Multi Bottom-up Tree Transducers

Andreas Maletti

Institute for Natural Language Processing
Universität Stuttgart, Germany

maletti@ims.uni-stuttgart.de

Avignon — April 24, 2012

Overview

Machine translation

Translation

- Input:
  Official forecasts predicted just 3 percent, Bloomberg said.
- We:
  die die offiziellen prognosen nur 3 prozent prognostizierten hat bloomberg gesagt.
- Google:
  Offizielle Prognosen vorhergesagt nur 3 Prozent, sagte Bloomberg.

Machine translation

Translation

- Input:
  The ECB wants to hold inflation to under two percent, or somewhere in that vicinity.
- We:
  die ezb will unter zwei inflation prozent oder halten irgendwo damit benachbarten gebieten,.
- Google:
  Die EZB will die Inflation auf unter zwei Prozent zu halten, oder irgendwo in der Nähe.

Syntax-based machine translation

Syntax-based systems

Input ⟶ Parser ⟶ Machine translation system ⟶ Language model ⟶ Output

What do we have?

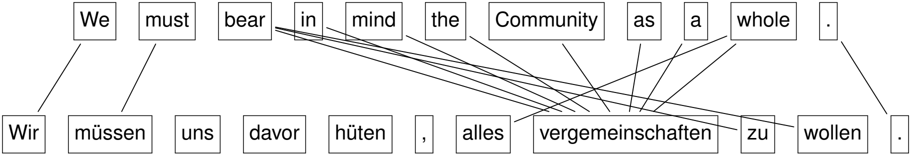
Input

- parallel text (English and German)
- here: EUROPARL

Example

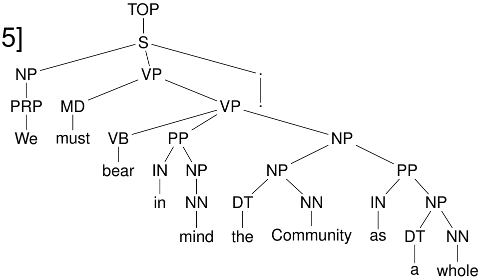
- “We must bear in mind the Community as a whole.”
- “Wir müssen uns davor hüten, alles vergemeinschaften zu wollen.”

Alignments by GIZA++ [OCH, NEY ’03]:

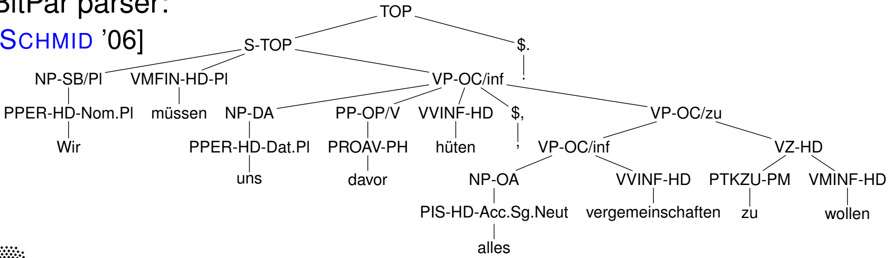


Parsing

CHARNIAK parser:
[CHARNIAK, JOHNSON '05]



BitPar parser:
[SCHMID '06]

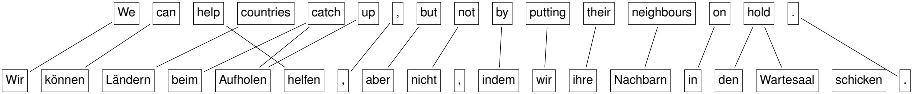


Better example

Example

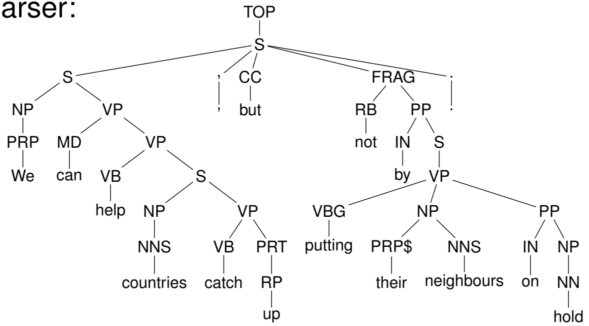
- “We can help countries catch up, but not by putting their neighbours on hold.”
- “Wir können Ländern beim Aufholen helfen, aber nicht, indem wir ihre Nachbarn in den Wartesaal schicken.”

Alignments by GIZA++:

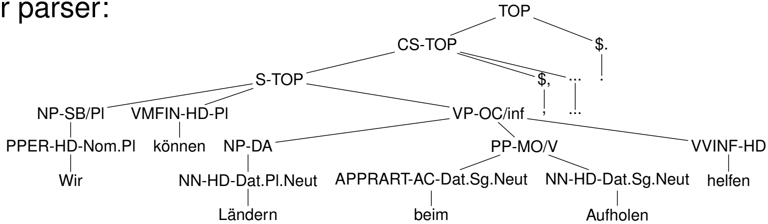


Better example

CHARNIAK parser:



BitPar parser:



Small example

Input

Yugoslav President Voislav signed for Serbia.

و تولى التوقيع عن صربيا الرئيس اليوغوسلافي فويسلاف

Transliteration: w twlY AltwqyE En SrbyA Alr}ys AlywgwslAfy fwyslAf.

And then the matter was decided, and everything was put in place.

ف كان ان تم الحسم و وضعت الأمور في نصاب ها

Transliteration: f kAn An tm AlHsm w wDEt Al>mwr fy nSAb hA.

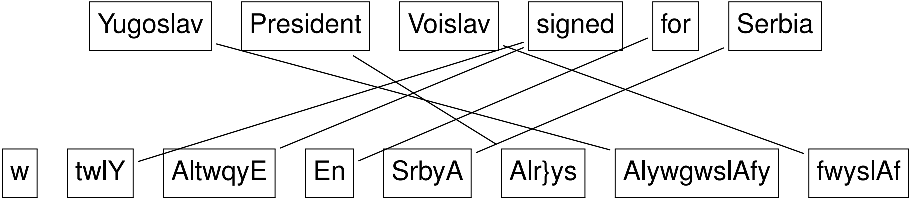
Below are the male and female winners in the different categories.

و هنا الأوائل و الأوليات في مختلف الفئات

Transliteration: w hnA Al>wA}l w Al>wlyAt fy mxtlf Alf}At.

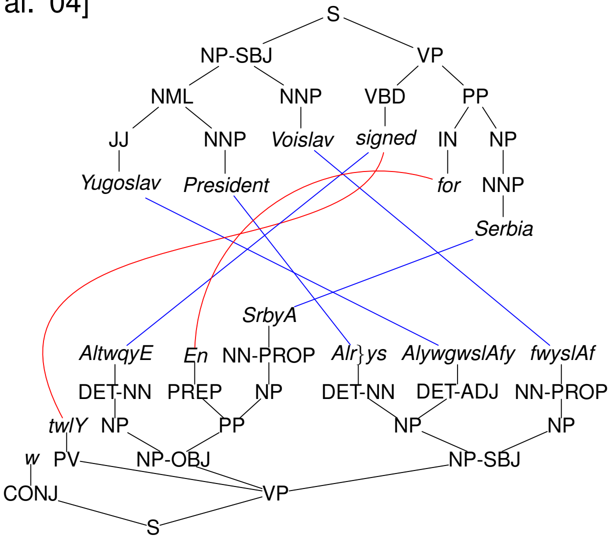
Small example

Alignment

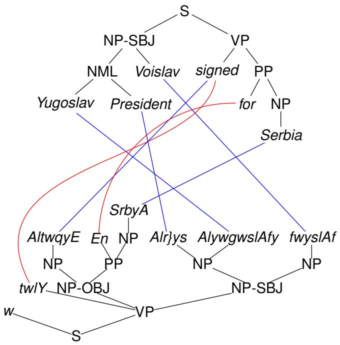


Rule extraction

[GALLEY et al. '04]

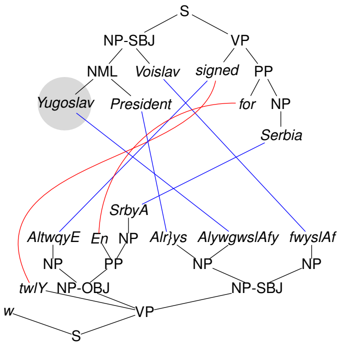


Rule extraction



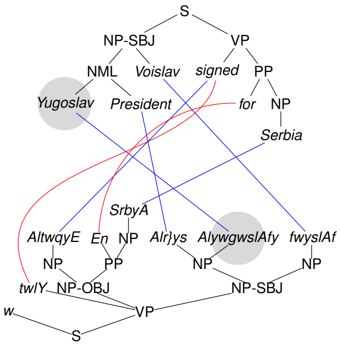
- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

Rule extraction



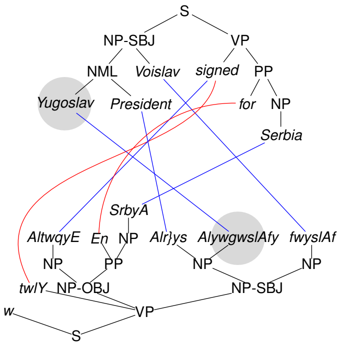
- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

Rule extraction



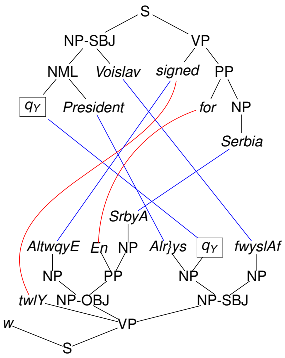
- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

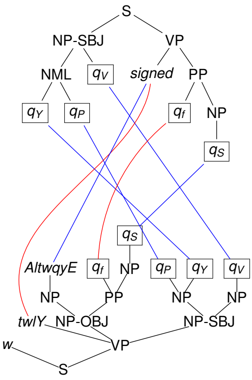
Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

$$Yugoslav \xrightarrow{q_Y} AlywgwslAfy$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

Yugoslav $\underline{\;q_Y\;}$ *AlywgwslAfy*

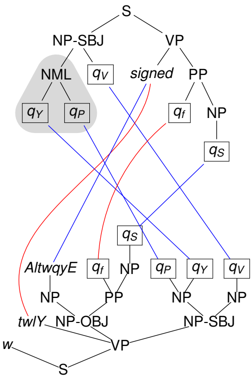
President $\underline{\;q_P\;}$ *Alr}ys*

Voislav $\underline{\;q_V\;}$ *fwyslAf*

for $\underline{\;q_f\;}$ *En*

Serbia $\underline{\;q_S\;}$ *SrbyA*

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

$$\textit{Yugoslav} \xrightarrow{q_Y} \textit{AlywgwslAfy}$$

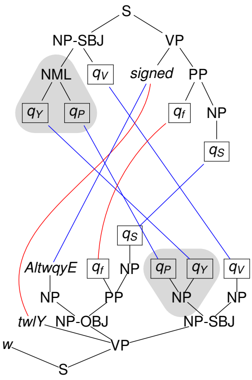
$$\textit{President} \xrightarrow{q_P} \textit{Alr\}ys}$$

$$\textit{Voislav} \xrightarrow{q_V} \textit{fwyslAf}$$

$$\textit{for} \xrightarrow{q_f} \textit{En}$$

$$\textit{Serbia} \xrightarrow{q_S} \textit{SrbyA}$$

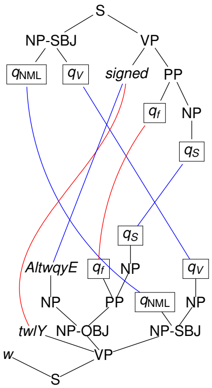
Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

$$\text{NML}(q_Y, q_P) \xrightarrow{q_{\text{NML}}} \text{NP}(q_P, q_Y)$$

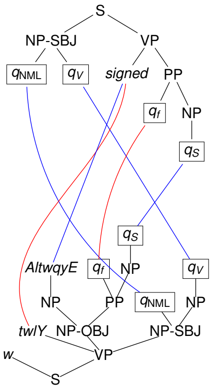
Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

$$\text{NML}(q_Y, q_P) \xrightarrow{q_{\text{NML}}} \text{NP}(q_P, q_Y)$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

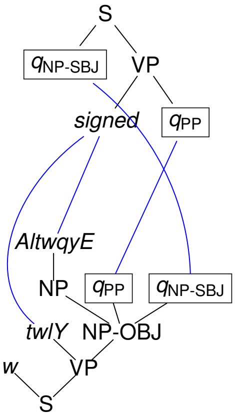
$$\mathrm{NML}(q_Y, q_P) \xrightarrow{q_{\mathrm{NML}}} \mathrm{NP}(q_P, q_Y)$$

$$\mathrm{NP}(q_S) \xrightarrow{q_{\mathrm{NP}}} \mathrm{NP}(q_S)$$

$$\mathrm{PP}(q_f, q_{\mathrm{NP}}) \xrightarrow{q_{\mathrm{PP}}} \mathrm{PP}(q_f, q_{\mathrm{NP}})$$

$$\mathrm{NP\text{-}SBJ}(q_{\mathrm{NML}}, q_V) \xrightarrow{q_{\mathrm{NP\text{-}SBJ}}} \mathrm{NP\text{-}SBJ}(q_{\mathrm{NML}}, \mathrm{NP}(q_V))$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

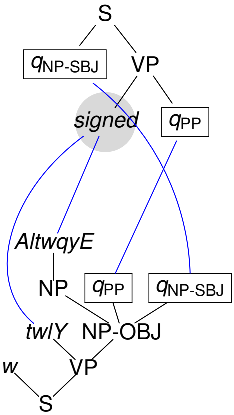
$$\text{NML}(q_Y, q_P) \xrightarrow{q_{\text{NML}}} \text{NP}(q_P, q_Y)$$

$$\text{NP}(q_S) \xrightarrow{q_{\text{NP}}} \text{NP}(q_S)$$

$$\text{PP}(q_f, q_{\text{NP}}) \xrightarrow{q_{\text{PP}}} \text{PP}(q_f, q_{\text{NP}})$$

$$\text{NP-SBJ}(q_{\text{NML}}, q_V) \xrightarrow{q_{\text{NP-SBJ}}} \text{NP-SBJ}(q_{\text{NML}}, \text{NP}(q_V))$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

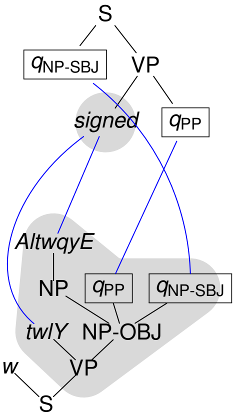
$$\text{NML}(q_Y, q_P) \stackrel{q_{\text{NML}}}{\longrightarrow} \text{NP}(q_P, q_Y)$$

$$\text{NP}(q_S) \stackrel{q_{\text{NP}}}{\longrightarrow} \text{NP}(q_S)$$

$$\text{PP}(q_f, q_{\text{NP}}) \stackrel{q_{\text{PP}}}{\longrightarrow} \text{PP}(q_f, q_{\text{NP}})$$

$$\text{NP-SBJ}(q_{\text{NML}}, q_V) \stackrel{q_{\text{NP-SBJ}}}{\longrightarrow} \text{NP-SBJ}(q_{\text{NML}}, \text{NP}(q_V))$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

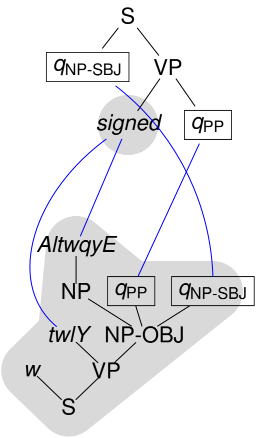
$$\text{NML}(q_Y, q_P) \xrightarrow{q_{\text{NML}}} \text{NP}(q_P, q_Y)$$

$$\text{NP}(q_S) \xrightarrow{q_{\text{NP}}} \text{NP}(q_S)$$

$$\text{PP}(q_f, q_{\text{NP}}) \xrightarrow{q_{\text{PP}}} \text{PP}(q_f, q_{\text{NP}})$$

$$\text{NP-SBJ}(q_{\text{NML}}, q_V) \xrightarrow{q_{\text{NP-SBJ}}} \text{NP-SBJ}(q_{\text{NML}}, \text{NP}(q_V))$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

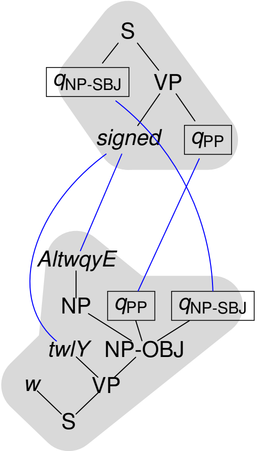
$$\text{NML}(q_Y, q_P) \xrightarrow{q_{\text{NML}}} \text{NP}(q_P, q_Y)$$

$$\text{NP}(q_S) \xrightarrow{q_{\text{NP}}} \text{NP}(q_S)$$

$$\text{PP}(q_f, q_{\text{NP}}) \xrightarrow{q_{\text{PP}}} \text{PP}(q_f, q_{\text{NP}})$$

$$\text{NP-SBJ}(q_{\text{NML}}, q_V) \xrightarrow{q_{\text{NP-SBJ}}} \text{NP-SBJ}(q_{\text{NML}}, \text{NP}(q_V))$$

Rule extraction



- Select next node bottom-up
- Identify maximal subtree of aligned nodes
- Identify subtree of nodes aligned to aligned nodes, etc.
- Extract rule and leave state
- Repeat

$$\text{NML}(q_Y, q_P) \stackrel{q_{\text{NML}}}{\longrightarrow} \text{NP}(q_P, q_Y)$$

$$\text{NP}(q_S) \stackrel{q_{\text{NP}}}{\longrightarrow} \text{NP}(q_S)$$

$$\text{PP}(q_f, q_{\text{NP}}) \stackrel{q_{\text{PP}}}{\longrightarrow} \text{PP}(q_f, q_{\text{NP}})$$

$$\text{NP-SBJ}(q_{\text{NML}}, q_V) \stackrel{q_{\text{NP-SBJ}}}{\longrightarrow} \text{NP-SBJ}(q_{\text{NML}}, \text{NP}(q_V))$$

Extended top-down tree transducer

Advantages

- ✓ simple and natural model
- ✓ easy to train (from linguistic resources) [GRAEHL et al. '08]
- ✓ symmetric

Implementation

- TIBURON [MAY, KNIGHT '06]

Extended top-down tree transducer

Advantages

- ✓ simple and natural model
- ✓ easy to train (from linguistic resources) [GRAEHL et al. '08]
- ✓ symmetric

Implementation

- TIBURON [MAY, KNIGHT '06]

Extended top-down tree transducer

Disadvantages (also of STSG)

- ✗ no discontinuities
- ✗ not binarizable
  [AHO, ULLMAN '72; ZHANG et al. '06]
- ✗ inefficient input/output restriction
  [M., SATTA '10]
- ✗ not composable
  [ARNOLD, DAUCHET '82]

Roadmap

Syntax

Definition

Extended multi bottom-up tree transducer (XMBOT)
system (Q, Σ, F, R)

- Q *ranked* alphabet (*states*)
- Σ ranked alphabet (input/output symbols)
- $F \subseteq Q_1$ (final states)
- R finite set of rules $\ell \to r$ (rules)
  - linear $\ell \in T_\Sigma(Q(X))$
  - $r \in Q(T_\Sigma(Y))$ with $Y = \text{var}(\ell)$

Definition

- linear if r is linear for all $\ell \to r \in R$
- nondeleting if $\text{var}(r) = \text{var}(\ell)$ for all $\ell \to r \in R$

Syntax

Definition

Extended multi bottom-up tree transducer (XMBOT)
system (Q, Σ, F, R)

- Q *ranked* alphabet (*states*)
- Σ ranked alphabet (input/output symbols)
- $F \subseteq Q_1$ (final states)
- R finite set of rules $\ell \to r$ (rules)
    - linear $\ell \in T_\Sigma(Q(X))$
    - $r \in Q(T_\Sigma(Y))$ with $Y = \text{var}(\ell)$

Definition

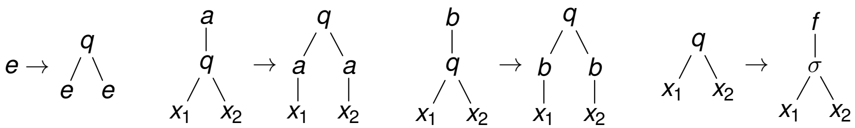
- linear if r is linear for all $\ell \to r \in R$
- nondeleting if $\text{var}(r) = \text{var}(\ell)$ for all $\ell \to r \in R$

Syntax

Example

XMBOT $(Q, \Sigma, \{f\}, R)$

- $Q = \{q^{(2)}, f^{(1)}\}$
- $\Sigma = \{\sigma^{(2)}, a^{(1)}, b^{(1)}, e^{(0)}\}$
- R contains:



Note

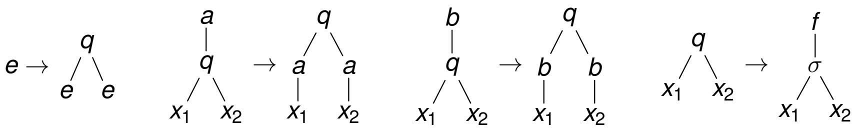
It is linear and nondeleting

Syntax

Example

XMBOT $(Q, \Sigma, \{f\}, R)$

- $Q = \{q^{(2)}, f^{(1)}\}$
- $\Sigma = \{\sigma^{(2)}, a^{(1)}, b^{(1)}, e^{(0)}\}$
- R contains:

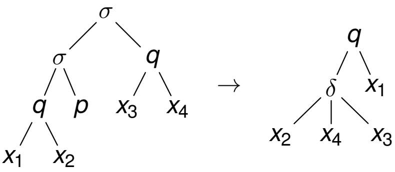


Note

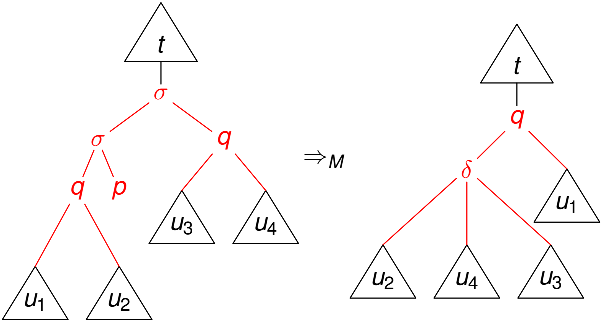
It is linear and nondeleting

Semantics

Rule:



Derivation:



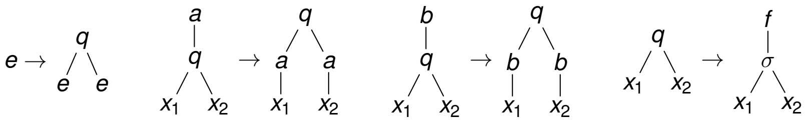
Semantics

$$e \to q(e, e) \qquad a(q(x_1, x_2)) \to q(a(x_1), a(x_2)) \qquad b(q(x_1, x_2)) \to q(b(x_1), b(x_2)) \qquad q(x_1, x_2) \to f(\sigma(x_1, x_2))$$

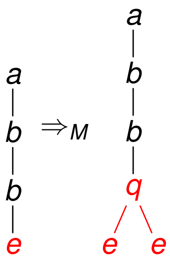
Example (Derivation)

$$a(b(b(e)))$$

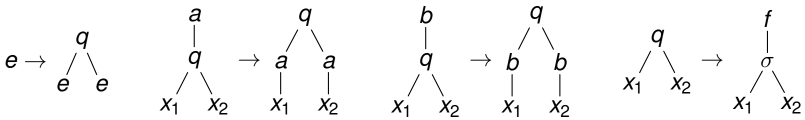
Semantics



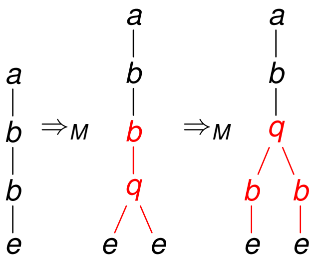
Example (Derivation)



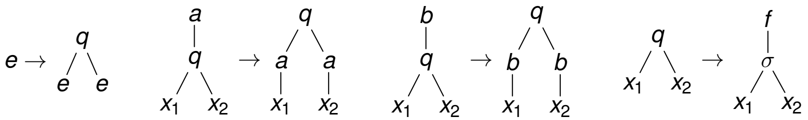
Semantics



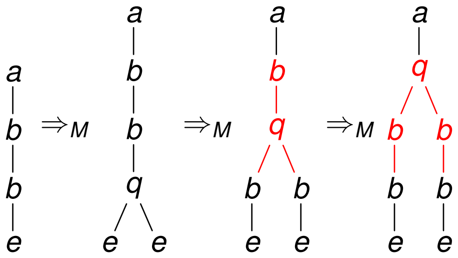
Example (Derivation)



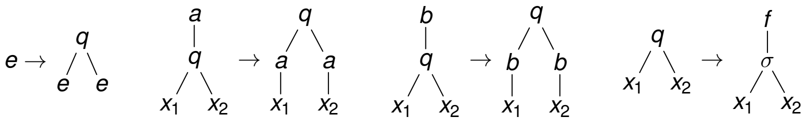
Semantics



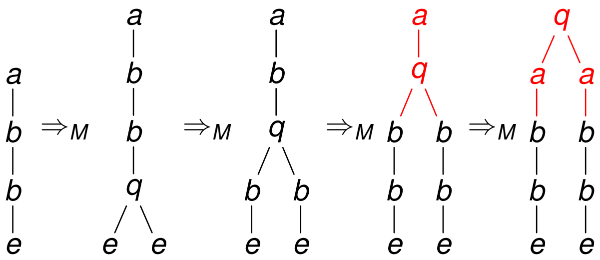
Example (Derivation)



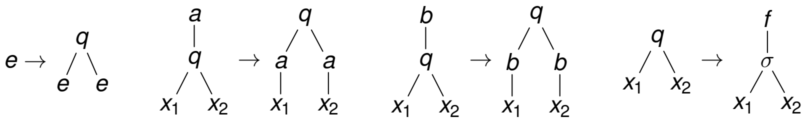
Semantics



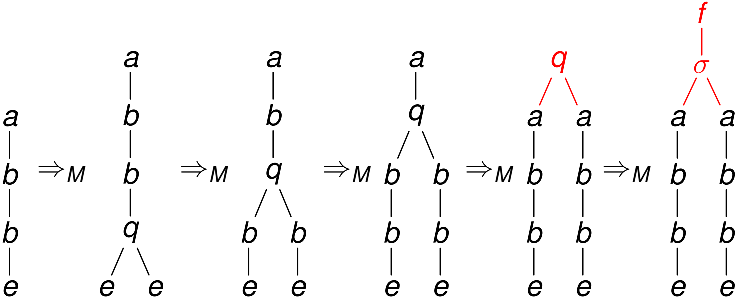
Example (Derivation)



Semantics



Example (Derivation)



Semantics

Definition

XMBOT $M = (Q, \Sigma, F, R)$

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Sigma \mid \exists q \in F \colon t \Rightarrow_M^* q(u)\}$$

Semantics

Definition

XMBOT $M = (Q, \Sigma, F, R)$

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Sigma \mid \exists q \in F \colon t \Rightarrow_M^* q(u)\}$$

Example

It computes $\{(t, \sigma(t, t)) \mid t \in T_\Sigma\}$

Its image is not recognizable

Restrictions

Definition

XMBOT (Q, Σ, F, R) is

- XBOT if $Q = Q_1$
- MBOT if $\ell \in \Sigma(Q(X))$ for all $\ell \to r \in R$

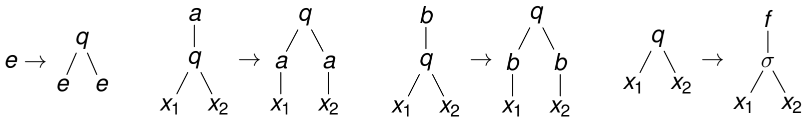
Restrictions

Definition

XMBOT (Q, Σ, F, R) is

- XBOT if $Q = Q_1$
- MBOT if $\ell \in \Sigma(Q(X))$ for all $\ell \to r \in R$

Example



It is neither XBOT nor MBOT

Table of Contents

Proper generalization

Theorem (Engelfriet et al. '09)

All linear XTOP can be simulated by linear XBOT

Proof.

Standard construction trading input-deletion for output-deletion
see l-TOP $\subseteq$ l-BOT by [Engelfriet '75] □

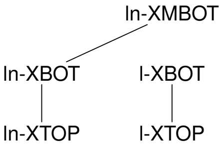
Proper generalization

Theorem (Engelfriet et al. '09)

All linear XTOP can be simulated by linear XBOT

Proof.

Standard construction trading input-deletion for output-deletion
see l-TOP $\subseteq$ l-BOT by [Engelfriet '75] □



Proper generalization

Theorem (Engelfriet et al. '09)

All XMBOT can be simulated by nondeleting XMBOT

Proof.

- Guess subtrees that will be deleted
- Process them in nullary states (i.e. look-ahead) □

Proper generalization

Theorem (ENGELFRIET et al. ’09)

All XMBOT can be simulated by nondeleting XMBOT

Proof.

- Guess subtrees that will be deleted
- Process them in nullary states (i.e. look-ahead) □

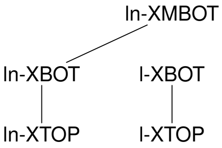
Proper generalization

Theorem (ENGELFRIET et al. ’09)

All XMBOT can be simulated by nondeleting XMBOT

Proof.

- Guess subtrees that will be deleted
- Process them in nullary states (i.e. look-ahead) □



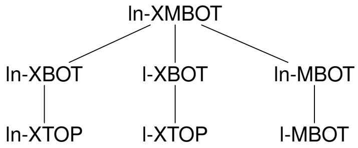
Proper generalization

Theorem (Engelfriet et al. ’09)

All XMBOT can be simulated by nondeleting XMBOT

Proof.

- Guess subtrees that will be deleted
- Process them in nullary states (i.e. look-ahead) □



Proper generalization

Theorem (Engelfriet et al. '09)

All XMBOT without recursive ε-rules can be simulated by MBOT

Proof.

- Decompose large left-hand sides using "multi"-states
- Attach finite effect of ε-rules

□

Proper generalization

Theorem (Engelfriet et al. ’09)

All XMBOT without recursive ε-rules can be simulated by MBOT

Proof.

- Decompose large left-hand sides using “multi”-states
- Attach finite effect of ε-rules □

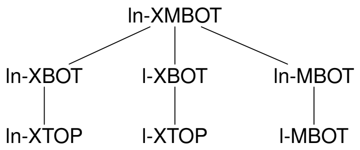
Proper generalization

Theorem (Engelfriet et al. '09)

All XMBOT without recursive ε-rules can be simulated by MBOT

Proof.

- Decompose large left-hand sides using "multi"-states
- Attach finite effect of ε-rules □



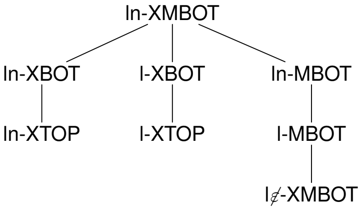
Proper generalization

Theorem (Engelfriet et al. '09)

All XMBOT without recursive ε-rules can be simulated by MBOT

Proof.

- Decompose large left-hand sides using "multi"-states
- Attach finite effect of ε-rules □



Proper generalization

Definition

- XTOP M sensible if $|u| \in \mathcal{O}(|t|)$ for all $(t, u) \in \tau_M$
- simple = linear and nondeleting

Theorem (Maletti '12)

All sensible XTOP can be simulated by simple MBOT

Proof.

- use (essentially) construction of [Engelfriet, Maneth '03]
- obtain finitely copying XTOP (without recursive ε-rules)
- apply [Engelfriet et al. '09] to obtain linear XMBOT
- previous theorems yield simple MBOT □

Proper generalization

Definition

- XTOP M sensible if $|u| \in \mathcal{O}(|t|)$ for all $(t, u) \in \tau_M$
- simple = linear and nondeleting

Theorem (Maletti '12)

All sensible XTOP can be simulated by simple MBOT

Proof.

- use (essentially) construction of [Engelfriet, Maneth '03]
- obtain finitely copying XTOP (without recursive ε-rules)
- apply [Engelfriet et al. '09] to obtain linear XMBOT
- previous theorems yield simple MBOT

□

Proper generalization

Definition

- XTOP M sensible if $|u| \in \mathcal{O}(|t|)$ for all $(t, u) \in \tau_M$
- simple = linear and nondeleting

Theorem (MALETTI '12)

All sensible XTOP can be simulated by simple MBOT

Proof.

- use (essentially) construction of [ENGELFRIET, MANETH '03]
- obtain finitely copying XTOP (without recursive ε-rules)
- apply [ENGELFRIET et al. '09] to obtain linear XMBOT
- previous theorems yield simple MBOT □

Proper generalization

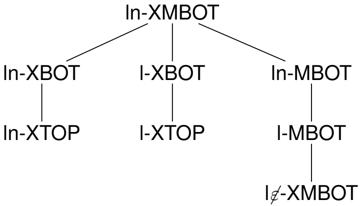
Corollary

All relevant XTOP can be simulated by simple XMBOT

Proper generalization

Corollary

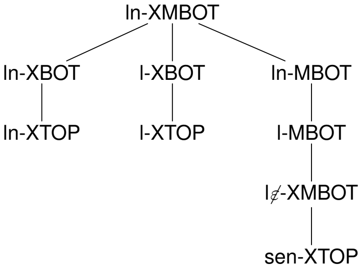
All relevant XTOP can be simulated by simple XMBOT



Proper generalization

Corollary

All relevant XTOP can be simulated by simple XMBOT



Proper generalization

Theorem

Simple MBOT cannot be simulated by XTOP

Proof.

- even simple MBOT can copy (Example)
- see BOT $\not\subseteq$ TOP by [ENGELFRIET '75] □

Theorem (GILDEA '12)

Simple MBOT cannot be weakly simulated by simple XTOP

Proper generalization

Theorem

Simple MBOT cannot be simulated by XTOP

Proof.

- even simple MBOT can copy (Example)
- see BOT $\not\subseteq$ TOP by [Engelfriet '75] □

Theorem (Gildea '12)

Simple MBOT cannot be weakly simulated by simple XTOP

Proper generalization

Theorem

Simple MBOT cannot be simulated by XTOP

Proof.

- even simple MBOT can copy (Example)
- see BOT $\not\subseteq$ TOP by [ENGELFRIET '75] □

Theorem (GILDEA '12)

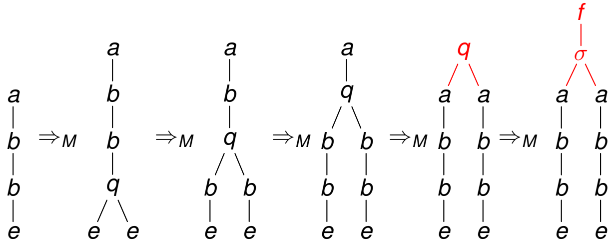
Simple MBOT cannot be weakly simulated by simple XTOP

Summary

- ✓ generalize XTOP (even properly)
- (b) discontinuities
- (c) binarizable
- (d) efficient input/output restriction
- (e) efficiently trainable
- (f) closed under composition

Discontinuities

Example (Derivation)



Discontinuities

- ✗ state covers 1 input subtree $\rightarrow$ no input discontinuities
- ✓ state covers several output subtrees $\rightarrow$ output discontinuities

Summary

- ✓ generalize XTOP (even properly)
- ✓ discontinuities (only output side)
- (c) binarizable
- (d) efficient input/output restriction
- (e) efficiently trainable
- (f) closed under composition

Binarization

Definition

XMBOT in 1-symbol normal form
if exactly 1 (input/output) symbol occurs per rule

Theorem (ENGELFRIET et al. '09)

All XMBOT can be simulated by 1-symbol normal form XMBOT

Binarization

Definition

XMBOT in 1-symbol normal form
if exactly 1 (input/output) symbol occurs per rule

Theorem (ENGELFRIET et al. '09)

All XMBOT can be simulated by 1-symbol normal form XMBOT

Binarization

Definition

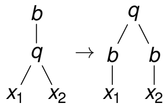
XMBOT in 1-symbol normal form
if exactly 1 (input/output) symbol occurs per rule

Theorem (ENGELFRIET et al. '09)

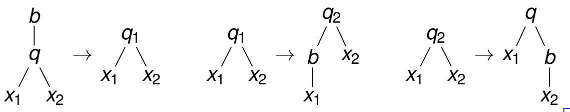
All XMBOT can be simulated by 1-symbol normal form XMBOT

Proof.

Original rule:



Replacement rules:



□

Binarization

Definition

XMBOT is fully binarized if ≤ 3 states per rule
(≤ 2 in left-hand side)

Theorem (M. ’11)

All XMBOT can be fully binarized (in linear time)

Binarization

Definition

XMBOT is fully binarized if ≤ 3 states per rule
(≤ 2 in left-hand side)

Theorem (M. '11)

All XMBOT can be fully binarized (in linear time)

Binarization

Definition

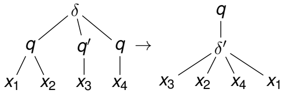
XMBOT is fully binarized if ≤ 3 states per rule
(≤ 2 in left-hand side)

Theorem (M. '11)

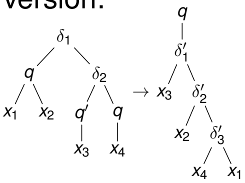
All XMBOT can be fully binarized (in linear time)

Proof (Binarize trees and transform into 1-symbol normal form).

Original rule:

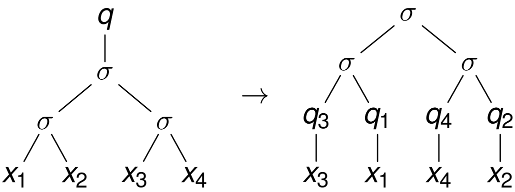


Binarized version:



□

Binarization



Comparison

STSG cannot be binarized, but people try . . .

- [ZHANG et al. '06]
- [DENERO et al. '09]

Corollary

*All XMBOT can be transformed (in linear time)
from joint to conditional distribution*

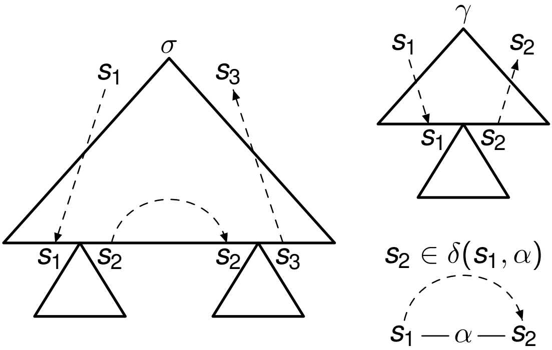
Summary

- ✓ generalize XTOP (even properly)
- ✓ discontinuities (only output side)
- ✓ binarizable
- (d) efficient input/output restriction
- (e) efficiently trainable
- (f) closed under composition

Input/output restriction

Definition

Input restriction restricts the string language of the domain of an XMBOT to a regular language



Input/output restriction

Theorem (M., Satta '10 & M. '11)

Restricting the … by FSA A is …

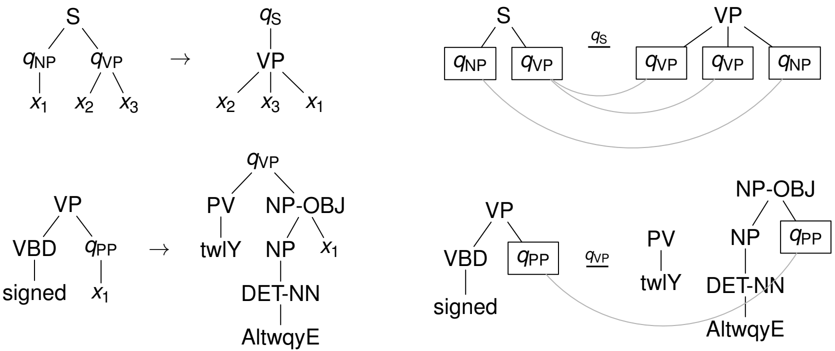
device	*input*	*output*
linear XMBOT M	$\mathcal{O}(\lvert M\rvert \cdot \lvert A\rvert^3)$	$\mathcal{O}(\lvert M\rvert \cdot \lvert A\rvert^x)$
simple XTOP M	$\mathcal{O}(\lvert M\rvert \cdot \lvert A\rvert^y)$	$\mathcal{O}(\lvert M\rvert \cdot \lvert A\rvert^y)$

with $x = 2\,\mathrm{rk}(M) + 2$ *and* $y = 2\,\mathrm{rk}(M) + 5$

Summary

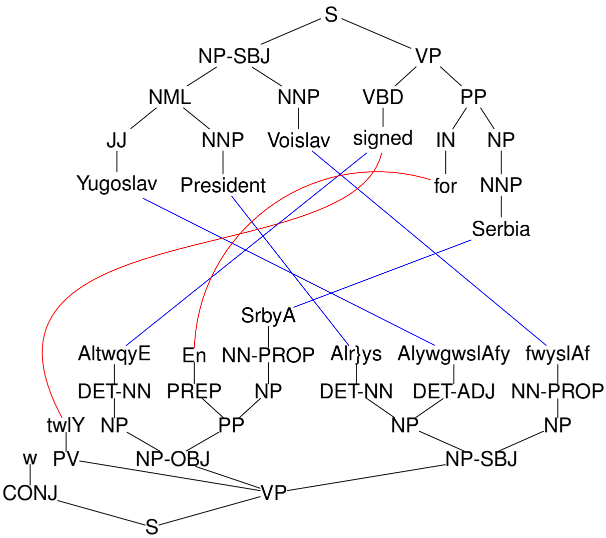
- ✓ generalize XTOP (even properly)
- ✓ discontinuities (only output side)
- ✓ binarizable
- ✓ efficient input/output restriction (less efficient for output)
- (e) efficiently trainable
- (f) closed under composition

A top-down variant

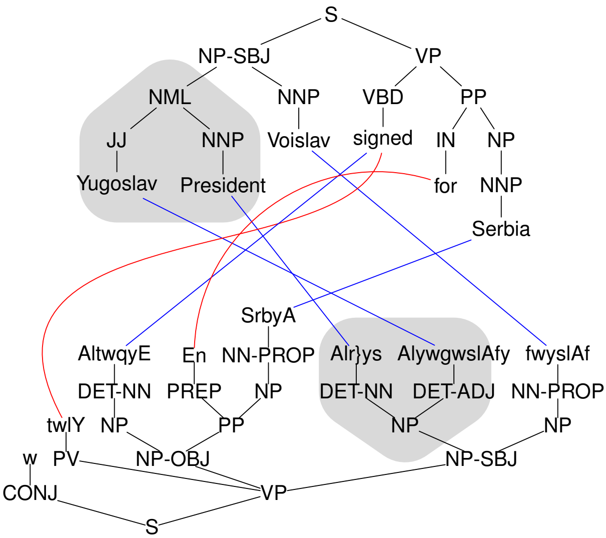


[M. '11]

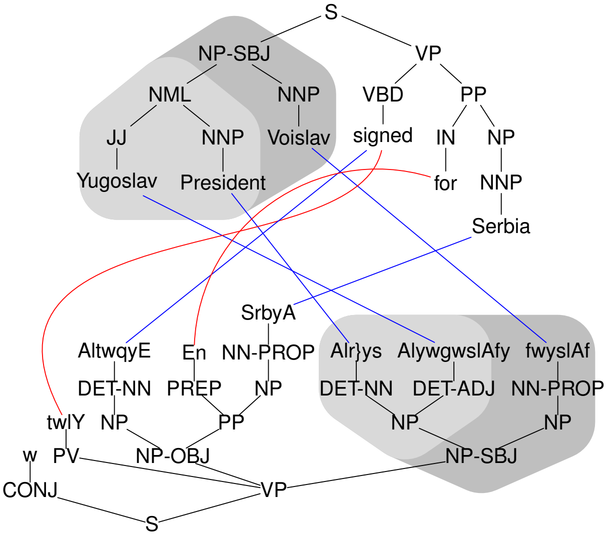
Rule extraction



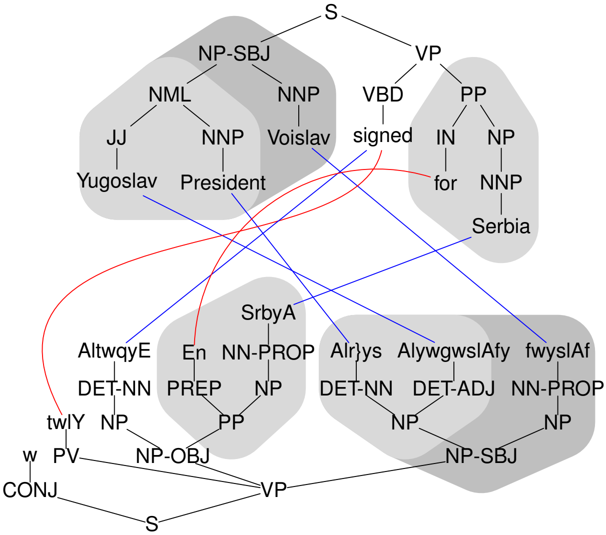
Rule extraction



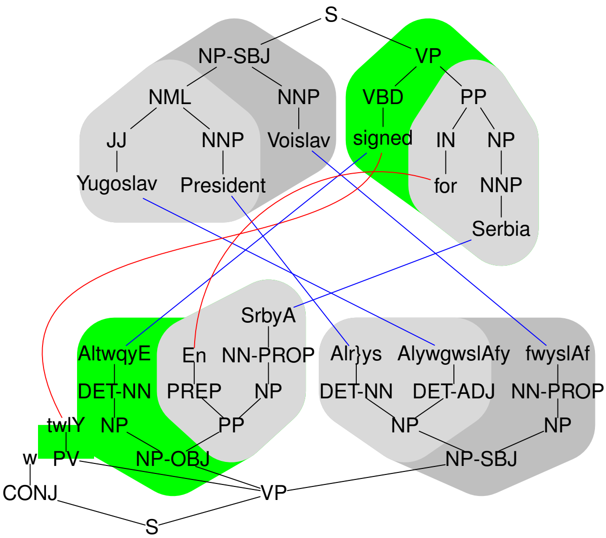
Rule extraction



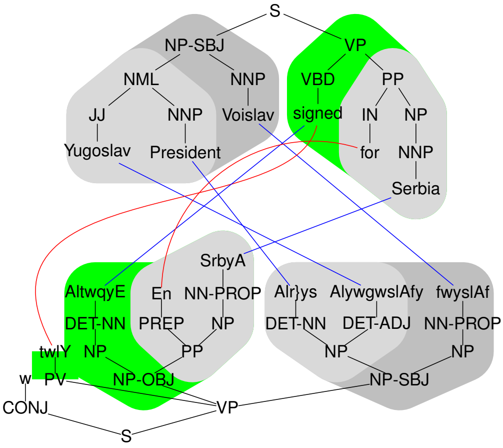
Rule extraction

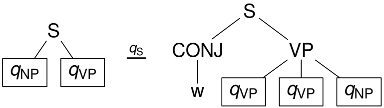


Rule extraction

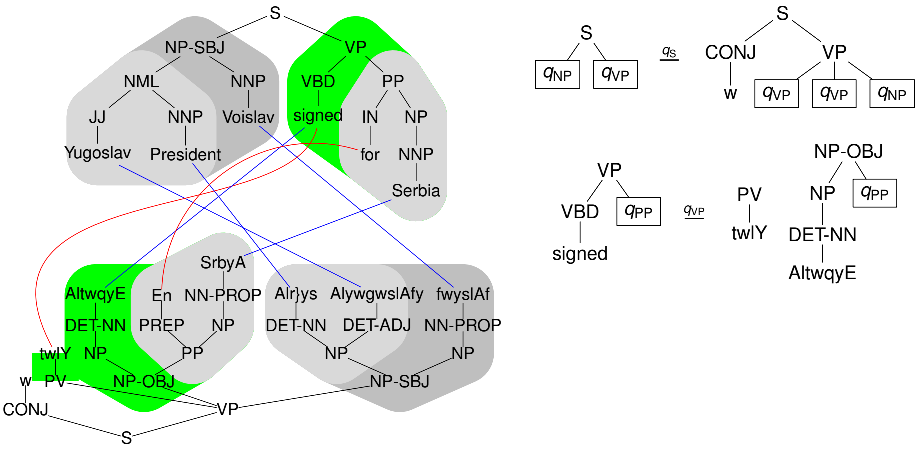


Rule extraction





Rule extraction



EM training

Theorem

Derivations of XMBOT are regular (even in the weighted case)

Conclusion

program of [GRAEHL et al '08] works

- given translation pair (s_1, s_2)
- input- and output restrict to s_1 and s_2
- build derivations
- compute relative "usefulness" of each rule
- move to the next training sentence (and start anew)

EM training

Theorem

Derivations of XMBOT are regular (even in the weighted case)

Conclusion

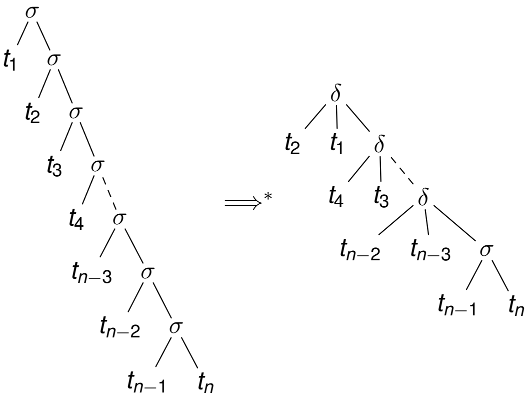
program of [Graehl et al '08] works

- given translation pair (s_1, s_2)
- input- and output restrict to s_1 and s_2
- build derivations
- compute relative "usefulness" of each rule
- move to the next training sentence (and start anew)

Summary

- ✓ generalize XTOP (even properly)
- ✓ discontinuities (only output side)
- ✓ binarizable
- ✓ efficient input/output restriction (less efficient for output)
- ✓ efficiently trainable (messy for permissive MBOT)
- (f) closed under composition

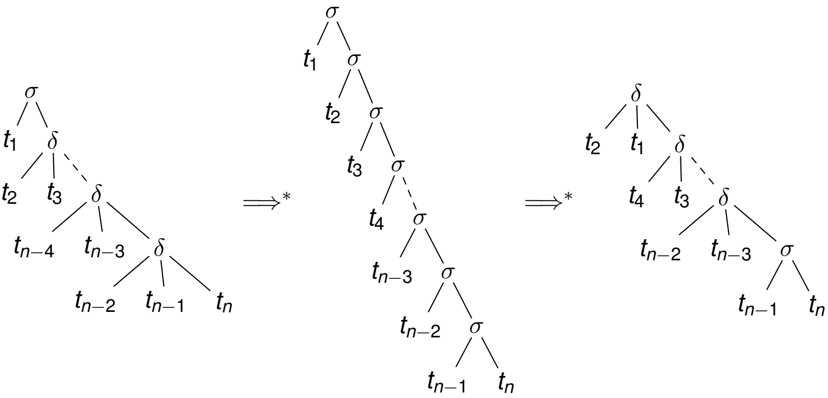
Composition of STSG



Conclusion

STSGs are not composable!

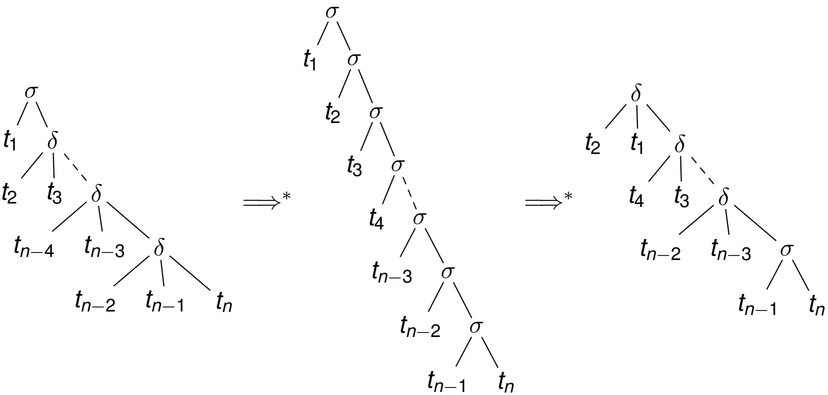
Composition of STSG



Conclusion

STSGs are not composable!

Composition of STSG



Conclusion

STSGs are not composable!

Composition of XTOP

restrictions	closed?	level of closure
simple, non-erasing, ε-free	✗	2
simple, non-erasing	✗	∞
simple, ε-free	✗	∞
simple	✗	∞
linear	✗	≥ 2
linear with regular look-ahead	✗	≥ 2
general	✗	∞

Composition of XMBOT

restrictions	closed?	level of closure
simple	✓	1
linear	✓	1
general	✗	∞ (?)

Composition construction

Definition

XMBOT $M = (Q, \Sigma, F, R)$ and $N = (Q', \Sigma, G, R')$
in 1-symbol normal form

$$M \,;\, N = (Q(Q'), \Sigma, F(G), R'')$$

with

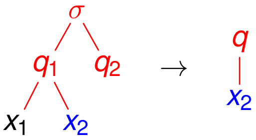
1. input-consuming rules from input-consuming rules of R
2. ε-rules from ε-rules of R'
3. ε-rules from ε-rule of R followed by input consuming rule of R'

Composition construction

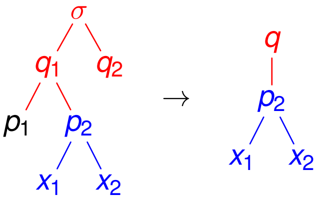
Example

(1) Input-consuming rule of R and resulting rule:

Input rule (of R):



Constructed rule:



Composition construction

Example

(2) ε-rule of R' and resulting rule:

Input rule (of R'): $p_1 \to p(\alpha)$

Constructed rule: $q_1(p_1, p_2(x_1, x_2)) \to q_1(p(\alpha), p_2(x_1, x_2))$

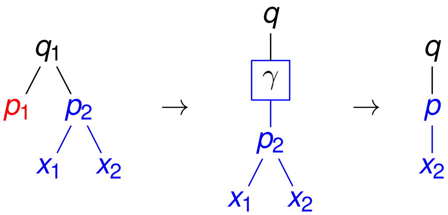
Composition construction

Example

(3) ε-rule of R and input-consuming of R' and resulting rule:

Input rules:

Constructed rule:



Composition construction

Theorem (Engelfriet et al. '09)

The standard bottom-up tree transducer composition results hold

Summary

- ✓ generalize XTOP (even properly)
- ✓ discontinuities (only output side)
- ✓ binarizable
- ✓ efficient input/output restriction (less efficient for output)
- ✓ efficiently trainable (messy for permissive MBOT)
- ✓ closed under composition (standard bottom-up results)

Summary

- ✓ generalize XTOP
- ✓ discontinuities
- ✓ binarizable
- ✓ efficient input/output restriction
- ✓ efficiently trainable
- ✓ closed under composition

Summary

- ✓ generalize XTOP
- ✓ discontinuities
- ✓ binarizable
- ✓ efficient input/output restriction
- ✓ efficiently trainable
- ✓ closed under composition
- ✓ preserve regularity backward

Summary

- ✓ generalize XTOP
- ✓ discontinuities
- ✓ binarizable
- ✓ efficient input/output restriction
- ✓ efficiently trainable
- ✓ closed under composition
- ✓ preserve regularity backward
- ✗ preserve regularity forward
- ✗ symmetric

Overview

XMBOT in machine translation

Moses [KOEHN et al. '07]

- framework for statistical MT
- implementations for many standard tasks (alignment, lexical scores, language model, BLEU scoring)
- supports syntax-based MT

We added

- XMBOT rule support
- XMBOT chart decoder
- adjusted language model calls

XMBOT in machine translation

Moses [Koehn et al. '07]

- framework for statistical MT
- implementations for many standard tasks (alignment, lexical scores, language model, BLEU scoring)
- supports syntax-based MT

We added

- XMBOT rule support
- XMBOT chart decoder
- adjusted language model calls

XMBOT in machine translation

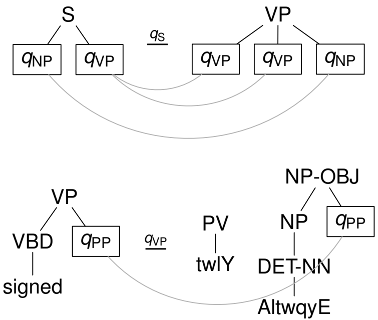
Moses [Koehn et al. '07]

- framework for statistical MT
- implementations for many standard tasks
(alignment, lexical scores, language model, BLEU scoring)
- supports syntax-based MT

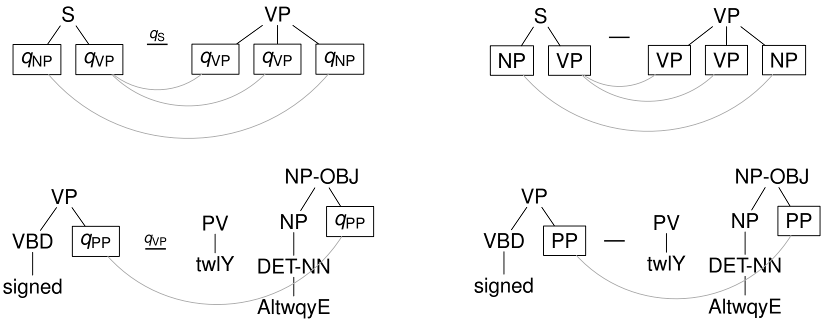
We added

- XMBOT rule support
- XMBOT chart decoder
- adjusted language model calls (still broken)

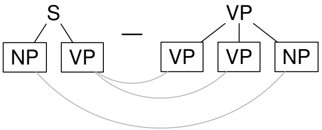
XMBOT rule encoding



XMBOT rule encoding

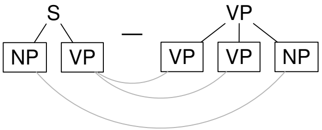


XMBOT rule encoding

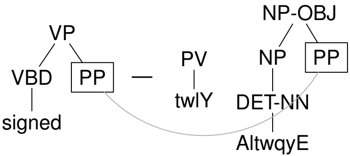



```
S(NP,VP) ||| VP(VP,VP,NP) ||| S ||| VP ||| 0-2 1-0 1-1 ||| ...
```

XMBOT rule encoding




```
S(NP,VP) ||| VP(VP,VP,NP) ||| S ||| VP ||| 0-2 1-0 1-1 ||| ...
```




```
VP(VBD(signed),PP) ||| PV(twlY) || NP-OBJ(NP(DET-NN(AltwqyE)),PP) |||
VP ||| PV NP-OBJ ||| || 0-0 ||| ...
```

XMBOT decoder

Fabienne Braune

- CYK-like chart parser
- only forward application (backward planned)
- supports all standard features
- integrated cube pruning with language model

Notes

- reasonably fast
- generated the examples in Motivation (with only translation weights)
- we are still working on it

XMBOT decoder

Fabienne Braune

- CYK-like chart parser
- only forward application (backward planned)
- supports all standard features
- integrated cube pruning with language model

Notes

- reasonably fast
- generated the examples in Motivation (with only translation weights)
- we are still working on it

XMBOT decoder

Fabienne Braune

- CYK-like chart parser
- only forward application (backward planned)
- supports all standard features
- integrated cube pruning with language model

Notes

- reasonably fast
- generated the examples in Motivation (with only translation weights)
- we are still working on it

XMBOT external tools

NINA SEEMANN

- rule extraction
- input/output restriction
- EM training
- conversion tools, pipeline scripts, . . .

Notes

- in PYTHON (not inside MOSES)
- computationally quite expensive
- variants for reduced POS-tags

XMBOT external tools

Nina Seemann

- rule extraction
- input/output restriction
- EM training
- conversion tools, pipeline scripts, ...

Notes

- in Python (not inside Moses)
- computationally quite expensive
- variants for reduced POS-tags

No BLEU score yet

No BLEU score yet, but we are close!

References

- Aho, Ullman: *The theory of parsing, translation, and compiling.* Prentice Hall. 1972
- Arnold, Dauchet: Morphismes et bimorphismes d'arbres. *Theoret. Comput. Sci.* 20(1):33–93, 1982
- Charniak, Johnson: Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *ACL* 2005
- DeNero, Pauls, Klein: Asynchronous binarization for synchronous grammars. In *ACL* 2009
- Engelfriet: Bottom-up and top-down tree transformations — a comparison. *Math. Systems Theory* 9(3), 1975
- Engelfriet, Maneth: Macro tree translations of linear size increase are MSO definable. *SIAM J. Comput.* 32(4):950–1006, 2003
- Engelfriet, Lilin, Maletti: Extended multi bottom-up tree transducers — composition and decomposition. *Acta Inf.* 46(8):561–590, 2009
- Galley, Hopkins, Knight, Marcu: What's in a translation rule? In *HLT-NAACL* 2004
- Graehl, Knight, May: Training tree transducers. *Comput. Linguist.* 34(3):391–427, 2008
- Gildea: On the string translations produced by multi bottom-up tree transducers. *Comput. Linguist.*, 2012 (to appear)
- Koehn, Hoang, Birch, Callison-Burch, Federico, Bertoldi, Cowan, Shen, Moran, Zens, Dyer, Bojar, Constantin, Herbst: Moses: open source toolkit for statistical machine translation. In ACL 2007
- Maletti, Satta: Parsing and translation algorithms based on weighted extended tree transducers. In *ATANLP* 2010
- Maletti: An alternative to synchronous tree substitution grammars. *J. Nat. Lang. Engrg.* 17(2):221–242, 2011
- Maletti: How to train your multi bottom-up tree transducer. In *ACL* 2011
- Maletti: Every sensible extended top-down tree transducer is a multi bottom-up tree transducer. In *HLT-NAACL* 2012
- May, Knight: Tiburon — a weighted tree automata toolkit. In *CIAA* 2006
- Och, Ney: A systematic comparison of various statistical alignment models. *Comput. Linguist.* 29(1):19–51, 2003
- Schmid: Trace prediction and recovery with unlexicalized PCFGs and slash features. In *COLING-ACL* 2006
- Zhang, Huang, Gildea, Knight: Synchronous binarization for machine translation. In *HLT-NAACL* 2006