Tree Automata in Parsing and Machine Translation

Andreas Maletti

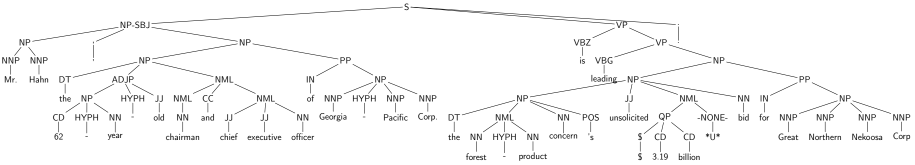
Institute of Computer Science
Universität Leipzig, Germany

Dagstuhl — April 4, 2017

Parsing

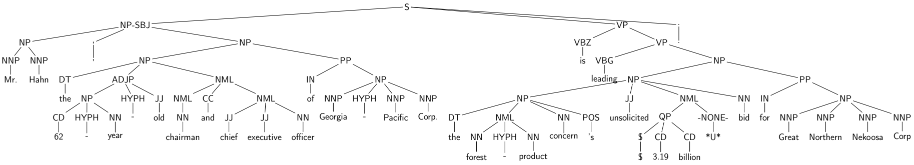
Parsing

- determining the syntactic structure of a sentence
- subject to a given theory of syntax (encoded in the training data)
    - constituent syntax
    - dependency syntax
    - . . .



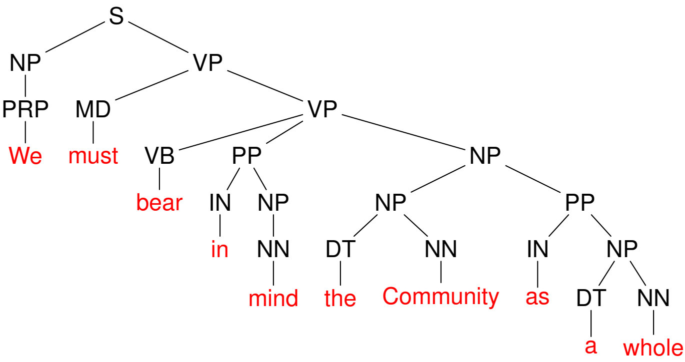
Parsing

- determining the syntactic structure of a sentence
- subject to a given theory of syntax (encoded in the training data)
  - constituent syntax
  - dependency syntax
  - ...



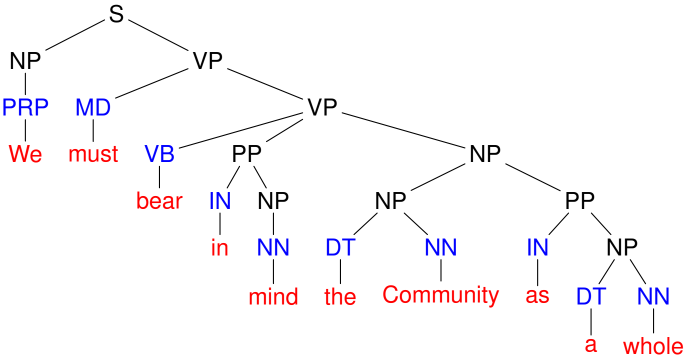
Constituent Parsing

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



Constituent Parsing

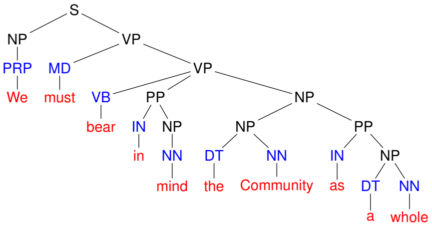
Example: We must bear in mind the Community as a whole



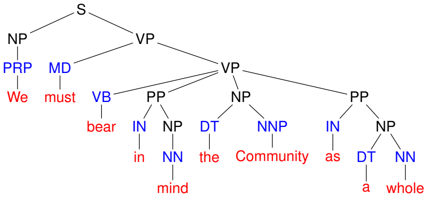
POS-tag: part-of-speech tag, “class” of a word

Constituent Parsing

Berkeley parser:



BLLIP parser:



Constituent Parsing

today

Linear-time dependency models; optimized by neural networks

2016

Subcategorization

manual: Collins (1999), Stanford (2003), BLLIP (2005)

automatic, e.g. Berkeley (2007)

2000

Statistical approach (cheap, automatically trained)

Penn and WSJ tree bank (1M and 30M words)

automatically obtained weighted CFG

1990

Chomskyan approach (perfect analysis, poor coverage)

hand-crafted CFG, TAG (refined via POS tags)

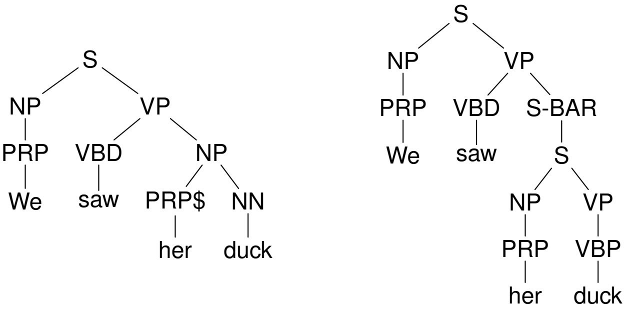
corrections and selection by human annotators

Constituent Parsing

grammar	F_1-score $\lvert w \rvert \leq 40$	full
CFG		62.7
TSG [Post, Gildea, 2009]	82.6	
TSG [Cohn et al., 2010]	85.4	84.7
$\mathrm{CFG}_{\mathrm{sub}}$ [Collins, 1999]	88.6	88.2
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, Klein, 2007]	90.6	90.1
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, 2010]		91.8
$\mathrm{TSG}_{\mathrm{sub}}$ [Shindo et al., 2012]	92.9	92.4

Constituent Parsing

All models use weights for disambiguation:

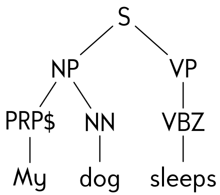


Subcategorization

Tags:

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

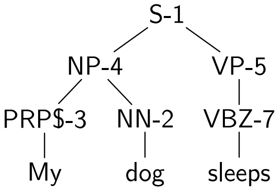
ADJA-Sup-Dat-Sg-Fem



Subcategorization

Tags:

- official tags often conservative
  - English: ≈ 50 tags
  - German: ≫ 200 tags ADJA-Sup-Dat-Sg-Fem
- all modern parsers use refined tags → subcategorization



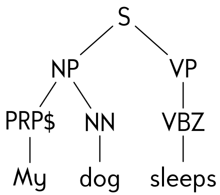
Subcategorization

Tags:

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

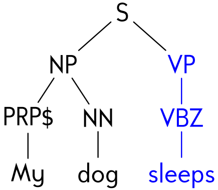
ADJA-Sup-Dat-Sg-Fem

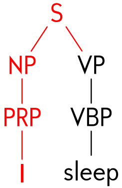
- all modern parsers use refined tags $\rightarrow$ subcategorization
- but return parse over official tags $\rightarrow$ relabeling



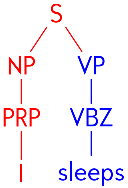
Subcategorization

These CFG derivations

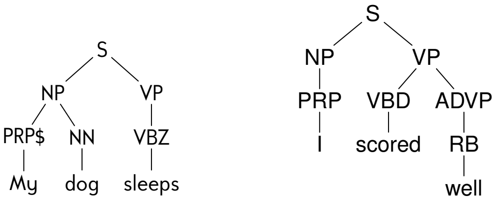




also admit



Constituent Parsing



Read off CFG productions:

$$
\begin{aligned}
S &\longrightarrow NP\ VP & NP &\longrightarrow PRP\$\ NN \\
PRP\$ &\longrightarrow My & NN &\longrightarrow dog \\
VP &\longrightarrow VBZ & VBZ &\longrightarrow sleeps \\
NP &\longrightarrow PRP & PRP &\longrightarrow I \\
VP &\longrightarrow VBD\ ADVP & VBD &\longrightarrow scored \\
ADVP &\longrightarrow RB & RB &\longrightarrow well
\end{aligned}
$$

Tree Automaton

Definition (Tree automaton)

Tuple (Q, Σ, I, R)

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

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

Example rules

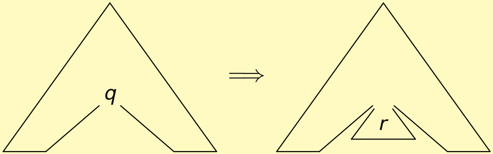
$q_4 \to \mathrm{VP}(q_5, q_2, q_3)$ $\qquad$ $q_0 \to \mathrm{S}(q_1, q_4)$ $\qquad$ $q_0 \to \mathrm{S}(q_6, q_2)$

Tree Automaton

Definition (Derivation semantics and recognized tree language)

Let (Q, Σ, I, R) tree automaton

- for each leaf position labeled q and rule $q \to r \in R$

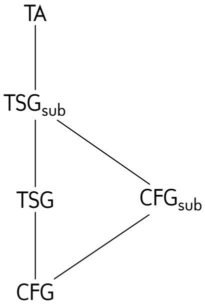


- recognized tree language

$$\{t \mid \exists q \in I \colon q \Rightarrow^* t\}$$

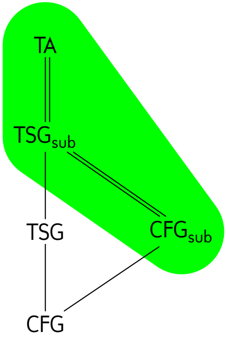
Constituent Parsing

grammar	F_1-score $\lvert w\rvert \leq 40$	F_1-score full
CFG		62.7
TSG [Post, Gildea, 2009]	82.6	
TSG [Cohn et al., 2010]	85.4	84.7
$\mathrm{CFG}_{\mathrm{sub}}$ [Collins, 1999]	88.6	88.2
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, Klein, 2007]	90.6	90.1
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, 2010]		91.8
$\mathrm{TSG}_{\mathrm{sub}}$ [Shindo et al., 2012]	92.9	92.4



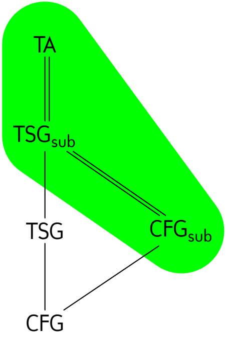
Constituent Parsing

grammar	F_1-score $\lvert w \rvert \leq 40$	F_1-score full
CFG		62.7
TSG [Post, Gildea, 2009]	82.6	
TSG [Cohn et al., 2010]	85.4	84.7
$\mathrm{CFG}_{\mathrm{sub}}$ [Collins, 1999]	88.6	88.2
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, Klein, 2007]	90.6	90.1
$\mathrm{CFG}_{\mathrm{sub}}$ [Petrov, 2010]		91.8
$\mathrm{TSG}_{\mathrm{sub}}$ [Shindo et al., 2012]	92.9	92.4



Constituent Parsing

grammar	F_1-score $\lvert w \rvert \leq 40$	full
CFG		62.7
TSG [Post, Gildea, 2009]	82.6	
TSG [Cohn et al., 2010]	85.4	84.7
CFG_{sub} [Collins, 1999]	88.6	88.2
CFG_{sub} [Petrov, Klein, 2007]	90.6	90.1
CFG_{sub} [Petrov, 2010]		91.8
TSG_{sub} [Shindo et al., 2012]	92.9	92.4



Hence:

- subcategorization = finite-state
- all modern models equivalent to tree automata in expressive power

Constituent Parsing

Comparison:

- rule of subcategorized CFG vs. corresponding rule of tree automaton

$$\text{S-1} \to \text{ADJP-2} \quad \text{S-1} \qquad\qquad \text{S-1} \to \text{S}\big(\text{ADJP-2},\ \text{S-1}\big)$$

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
- ...

Parsing

Parsing

- determining the syntactic structure of a sentence
- subject to a given theory of syntax (encoded in the training data)
  - constituent syntax
  - dependency syntax
  - . . .

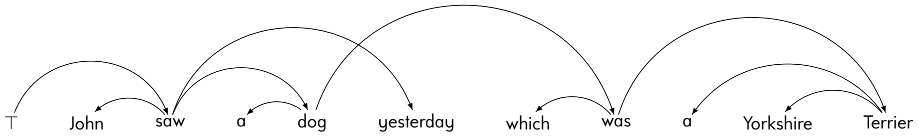
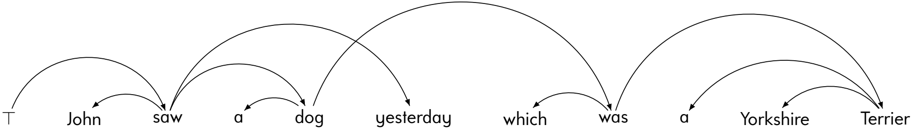
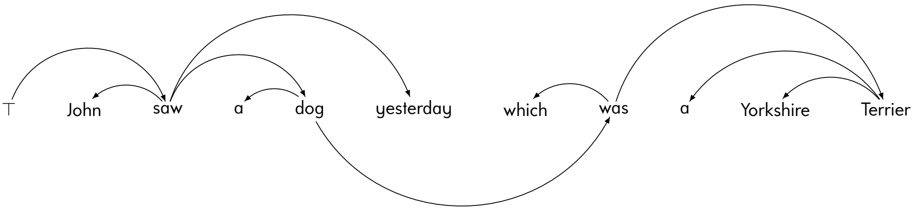


Illustration page-number:



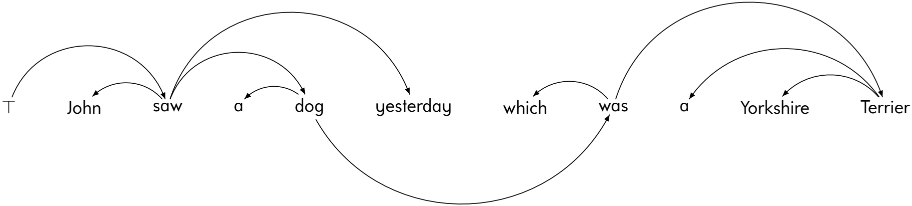
Dependency Parsing

Illustration page-number:



Dependency Parsing

Illustration page-number:



Practical results:

- linear-time statistical parsers
- Google's "Parsey McParseface" [Andor et al., 2016]
  94% F_1-score; linguists achieve 96–97%

Dependency Parsing

Theoretical 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

$$\dfrac{\dfrac{\dfrac{\begin{array}{c} c \\ \vdots \\ C \end{array} \quad \dfrac{\dfrac{\begin{array}{c} c \\ \vdots \\ C \end{array} \quad \begin{array}{c} d \\ \vdots \\ D/E/D\backslash C \end{array}}{D/E/D} \quad \begin{array}{c} d \\ \vdots \\ D/E\backslash C \end{array}}{D/E/E\backslash C}}{D/E/E} \quad \begin{array}{c} e \\ \vdots \\ E \end{array}}{D/E} \quad \begin{array}{c} e \\ \vdots \\ E \end{array}}{D}$$

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

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

Combinatory Categorial Grammars

Theoretical 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

ongoing work with Marco Kuhlmann

Lexicalization

Definition (lexicalized)

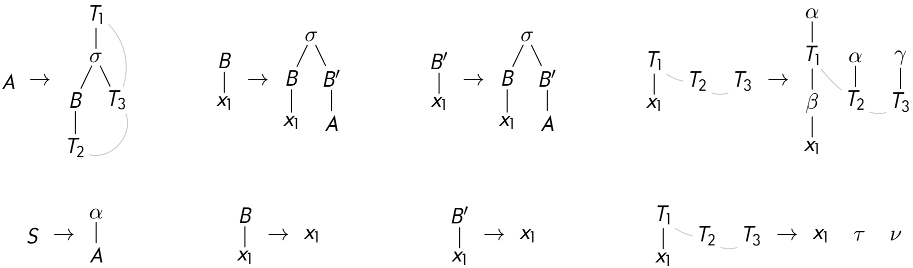
A grammar is lexicalized if each rule contains a lexical item

Existing results

- CFG weakly lexicalize themselves — Greibach normal form
- TAG weakly lexicalize themselves — [Schabes, 1990]
- TAG strongly lexicalize CFG and TSG — [Schabes, 1990]
- CFTG strongly lexicalize TAG and themselves — [M, Engelfriet, 2012]
- $(d+1)$-TAG strongly lexicalize d-TAG — [De Santo et al., 2016]

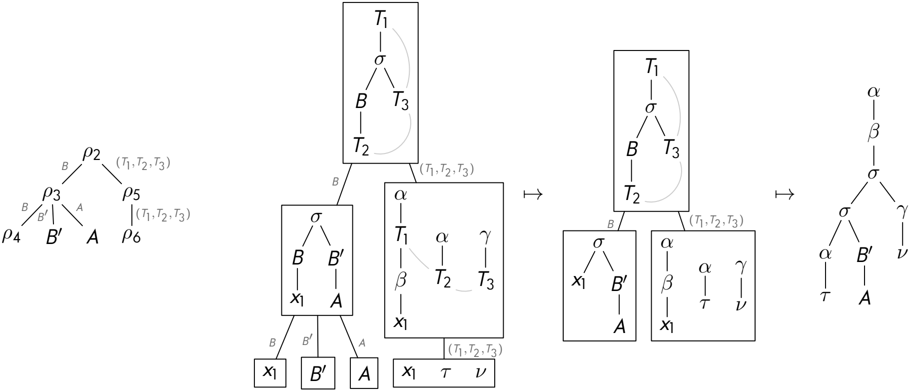
Lexicalization

Multiple context-free tree grammar:



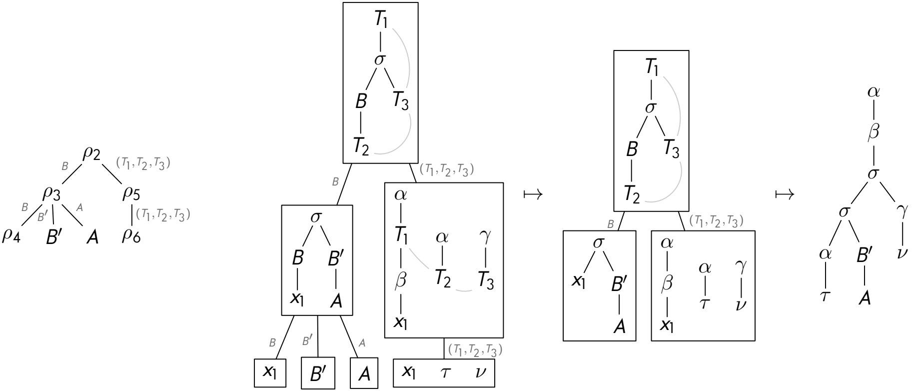
Lexicalization

Derivation tree and evaluation:



Lexicalization

Derivation tree and evaluation:



MCFTG strongly lexicalize themselves and inv. of their expressive power

ongoing work with Joost Engelfriet and Sebastian Maneth

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. 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. 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. ちょっと寒かったけれど、朝の散策はグッドでしたよ。

Machine Translation

Review translation [by Google Translate after 2016]

1. The room was simple, but it was not small, and the bathtub was also attached.
2. Wi-fi, TV was also available.
3. When I woke up in the morning and opened the curtain, I saw the church.
4. It was a bit cold, but walking in the morning 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. 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:

today

Neural networks

2016

Reformation

phrase-based and syntax-based systems

statistical approach (cheap, automatically trained)

1991

Dark age

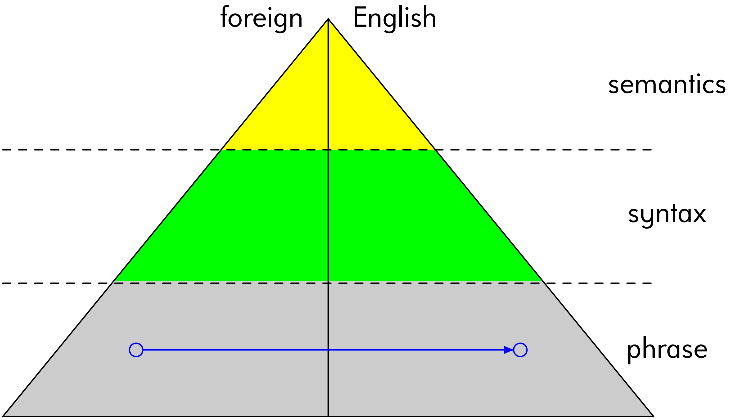
rule-based systems (e.g., SYSTRAN)

Chomskyan approach (perfect translation, poor coverage)

1960

Machine Translation

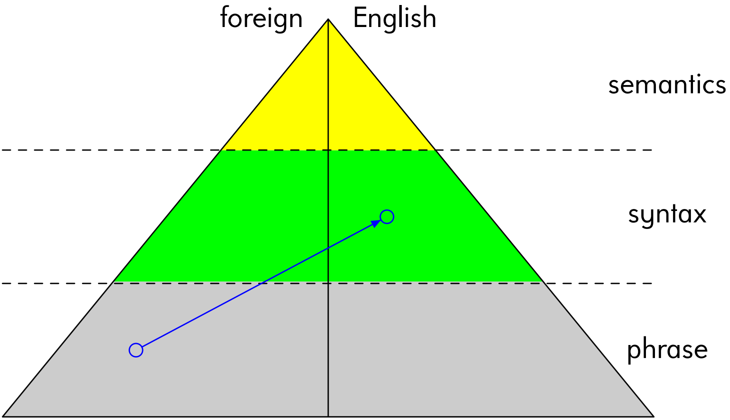
Vauquois triangle:



Translation model: string-to-string

Machine Translation

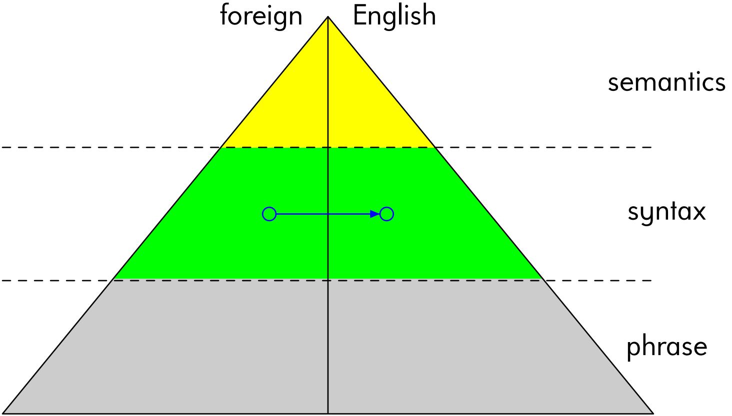
Vauquois triangle:



Translation model: string-to-tree

Machine Translation

Vauquois triangle:



Translation model: tree-to-tree

Machine Translation

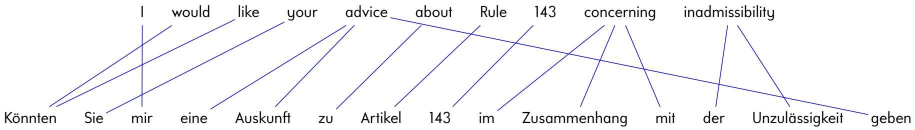
parallel corpus, word alignments, parse tree

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

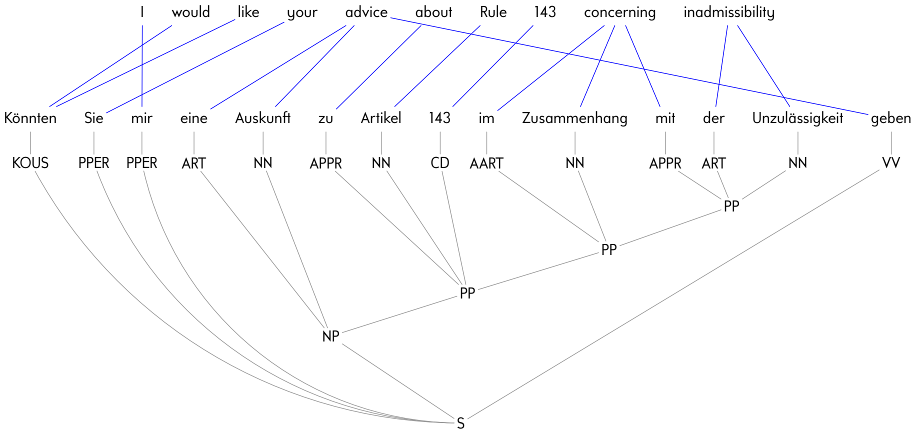
parallel corpus, word alignments, parse tree



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

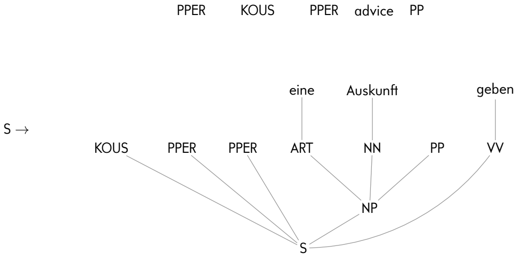


via Berkeley parser [Petrov, Barrett, Thibaux, Klein: Learning accurate, compact, and interpretable tree annotation. *Proc. ACL*, 2006]

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



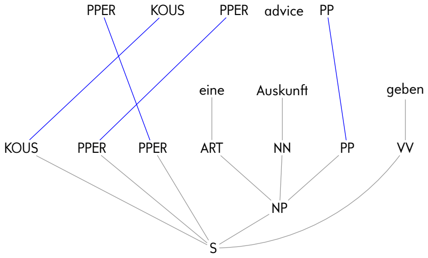
variant of [M., Graehl, Hopkins, Knight: The power of extended top-down tree transducers. *SIAM Journal on Computing* 39(2), 2009]

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

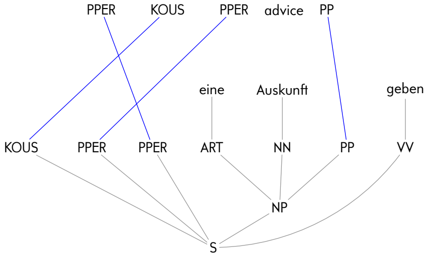
S →



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

Synchronous Grammars

S →

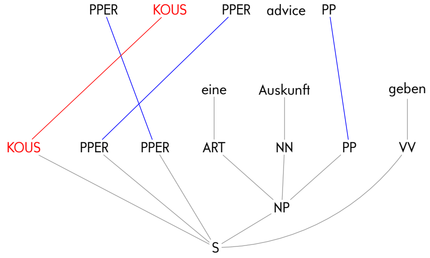


Production application:

1. Selection of synchronous nonterminals

Synchronous Grammars

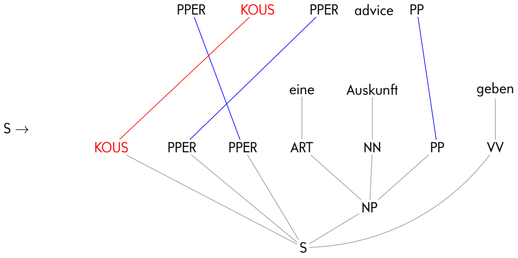
S →



Production application:

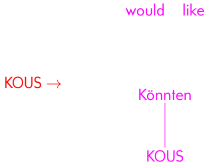
1. Selection of synchronous nonterminals

Synchronous Grammars

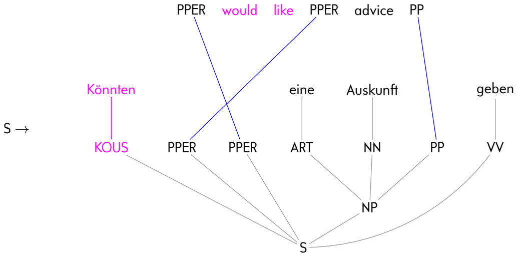


Production application:

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

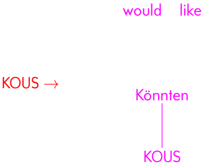


Synchronous Grammars

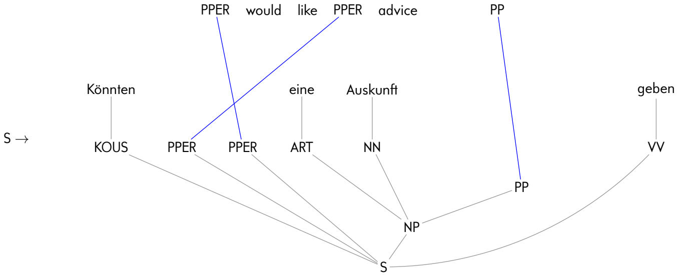


Production application:

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



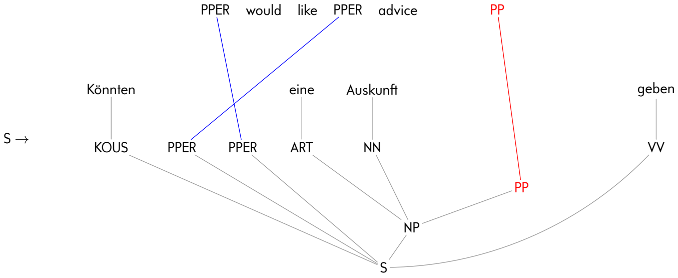
Synchronous Grammars



Production application:

1. synchronous nonterminals

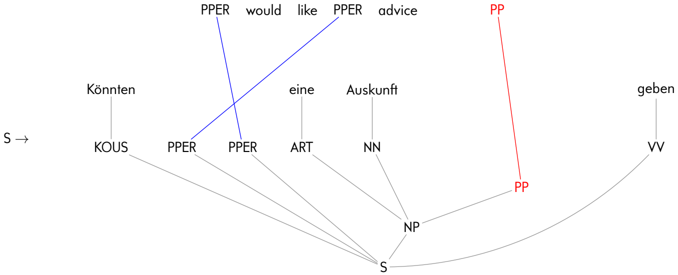
Synchronous Grammars



Production application:

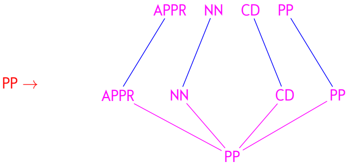
1. synchronous nonterminals

Synchronous Grammars



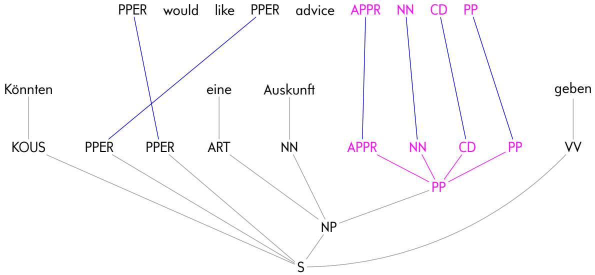
Production application:

1. synchronous nonterminals
2. suitable production



Synchronous Grammars

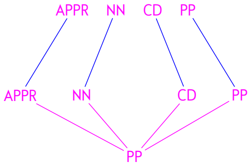
S →



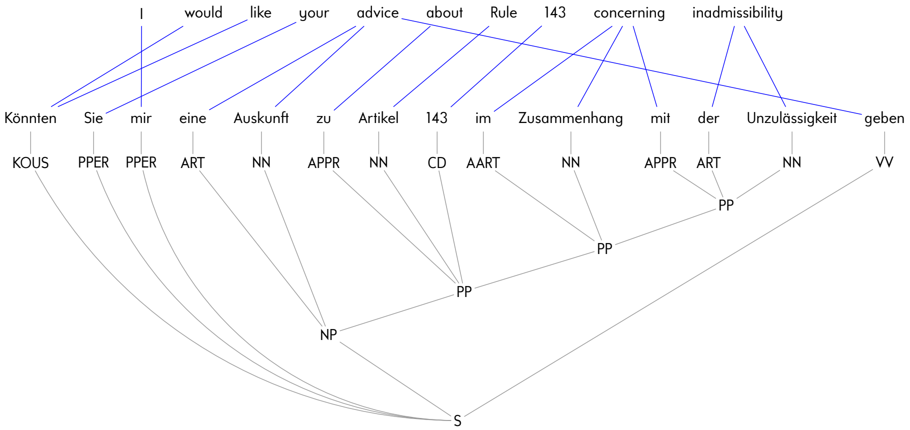
Production application:

1. synchronous nonterminals
2. suitable production
3. replacement

PP →



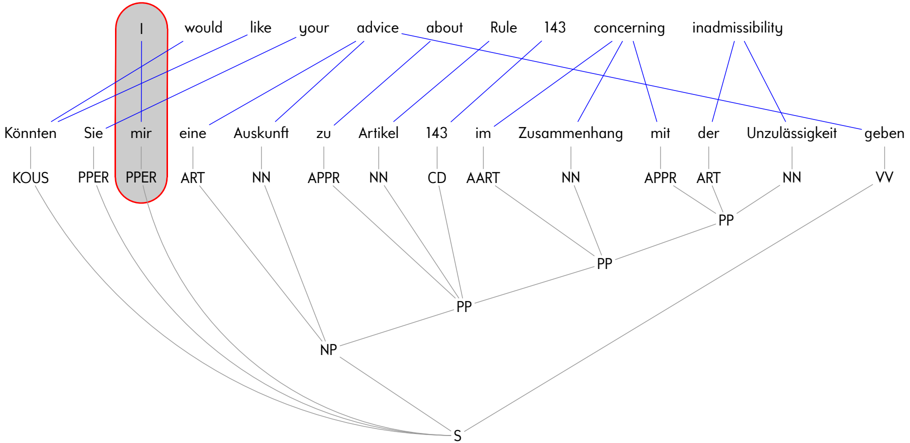
Production Extraction



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

Production Extraction

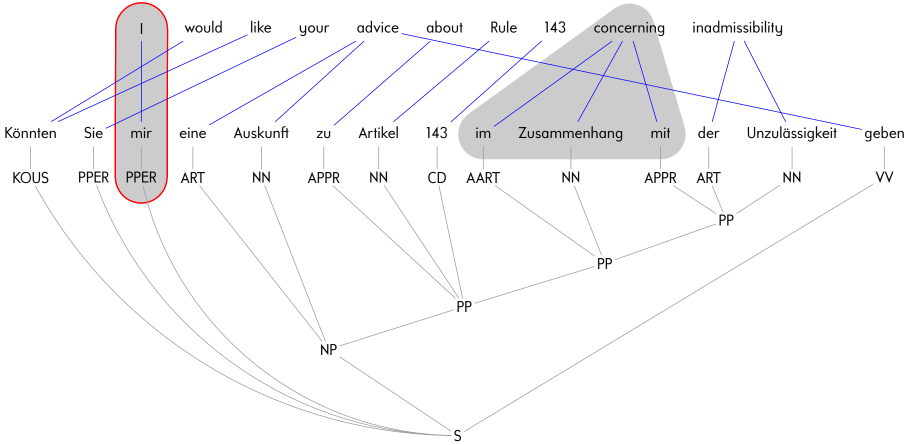
(extractable productions marked in red)



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

Production Extraction

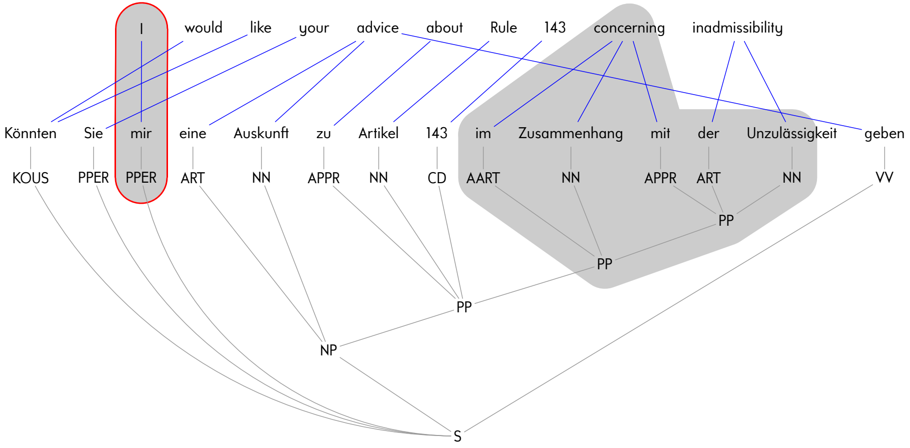
(extractable productions marked in red)



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

Production Extraction

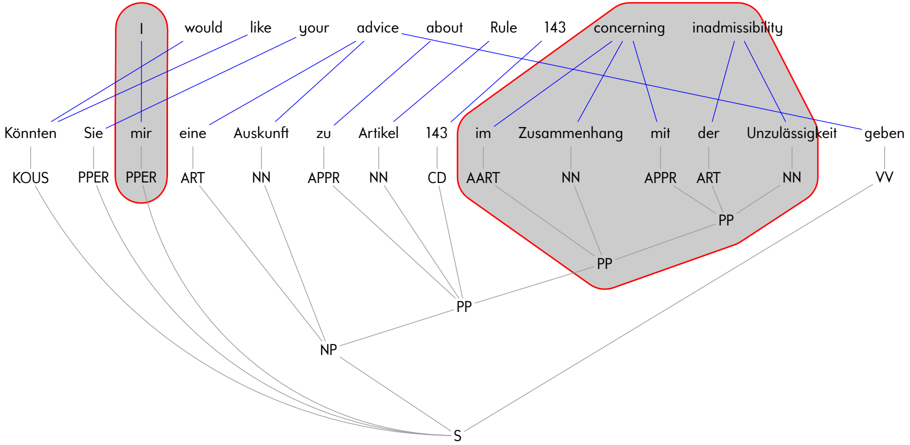
(extractable productions marked in red)



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

Production Extraction

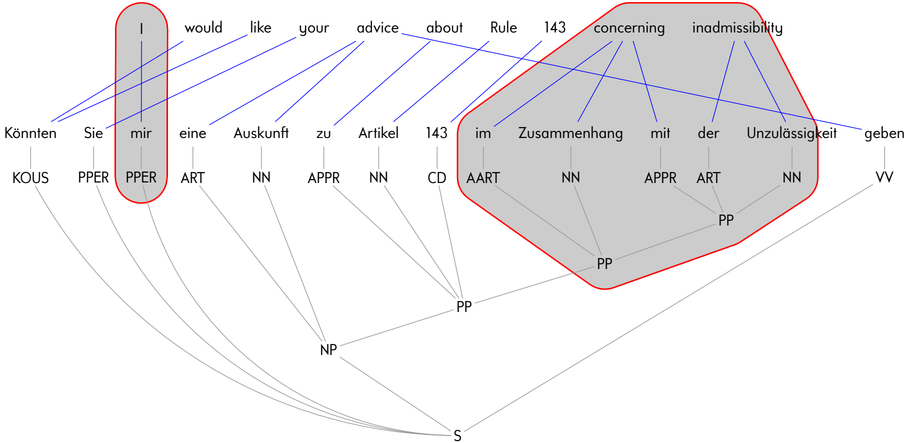
(extractable productions marked in red)



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

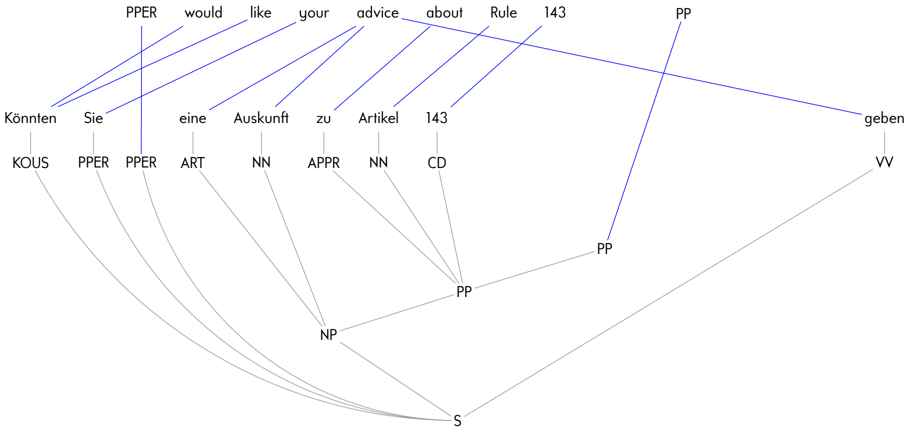
Production Extraction

Removal of extractable production:



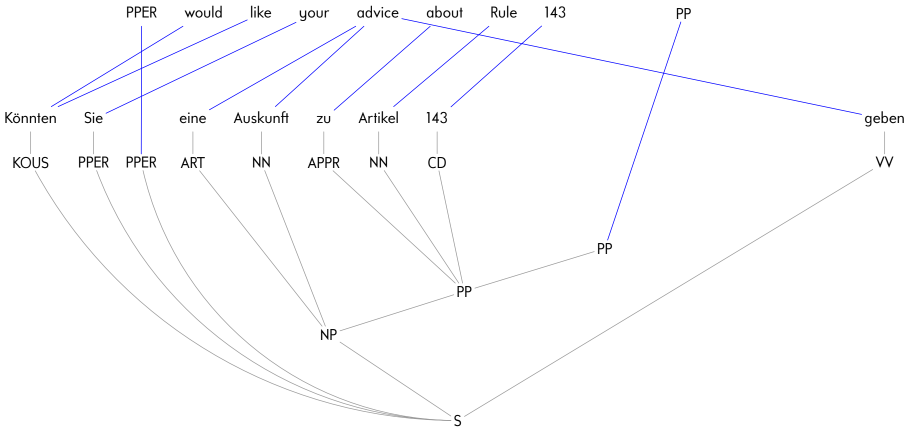
Production Extraction

Removal of extractable production:



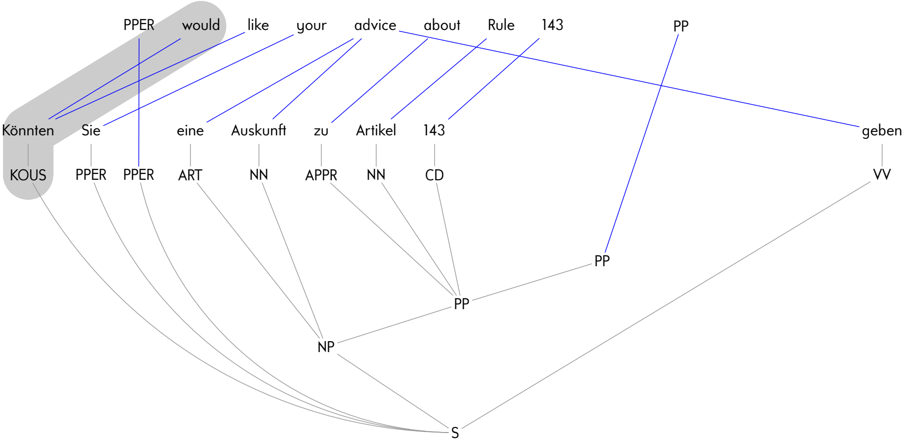
Production Extraction

Repeated production extraction:



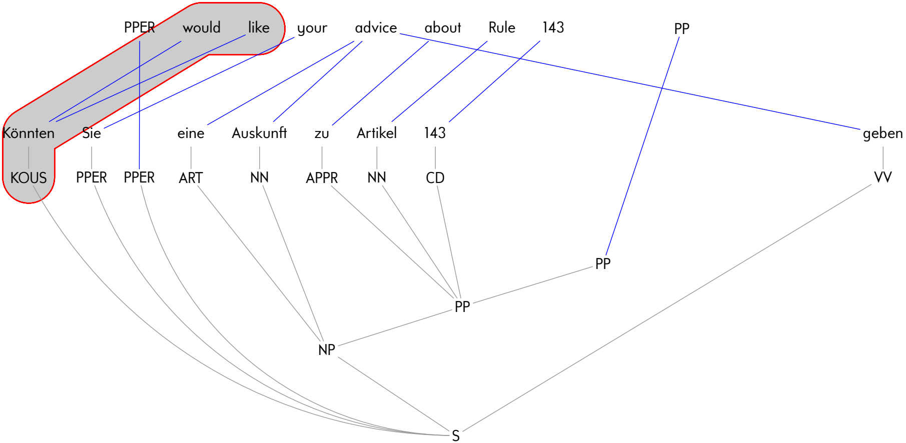
Production Extraction

Repeated production extraction:



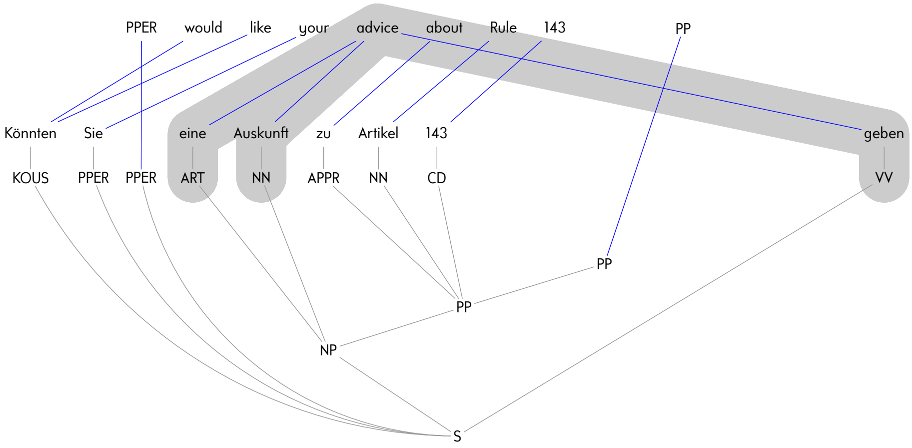
Production Extraction

Repeated production extraction: (extractable productions marked in red)



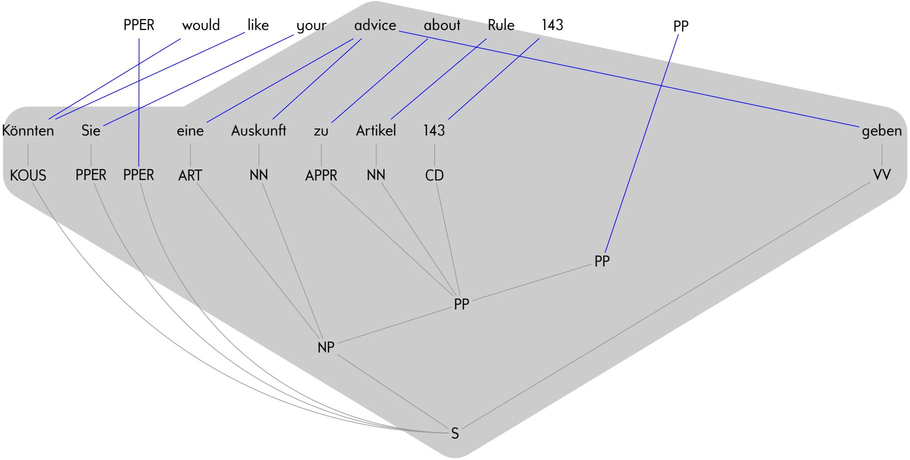
Production Extraction

Repeated production extraction: (extractable productions marked in red)



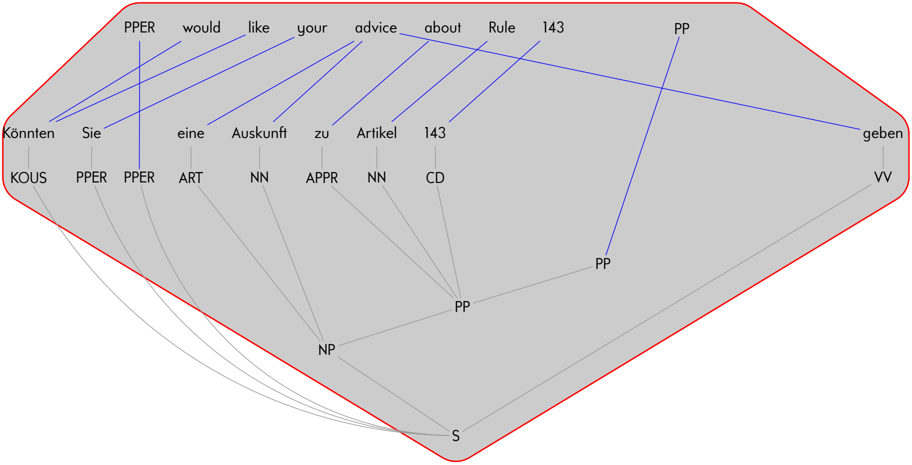
Production Extraction

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-string	FST	16.8	20.3	25.2
string-to-tree	STSG	15.2	19.4	24.5
tree-to-tree	STSG	14.5	—	15.3

STSG = synchronous tree substitution grammar

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]

Evaluation

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

Type	**System**	**BLEU**		
		vanilla	WMT 2013	WMT 2015
string-to-string	FST	16.8	20.3	25.2
string-to-tree	STSG	15.2	19.4	24.5
tree-to-tree	STSG	14.5	—	15.3

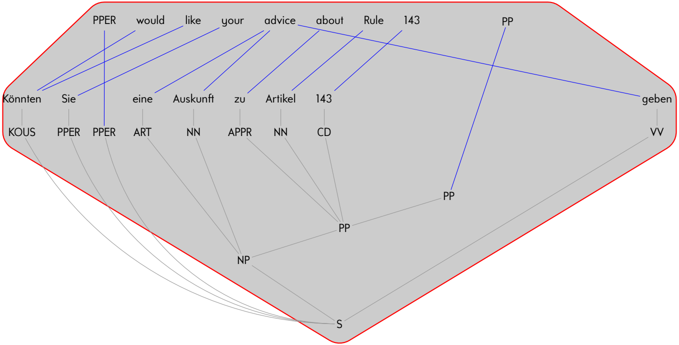
STSG = synchronous tree substitution grammar

Observations:

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

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]

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]

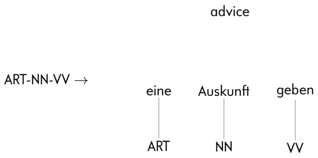
- nonterminal N
- right-hand side r of context-free grammar production
- right-hand sides $r_1, \ldots, r_n$ of regular tree grammar production

Synchronous Grammars

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

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

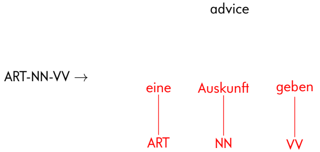


Synchronous Grammars

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

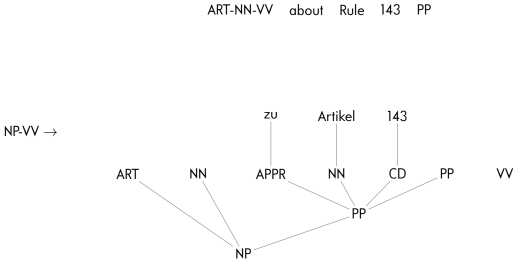


Synchronous Grammars

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

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 $r_1, \dots, r_n$ of regular tree grammar production

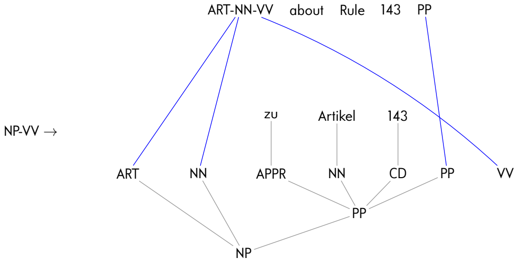


Synchronous Grammars

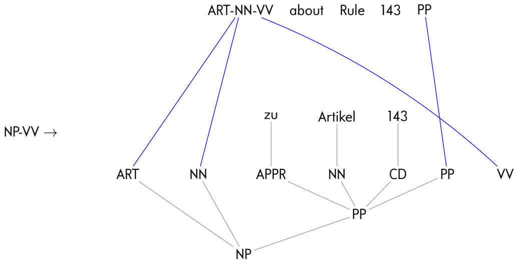
Synchronous multi tree substitution grammar: $N \to (r, \langle r_1, \ldots, r_n \rangle)$

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
- synchronization via map NT $r_1, \ldots, r_n$ to NT r



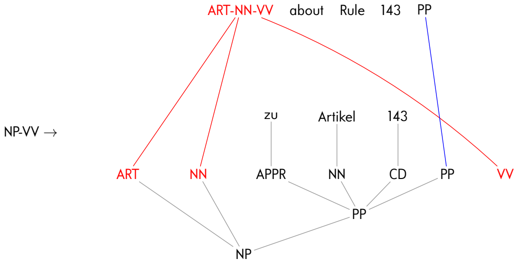
Synchronous Grammars



Production application:

1. synchronous nonterminals

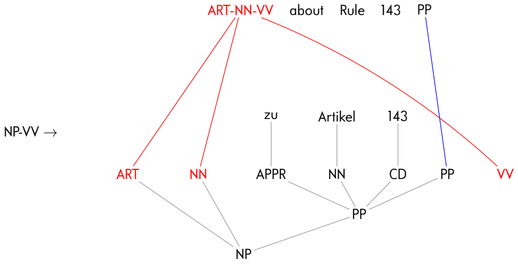
Synchronous Grammars



Production application:

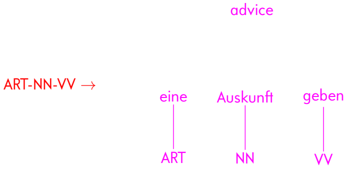
1. synchronous nonterminals

Synchronous Grammars

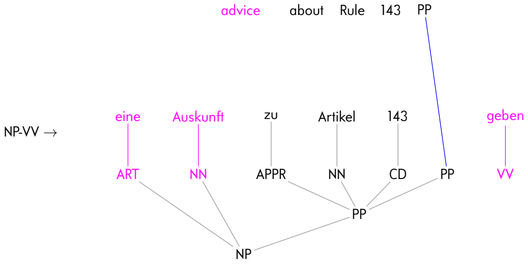


Production application:

1. synchronous nonterminals
2. suitable production

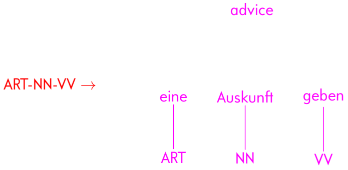


Synchronous Grammars

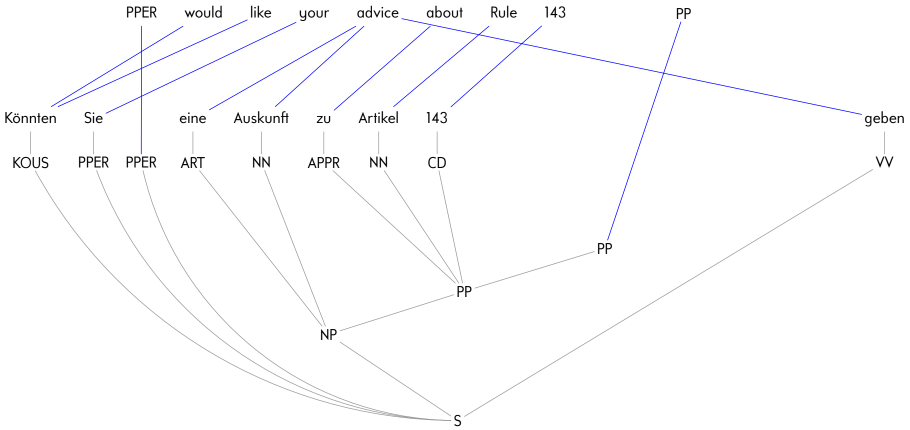


Production application:

1. synchronous nonterminals
2. suitable production
3. replacement

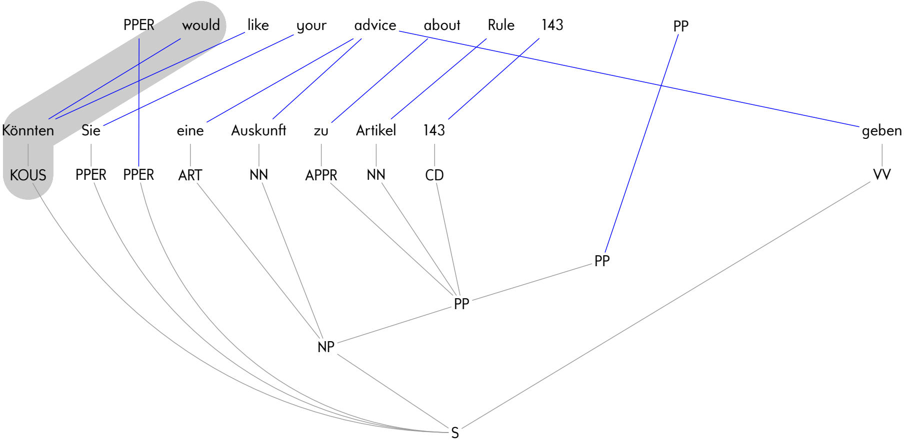


Production Extraction



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

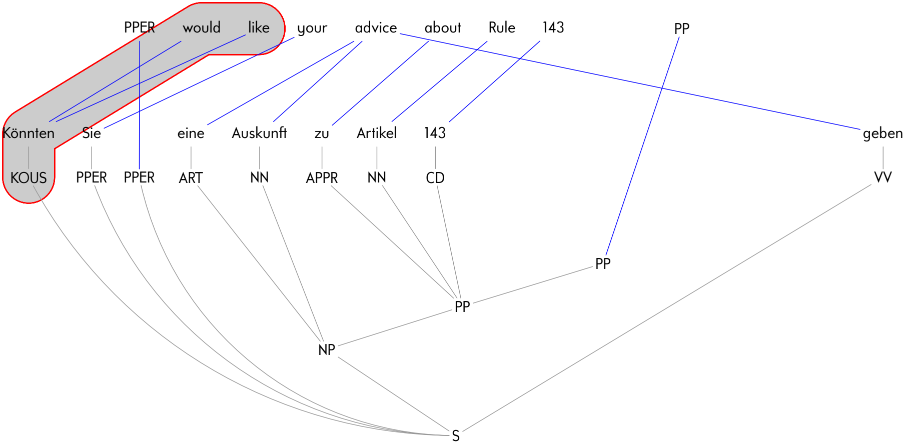
Production Extraction



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

Production Extraction

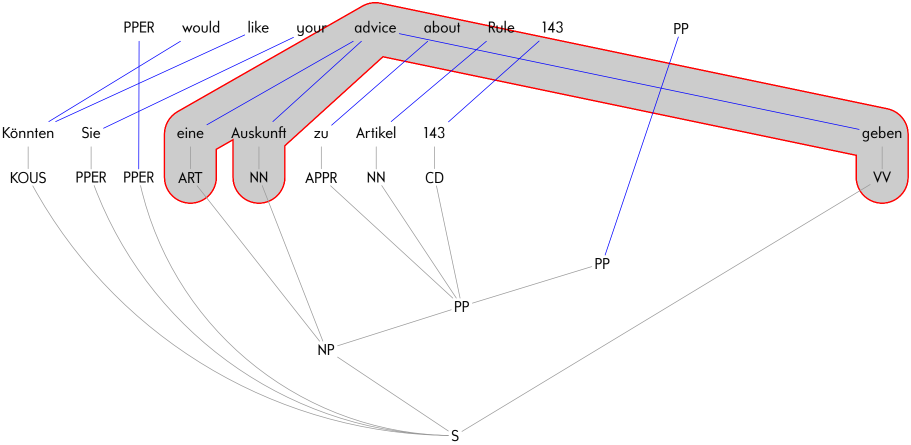
(extractable productions marked in red)



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 (tree-level) or non-context-free (string-level)
  (in fact output is captured by MRTG = MCFTG without variables)
- not symmetric (input context-free; output not)

Evaluation

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

STSG = synchronous tree substitution grammar
SMTSG = synchronous multi tree substitution grammar

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

Task	BLEU		Productions	
	STSG	SMTSG	STSG	SMTSG
English → German	15.0	*15.5	14M	144M
English → Arabic	48.2	*49.1	55M	491M
English → Chinese	17.7	*18.4	17M	162M
English → Polish	21.3	*23.4	—	—
English → Russian	24.7	*26.1	—	—

STSG = synchronous tree substitution grammar
SMTSG = synchronous multi tree substitution grammar

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

Task	BLEU		Productions	
	STSG	SMTSG	STSG	SMTSG
English → German	15.0	*15.5	14M	144M
English → Arabic	48.2	*49.1	55M	491M
English → Chinese	17.7	*18.4	17M	162M
English → Polish	21.3	*23.4	—	—
English → Russian	24.7	*26.1	—	—

STSG = synchronous tree substitution grammar

SMTSG = synchronous multi tree substitution grammar

Observations:

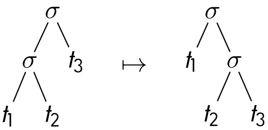
- consistent improvements
- 1 magnitude more productions
- SMTSG alleviate some of the problems of syntax-based systems

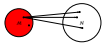
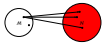
from [Seemann, Braune, M.: A systematic evaluation of MBOT in statistical machine translation. *Proc. MT-Summit*, 2015]
and [Seemann, M.: Discontinuous statistical machine translation with target-side dependency syntax. *Proc. WMT*, 2015]

Synchronous Grammars

Evaluation properties:

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

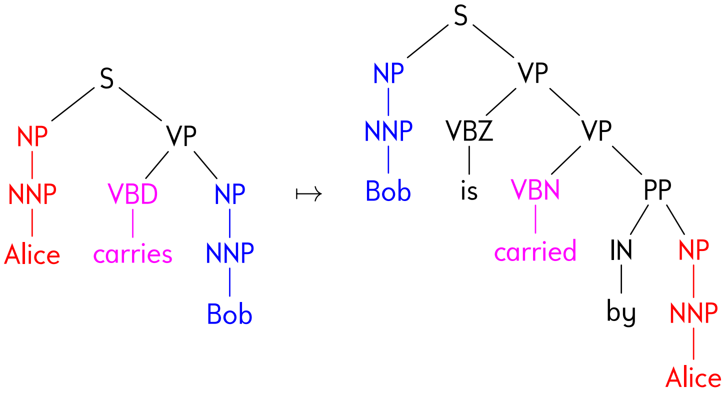


-  symmetric?
-  domain regular?
-  range regular?
- closed under composition?

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]

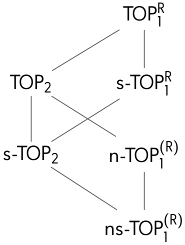
Icons by interactivemania (`http://www.interactivemania.com/`) and UN Office for the Coordination of Humanitarian Affairs

Illustration of rotation:

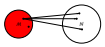
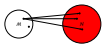


Top-down Tree Transducer

Hasse diagram:

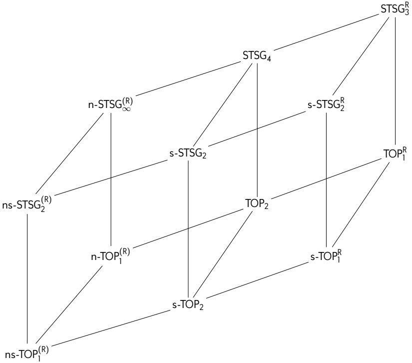


(composition closure in subscript)

Model	**Property**				
					
ns-TOP	✗	✗	✓	✓	✓
n-TOP	✗	✗	✓	✓	✓
s-TOP	✗	✗	✓	✓	✗$_2$
s-TOP^R	✗	✗	✓	✓	✓
TOP	✗	✗	✓	✓	✗$_2$
TOP^R	✗	✗	✓	✓	✓

Synchronous Tree Substitution Grammars

Hasse diagram:



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

(composition closure 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 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

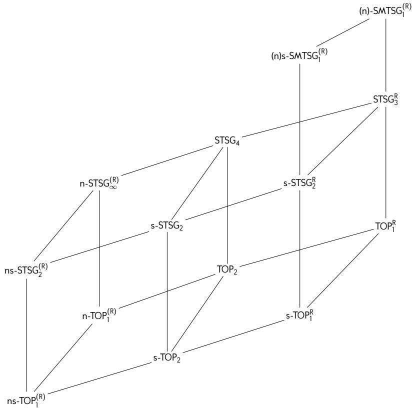
Disadvantages of SMTSG:

- non-regular range (theoretically interesting?)

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

Synchronous Multi Tree Substitution Grammars

Hasse diagram:



(composition closure in subscript)

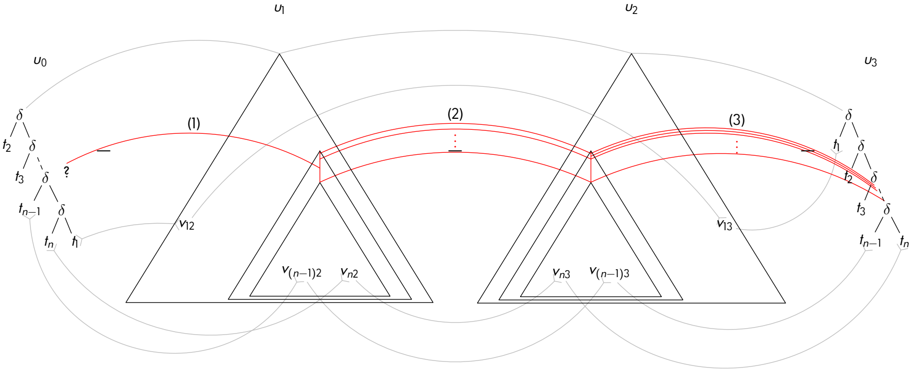
Model	Property				
n-TOP	✗	✗	✓	✓	✓
TOP	✗	✗	✓	✓	✗$_2$
TOP^R	✗	✗	✓	✓	✓
ns-STSG	✓	✓	✓	✓	✗$_2$
n-STSG	✓	✗	✓	✓	✗$_\infty$
s-STSG$^{(R)}$	✓	✗	✓	✓	✗$_2$
STSG	✓	✗	✓	✓	✗$_4$
STSG^R	✓	✗	✓	✓	✗$_3$
SMTSG	✓	✗	✓	✗	✓
reg. range	✓	✗	✓	✓	✓
symmetric	✓	✓	✓	✓	✓

(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

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



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

Summary

Parsing:

- tree automata = CFG with subcategorization
  (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing
  (alternative theories seem to be on the rise; "Parsey McParseface")

Summary

Parsing:

- tree automata = CFG with subcategorization
  (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing
  (alternative theories seem to be on the rise; “Parsey McParseface”)

Machine translation:

- all major translation models in use are grammar-based
  (and their expressive power is often ill-understood)
- combination of parser and translation model challenging
  (although that is typically just a regular domain restriction)
- evaluation of theoretically well-behaved models (in practice)

Summary

Parsing:

- tree automata = CFG with subcategorization
  (which are the state-of-the-art models for many languages)
- wealth of open problems for non-constituent parsing
  (alternative theories seem to be on the rise; "Parsey McParseface")

Machine translation:

- all major translation models in use are grammar-based
  (and their expressive power is often ill-understood)
- combination of parser and translation model challenging
  (although that is typically just a regular domain restriction)
- evaluation of theoretically well-behaved models (in practice)

Thank you for the attention.