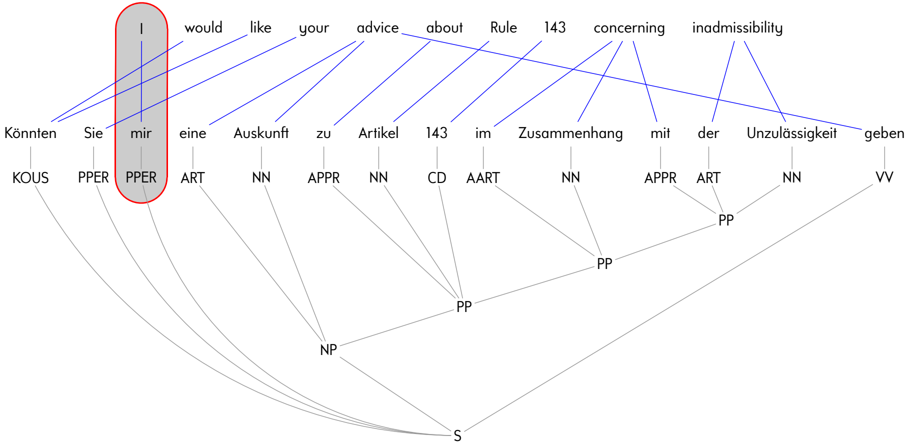
Production Extraction



following [Galley, Hopkins, Knight, Marcu: What's in a translation rule? *Proc. NAACL,* 2004]

Production Extraction

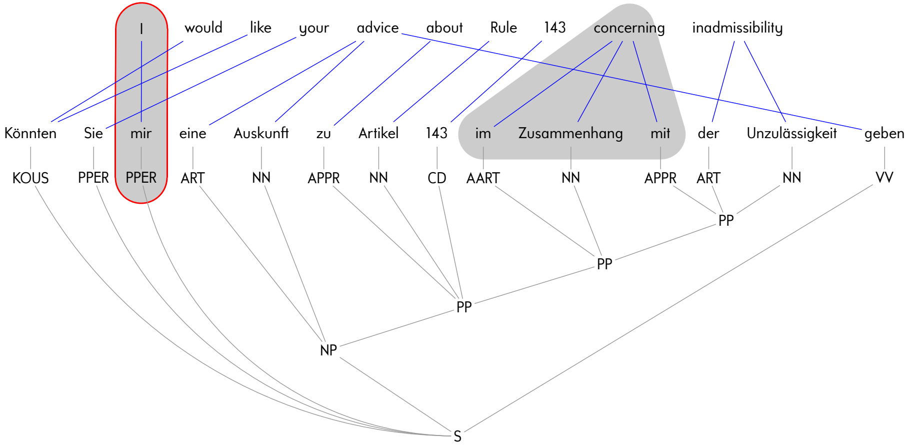
(extractable productions marked in red)



following [Galley, Hopkins, Knight, Marcu: What's in a translation rule? *Proc. NAACL*, 2004]

Production Extraction

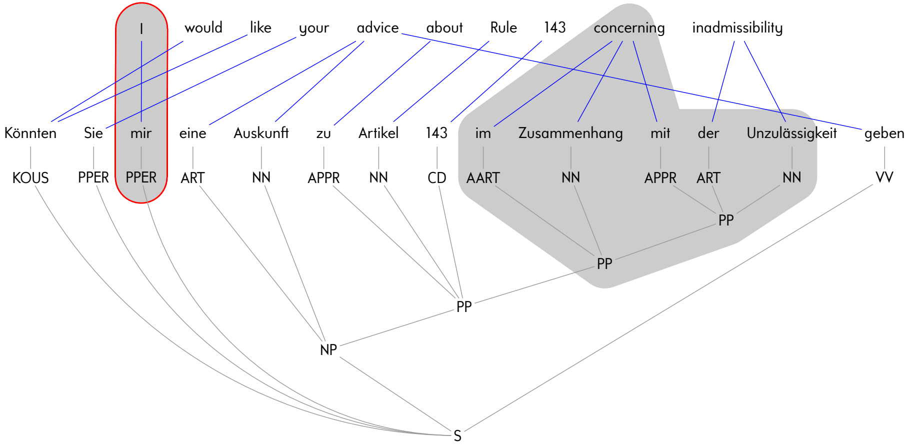
(extractable productions marked in red)



following [Galley, Hopkins, Knight, Marcu: What's in a translation rule? *Proc. NAACL*, 2004]

Production Extraction

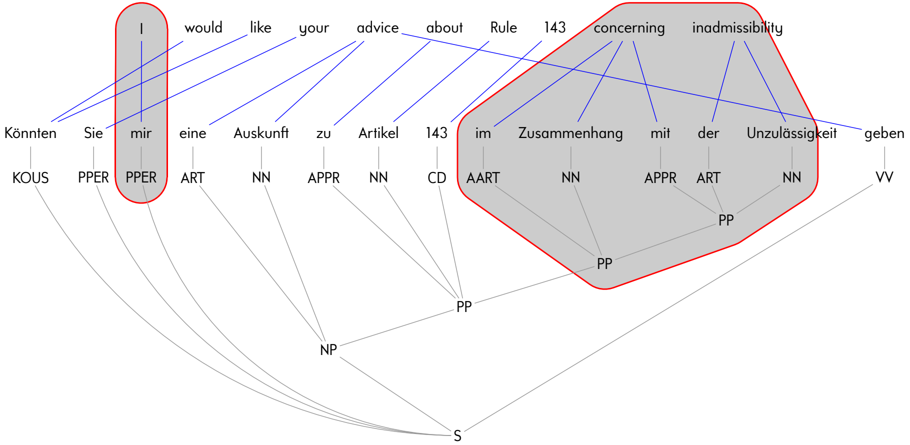
(extractable productions marked in red)



following [Galley, Hopkins, Knight, Marcu: What's in a translation rule? *Proc. NAACL*, 2004]

Production Extraction

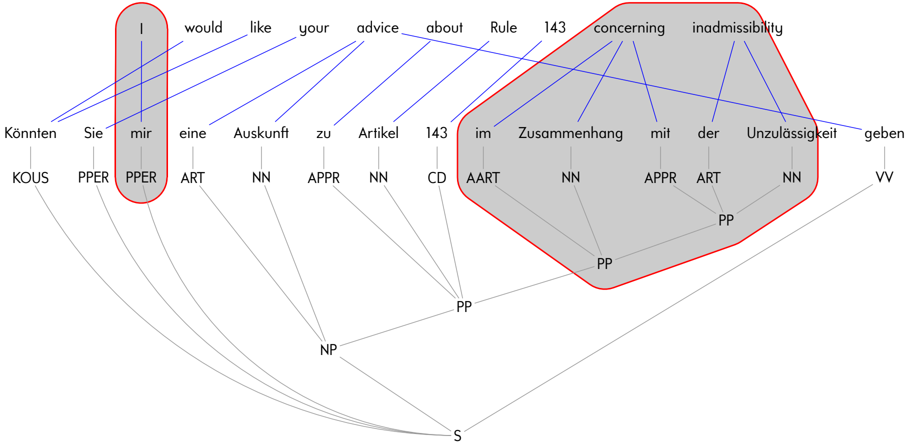
(extractable productions marked in red)



following [Galley, Hopkins, Knight, Marcu: What's in a translation rule? *Proc. NAACL*, 2004]

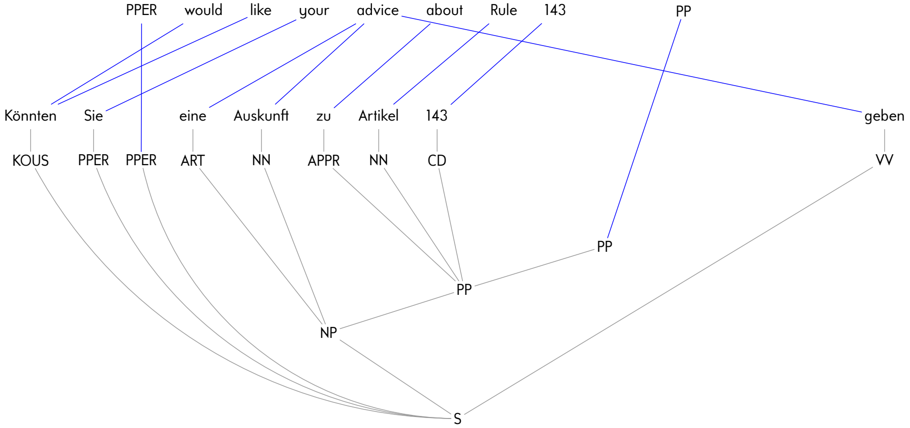
Production Extraction

Removal of extractable production:



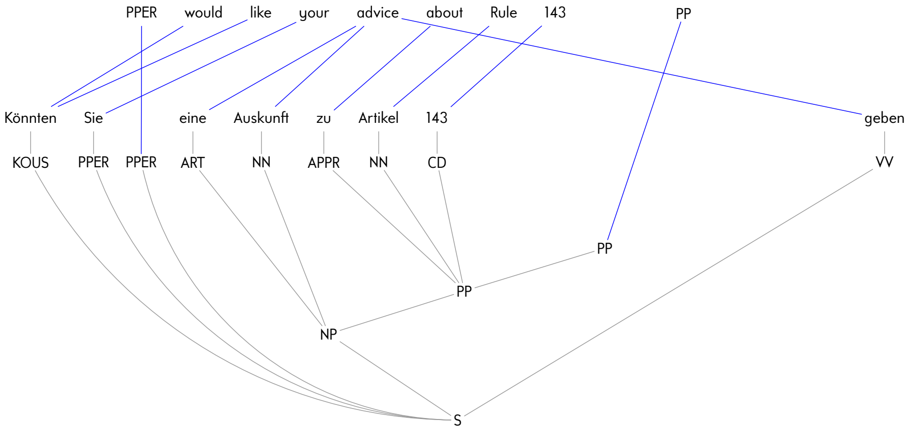
Production Extraction

Removal of extractable production:



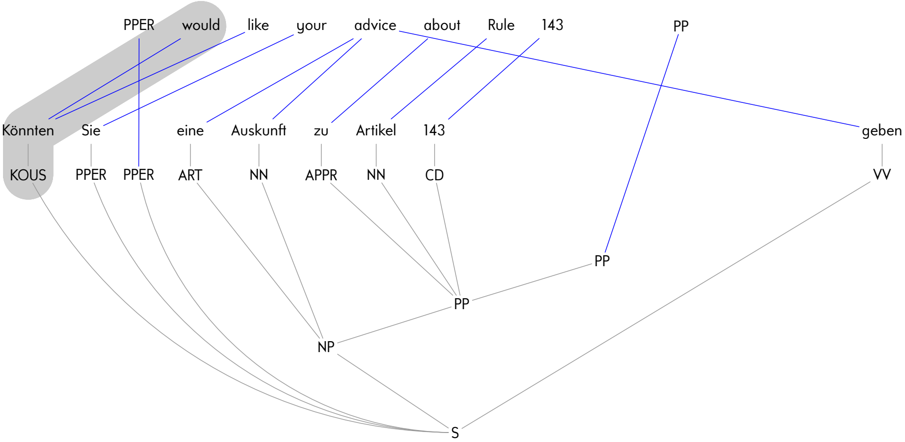
Production Extraction

Repeated production extraction:



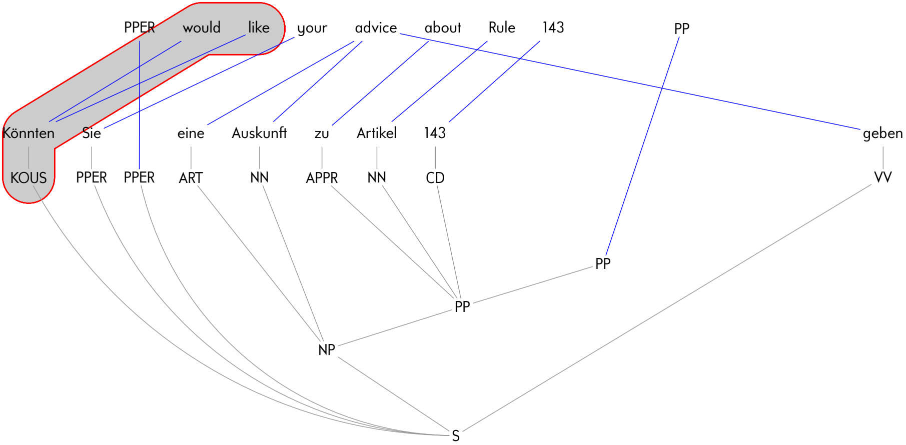
Production Extraction

Repeated production extraction:



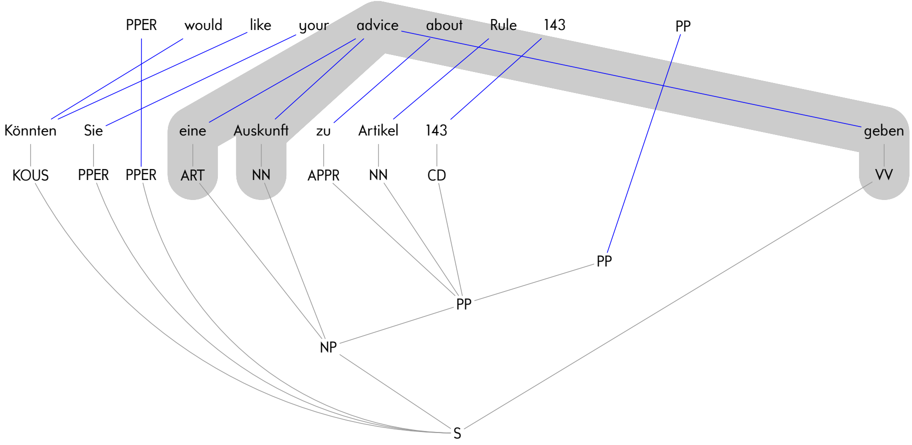
Production Extraction

Repeated production extraction: (extractable productions marked in red)



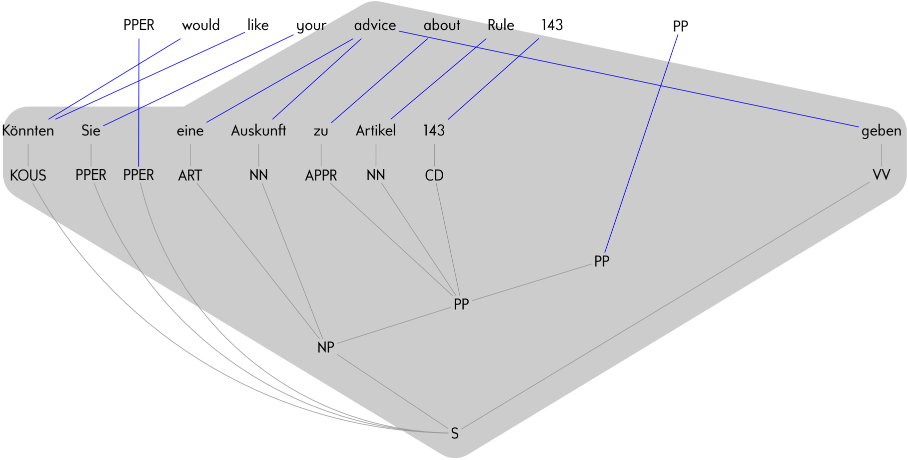
Production Extraction

Repeated production extraction: (extractable productions marked in red)



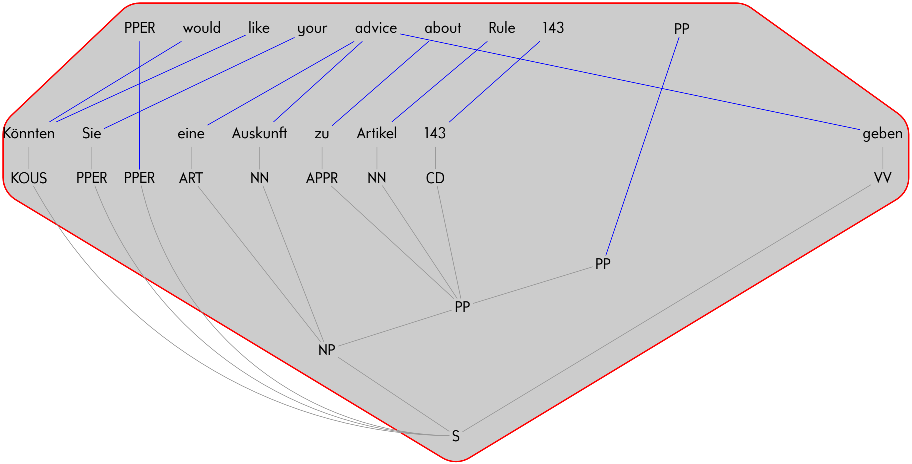
Production Extraction

Repeated production extraction: (extractable productions marked in red)



Production Extraction

Repeated production extraction: (extractable productions marked in red)



Synchronous Tree Substitution Grammars

Advantages

- very simple
- implemented in framework 'Moses'
  [Koehn et al.: Moses — Open source toolkit for statistical machine translation. *Proc. ACL,* 2007]
- "context-free"

Synchronous Tree Substitution Grammars

Advantages

- very simple
- implemented in framework 'Moses'
  [Koehn et al.: Moses — Open source toolkit for statistical machine translation. *Proc. ACL*, 2007]
- "context-free"

Disadvantages

- problems with discontinuities
- composition and binarization not possible
  [M., Graehl, Hopkins, Knight: The power of extended top-down tree transducers. *SIAM Journal on Computing* 39(2), 2009]
  [Zhang, Huang, Gildea, Knight: Synchronous Binarization for Machine Translation. *Proc. NAACL*, 2006]
- "context-free"

Evaluation

English → German translation task: (higher BLEU is better)

Type	System	BLEU		
		vanilla	WMT 2013	WMT 2015
string-to-tree	STSG	15.2	19.4	24.5
tree-to-tree	STSG	14.5	—	15.3

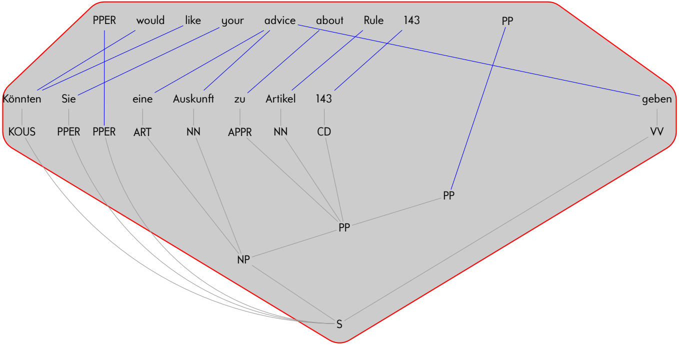
from [Seemann, Braune, M.: A systematic evaluation of MBOT in statistical machine translation. *Proc. MT-Summit*, 2015]
and [Bojar et al.: Findings of the 2013 workshop on statistical machine translation. *Proc. WMT*, 2013]
and [Bojar et al.: Findings of the 2015 workshop on statistical machine translation. *Proc. WMT*, 2015]

Conclusion

Observation

- syntax-based systems competitive with manual adjustments
- much less so for vanilla systems
- very unfortunate situation [more supervision yields lower scores]

Production Extraction



- very specific production
- every production for ‘advice’ contains sentence structure

(syntax “in the way”)

Synchronous Grammars

Synchronous multi tree substitution grammar: $N \to \big(r,\, \langle r_1, \ldots, r_n \rangle\big)$

variant of [M.: Why synchronous tree substitution grammars?. *Proc. NAACL*, 2010]

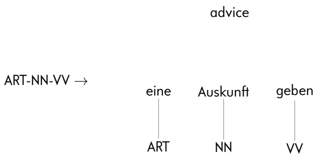
- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production

Synchronous Grammars

Synchronous multi tree substitution grammar: $N \to (r, \langle r_1, \ldots, r_n \rangle)$

variant of [M.: Why synchronous tree substitution grammars?. *Proc. NAACL*, 2010]

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production

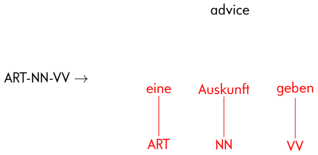


Synchronous Grammars

Synchronous multi tree substitution grammar: $N \to (r, \langle r_1, \ldots, r_n \rangle)$

variant of [M.: Why synchronous tree substitution grammars?. *Proc. NAACL*, 2010]

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production



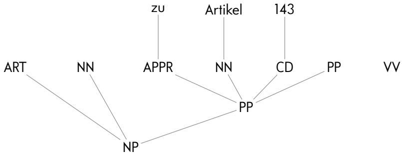
Synchronous multi tree substitution grammar: $N \to (r, \langle r_1, \ldots, r_n \rangle)$

variant of [M.: Why synchronous tree substitution grammars?. *Proc. NAACL*, 2010]

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production

ART-NN-VV about Rule 143 PP

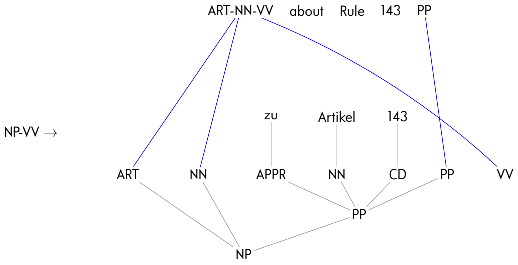
NP-VV $\to$



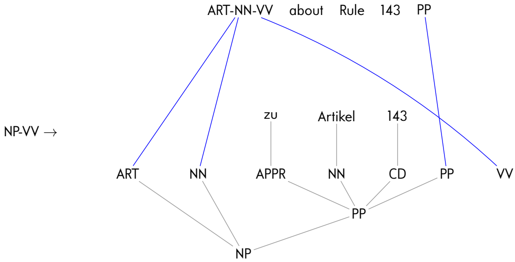
Synchronous multi tree substitution grammar: $N \to (r, \langle r_1, \ldots, r_n \rangle)$

variant of [M.: Why synchronous tree substitution grammars?. *Proc. NAACL*, 2010]

- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production
- synchronization via map NT $r_1, \ldots, r_n$ to NT r



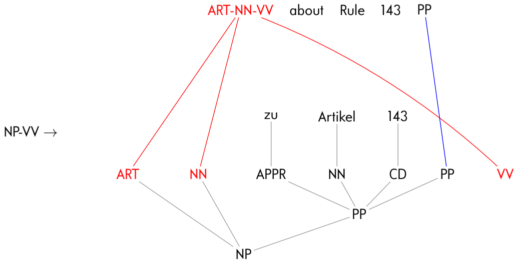
Synchronous Grammars



Production application

1. synchronous nonterminals

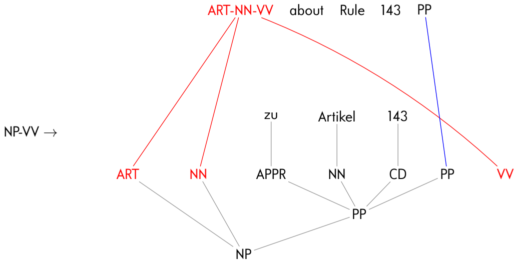
Synchronous Grammars



Production application

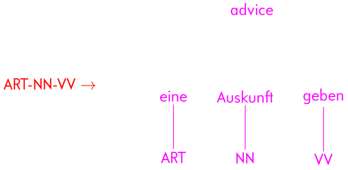
1. synchronous nonterminals

Synchronous Grammars

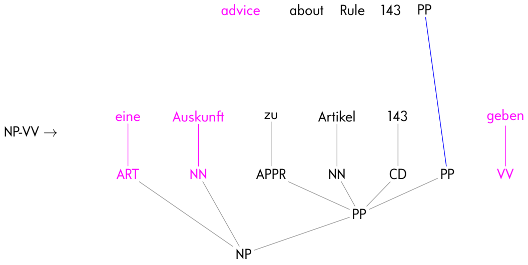


Production application

1. synchronous nonterminals
2. suitable production

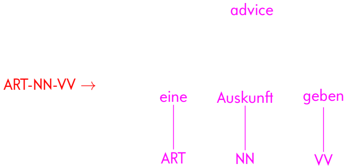


Synchronous Grammars

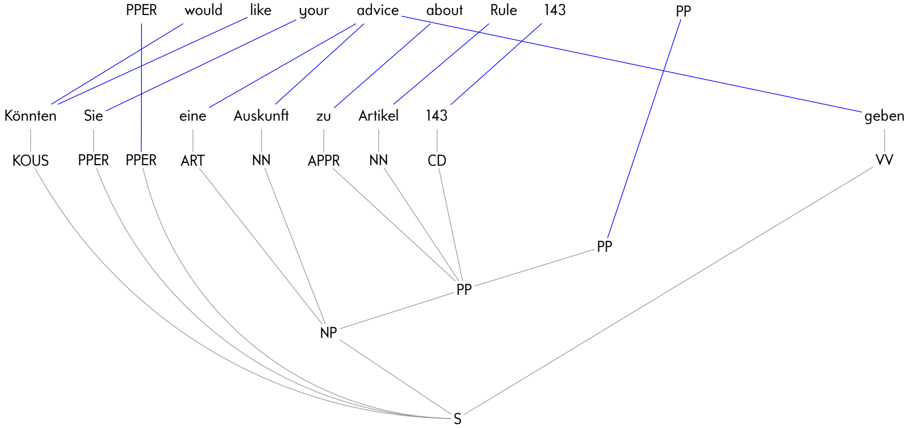


Production application

1. synchronous nonterminals
2. suitable production
3. replacement

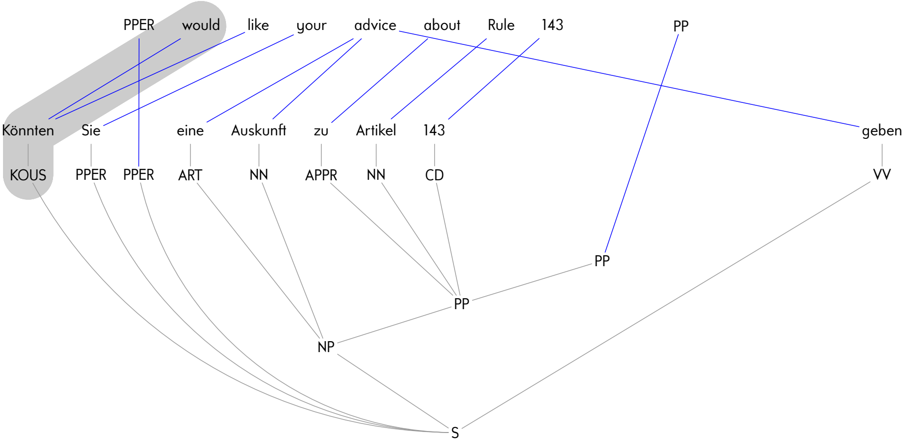


Production Extraction



variant of [M.: How to train your multi bottom-up tree transducer. *Proc. ACL*, 2011]

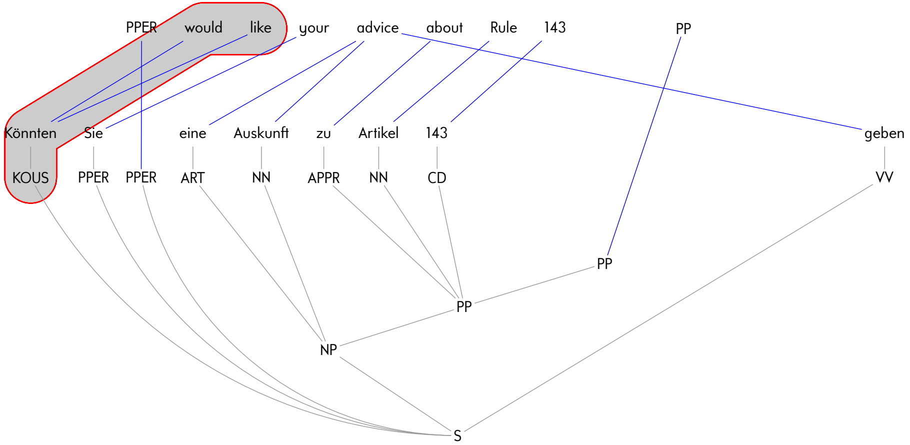
Production Extraction



variant of [M.: How to train your multi bottom-up tree transducer. *Proc. ACL*, 2011]

Production Extraction

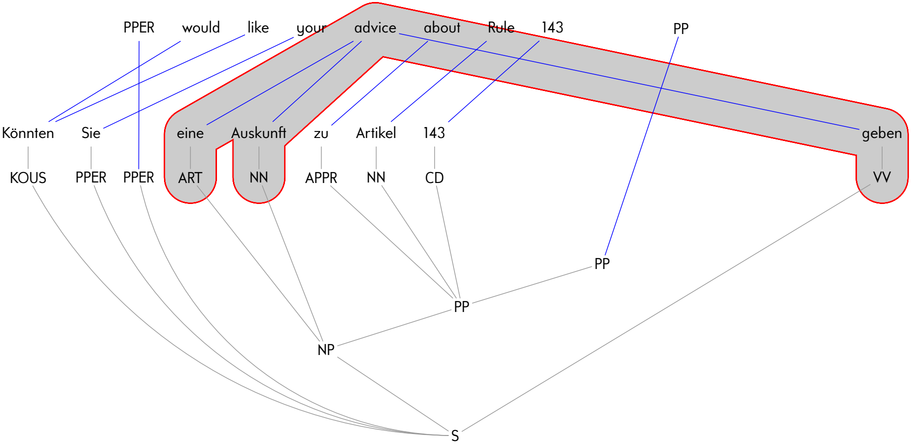
(extractable productions marked in red)



variant of [M.: How to train your multi bottom-up tree transducer. *Proc. ACL*, 2011]

Production Extraction

(extractable productions marked in red)



variant of [M.: How to train your multi bottom-up tree transducer. *Proc. ACL*, 2011]

Synchronous Multi Tree Substitution Grammars

Advantages

- complicated discontinuities
- implemented in framework 'Moses'
  [Braune, Seemann, Quernheim, M.: Shallow local multi bottom-up tree transducers in SMT. *Proc. ACL*, 2013]
- binarizable, composable

Synchronous Multi Tree Substitution Grammars

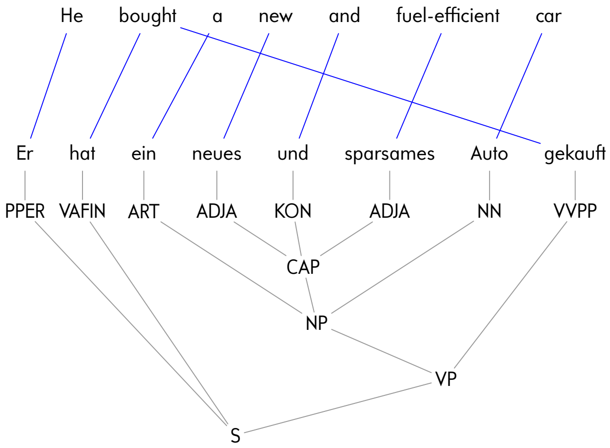
Advantages

- complicated discontinuities
- implemented in framework ‘Moses’
  [Braune, Seemann, Quernheim, M.: Shallow local multi bottom-up tree transducers in SMT. *Proc. ACL*, 2013]
- binarizable, composable

Disadvantages

- output non-regular (trees) or non-context-free (strings)
- not symmetric (input context-free; output not)

Discontinuity



Evaluation

System	Number of productions		
	E.-to-German	E.-to-Arabic	E.-to-Chinese
t-to-t STSG	7M	24M	8M
t-to-t SMTSG	41M	151M	84M
s-to-t STSG	14M	55M	17M
s-to-t SMTSG	144M	491M	162M

from [Seemann, Braune, M.: A systematic evaluation of MBOT in statistical machine translation. *Proc. MT-Summit,* 2015]

Evaluation

String-to-tree systems:

Task	BLEU	
	STSG	SMTSG
English $\rightarrow$ German	15.0	*15.5
English $\rightarrow$ Arabic	48.2	*49.1
English $\rightarrow$ Chinese	17.7	*18.4
English $\rightarrow$ Polish	21.3	*23.4
English $\rightarrow$ Russian	24.7	*26.1

from [Seemann, Braune, M.: A systematic evaluation of MBOT in statistical machine translation. *Proc. MT-Summit*, 2015]
and [Seemann, M.: Discontinuous statistical machine translation with target-side dependency syntax. *Proc. WMT*, 2015]

Evaluation

Conclusions

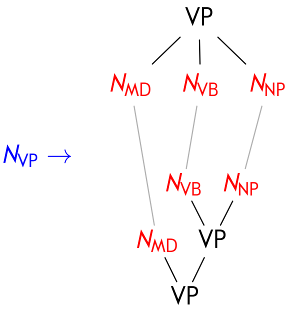
- consistent improvements
- 1 magnitude more productions
- SMTSG alleviate some of the problems of syntax-based systems

Synchronous Grammars

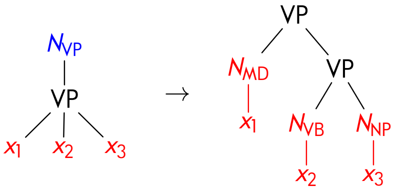
Notes

- tree-to-tree models easier for theoretical investigation
- strongly related to tree transducers
- we disallow trivial input sides of just a nonterminal (ε-free)

Synchronous grammar:



Tree transducer:



Synchronous Grammars

Major linear tree transducers:

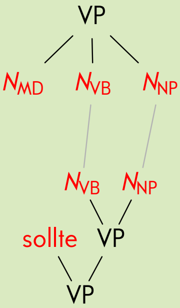
synchronization / input sides	**bijective**	**injective** (output $\to$ input)
shallow	nondeleting top-down ...	top-down ...
general	nondeleting extended ...	extended ...

Further distinction

- allow productions on disconnected input nonterminals
  $\to$ regular look-ahead
- allow arbitrary trees for disconnected input nonterminals
  $\to$ no look-ahead

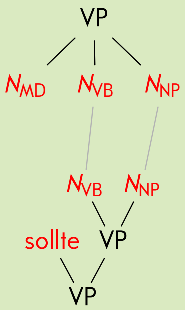
Synchronous Grammars

Illustration



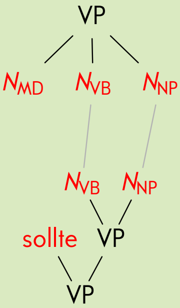
- no look-ahead: can plug any (terminal) tree for N_{MD}
  [e.g., NP(DT(the), NN(tower))]

Illustration



- no look-ahead: can plug any (terminal) tree for N_{MD} [e.g., NP(DT(the), NN(tower))]
- regular look-ahead: use special "no-output"-productions $N \to (r)$ [e.g., $N_{MD} \to$ (MD(should))]

Illustration



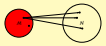
- no look-ahead: can plug any (terminal) tree for N_{MD} [e.g., $NP(DT(the), NN(tower))$]
- regular look-ahead: use special "no-output"-productions $N \to (r)$ [e.g., $N_{MD} \to (MD(should))$]
- SMTSG always have regular look-ahead (any number of components includes 0)

Evaluation criteria

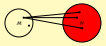
- rotations implementable? (for arbitrary t_1, t_2, t_3)

$$\sigma(\sigma(t_1, t_2), t_3) \mapsto \sigma(t_1, \sigma(t_2, t_3))$$

- symmetric?



- domain regular?



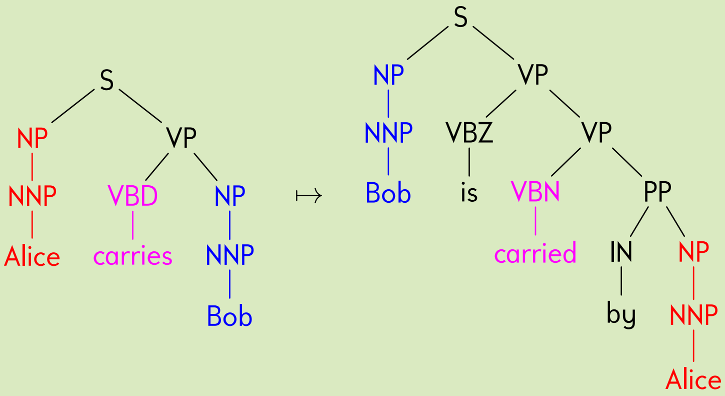
- range regular?
- closed under composition?

following [Knight: Capturing practical natural language transformations. *Machine Translation* 21(2), 2007]
and [May, Knight, Vogler: Efficient inference through cascades of weighted tree transducers. *Proc. ACL*, 2010]

Icons by interactivemania (`http://www.interactivemania.com/`) and UN Office for the Coordination of Humanitarian Affairs

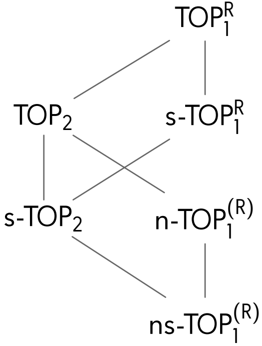
Synchronous Grammars

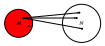
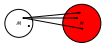
Illustration of rotations



Top-down Tree Transducer

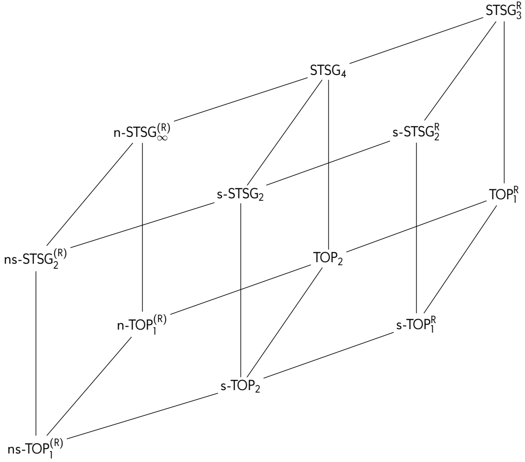
Hasse diagram with composition closure indicated in subscript:



Model \ Criterion					
ns-TOP	✗	✗	✓	✓	✓
n-TOP	✗	✗	✓	✓	✓
s-TOP	✗	✗	✓	✓	$✗_2$
s-TOP^R	✗	✗	✓	✓	✓
TOP	✗	✗	✓	✓	$✗_2$
TOP^R	✗	✗	✓	✓	✓

Synchronous Tree Substitution Grammars

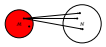
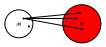
Hasse diagram with the composition closure indicated in subscript:



composition closures by

[Engelfriet, Fülöp, M.: Composition closure of linear extended top-down tree transducers. *Theory of Computing Systems*, to appear 2016]

Synchronous Tree Substitution Grammars

Model \ Criterion					
n-TOP	✗	✗	✓	✓	✓
TOP	✗	✗	✓	✓	✗$_2$
TOP^R	✗	✗	✓	✓	✓
ns-STSG	✓	✓	✓	✓	✗$_2$
n-STSG	✓	✗	✓	✓	✗$_\infty$
s-STSG$^{(R)}$	✓	✗	✓	✓	✗$_2$
STSG	✓	✗	✓	✓	✗$_4$
STSG^R	✓	✗	✓	✓	✗$_3$

Synchronous Multi Tree Substitution Grammars

Advantages of SMTSG

- always have regular look-ahead
- can always be made nondeleting & shallow
- closed under composition

[Engelfriet, Lilin, M.: Extended multi bottom-up tree transducers — composition and decomposition. *Acta Informatica* 46(8), 2009]

Synchronous Multi Tree Substitution Grammars

Advantages of SMTSG

- always have regular look-ahead
- can always be made nondeleting & shallow
- closed under composition

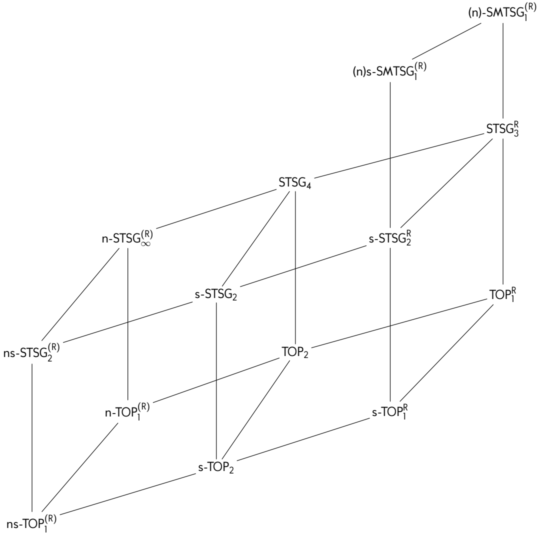
Disadvantages of SMTSG

- non-regular range

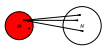
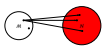
[Engelfriet, Lilin, M.: Extended multi bottom-up tree transducers — composition and decomposition. *Acta Informatica* 46(8), 2009]

Synchronous Multi Tree Substitution Grammars

Hasse diagram with the composition closure indicated in subscript:



Synchronous Multi Tree Substitution Grammars

Model \ Criterion					
n-TOP	✗	✗	✓	✓	✓
TOP	✗	✗	✓	✓	✗$_2$
TOP^R	✗	✗	✓	✓	✓
ns-STSG	✓	✓	✓	✓	✗$_2$
n-STSG	✓	✗	✓	✓	✗$_\infty$
s-STSG$^{(R)}$	✓	✗	✓	✓	✗$_2$
STSG	✓	✗	✓	✓	✗$_4$
STSG^R	✓	✗	✓	✓	✗$_3$
(n)s-SMTSG$^{(R)}$	✓	✗	✓	✗	✓
(n)-SMTSG$^{(R)}$	✓	✗	✓	✗	✓
reg.-range SMTSG	✓	✗	✓	✓	✓
symmetric SMTSG	✓	✓	✓	✓	✓

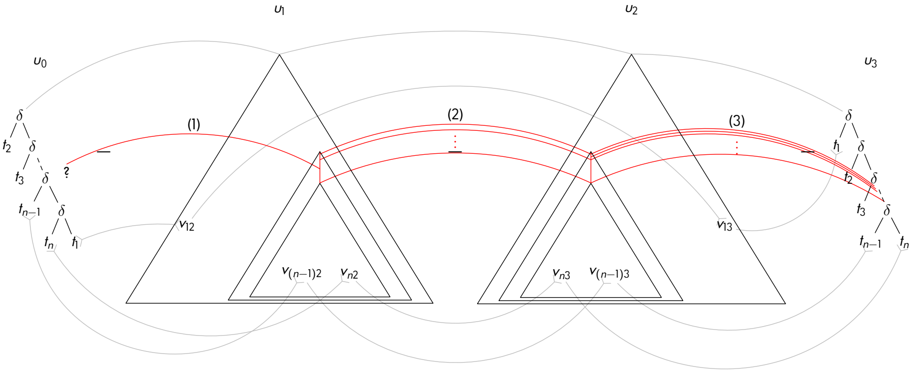
(string-level) range characterization by

[Gildea: On the string translations produced by multi bottom-up tree transducers. *Computational Linguistics* 38(3), 2012]

Synchronous Multi Tree Substitution Grammars

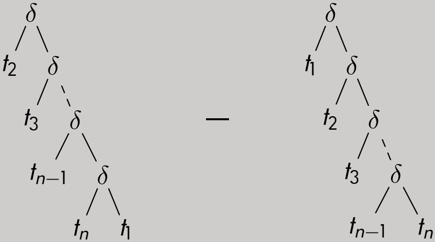
Theorem

$$\left(\mathrm{STSG}^{\mathrm{R}}\right)^3 \subsetneq \text{reg.-range SMTSG}$$



[M.: The power of weighted regularity-preserving multi bottom-up tree transducers. *Int. J. Found. Comput. Sci.* 26(7), 2015]

Counterexample relation



- abstracts a well-known linguistic transformation called topicalization
- implementable by SMTSG, but not by any composition of STSG

Synchronous Multi Tree Substitution Grammars

Illustration of topicalization

- It rained yesterday night.

  Topicalized: Yesterday night, it rained.

Synchronous Multi Tree Substitution Grammars

Illustration of topicalization

- It rained yesterday night.

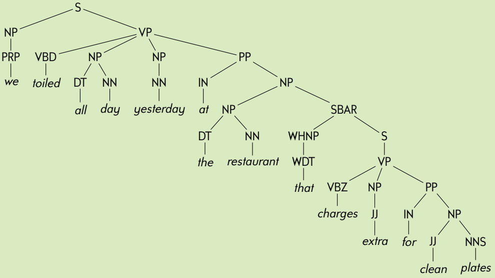
  Topicalized: Yesterday night, it rained.

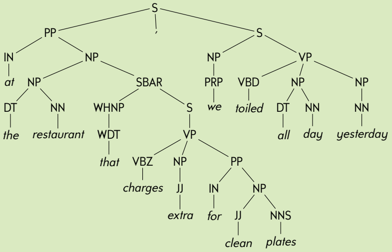
- We toiled all day yesterday
  at the restaurant that charges extra for clean plates.

  Topicalized: At the restaurant that charges extra for clean plates,
  we toiled all day yesterday.

Synchronous Multi Tree Substitution Grammars

On the tree level





Summary

Parsing

- weighted tree automata = CFG with subcategorization
- state-of-the-art models for many languages
- wealth of open problems for non-constituent parsing

Machine translation

- many grammar-based translation models in use
- expressive power ill-understood
- combination of parser grammar and translation model only understood for weighted tree automata
- “generalized” models might even work in practice

Thank you for the attention.