Extended Top-down Tree Transducers

Andreas Maletti

International Computer Science Institute
Berkeley, CA, USA

email: `maletti@icsi.berkeley.edu`

Umeå — June 25, 2008

Motivating samples (GOOGLE TRANSLATE)

Example (Input in English)

Holly picks flowers to tie them around July's neck.

Translation to German

Stechpalme wählt Blumen aus, um sie um Ansatz Julis zu binden.
Holly selects flowers to tie them to approach of July. (??)

Retranslation to English

Stechpalme selects flowers, in order to bind it around beginning of July.

Motivating samples (GOOGLE TRANSLATE)

Example (Input in English)

Holly picks flowers to tie them around July's neck.

Translation to German

Stechpalme wählt Blumen aus, um sie um Ansatz Julis zu binden.
Holly selects flowers to tie them to approach of July. (??)

Retranslation to English

Stechpalme selects flowers, in order to bind it around beginning of July.

Motivating samples (GOOGLE TRANSLATE)

Example (Input in English)

Holly picks flowers to tie them around July's neck.

Translation to German

Stechpalme wählt Blumen aus, um sie um Ansatz Julis zu binden.
Holly selects flowers to tie them to approach of July. (??)

Retranslation to English

Stechpalme selects flowers, in order to bind it around beginning of July.

Another Sample

Example (Input in German plus reference translation)

Man könnte meinen, meine Meinung zählt nicht.

One could believe that my opinion does not count.

Translation to English

One could not do mine, my opinion counts.

Another Sample

Example (Input in German plus reference translation)

Man könnte meinen, meine Meinung zählt nicht.

One could believe that my opinion does not count.

Translation to English

One could not do mine, my opinion counts.

Syntax-based Machine Translation

Overview

Parse Trees	$\Rightarrow$	Translation (e.g. TST)	$\Rightarrow$	Parse Trees	$\Rightarrow$	Reranking (e.g. WTA)	$\Rightarrow$	Parse Trees
$\Uparrow$								$\Downarrow$
Input Text	$\Rightarrow$	Translation (e.g. FST)	$\Rightarrow$	Output Texts	$\Rightarrow$	Reranking (e.g. FSA)	$\Rightarrow$	Final Outputs

Abbreviations

- FST = Finite State Transducer
- FSA = Finite State Automaton

Syntax-based Machine Translation

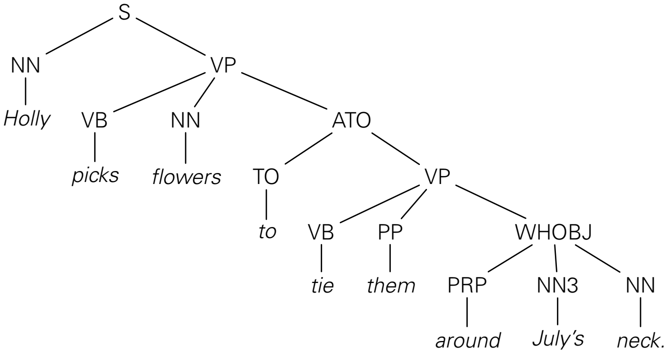
Overview

Parse Trees	$\Rightarrow$	Translation (e.g. TST)	$\Rightarrow$	Parse Trees	$\Rightarrow$	Reranking (e.g. WTA)	$\Rightarrow$	Parse Trees
$\Uparrow$								$\Downarrow$
Input Text	$\Rightarrow$	Translation (e.g. FST)	$\Rightarrow$	Output Texts	$\Rightarrow$	Reranking (e.g. FSA)	$\Rightarrow$	Final Outputs

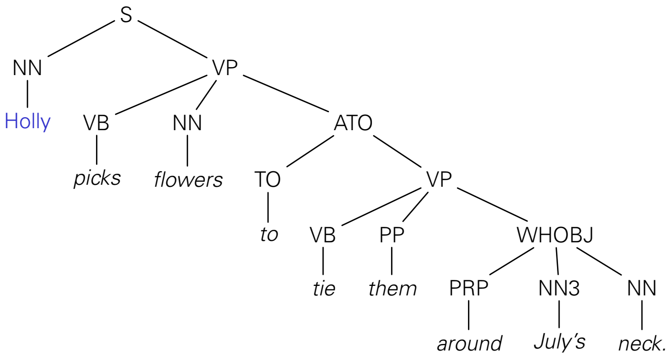
Abbreviations

- TST = Tree Series Transducer
- WTA = Weighted Tree Automaton

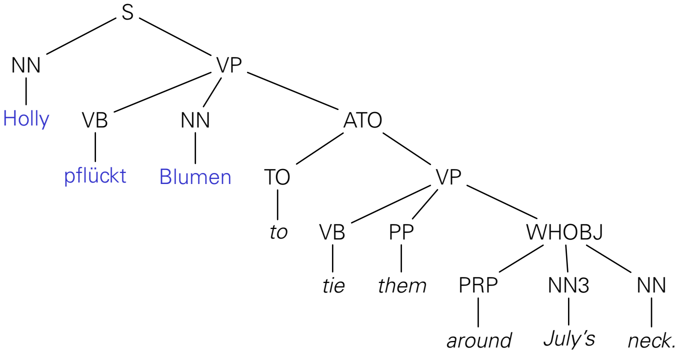
Syntactic Analysis



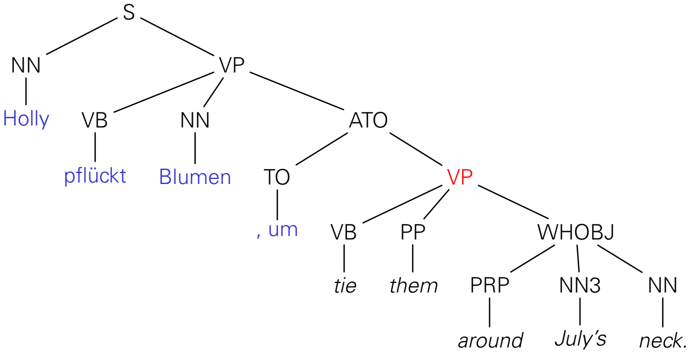
Syntax-based Machine Translation



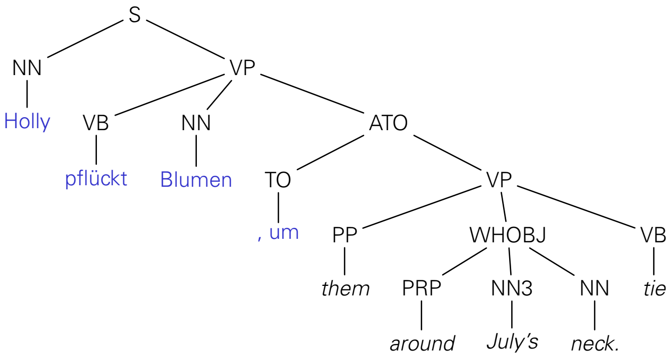
Syntax-based Machine Translation



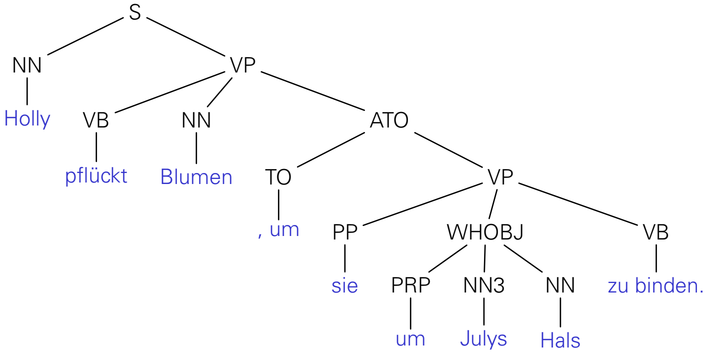
Syntax-based Machine Translation



Syntax-based Machine Translation



Syntax-based Machine Translation



Why Syntax?

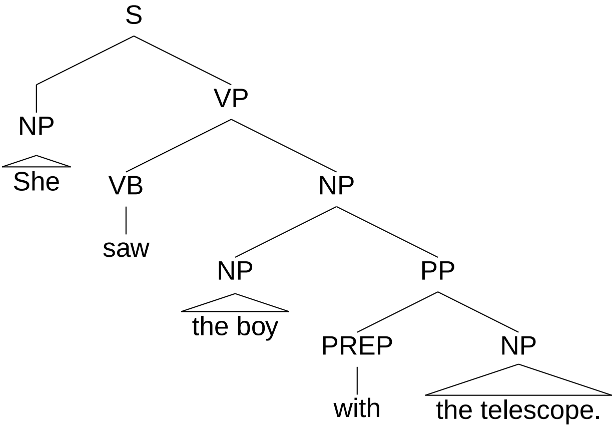
Example

She saw the boy with the telescope.

Why Syntax?

Example

She saw the boy with the telescope.

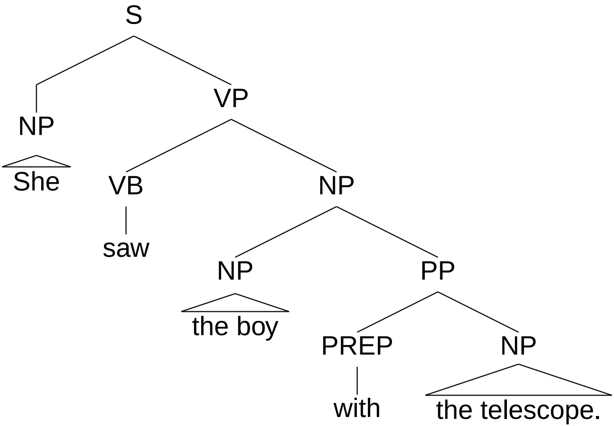


maybe

Why Syntax?

Example

She saw the boy with the telescope.

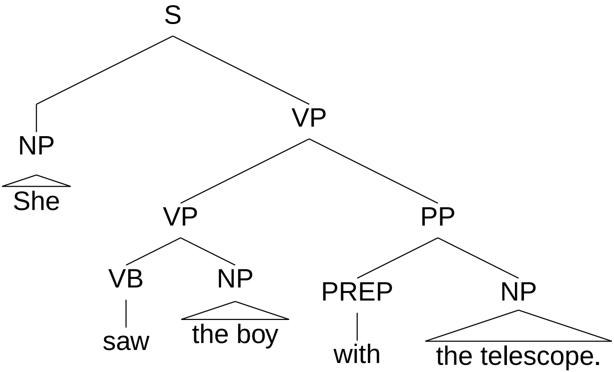


0.33

Why Syntax?

Example

She saw the boy with the telescope.

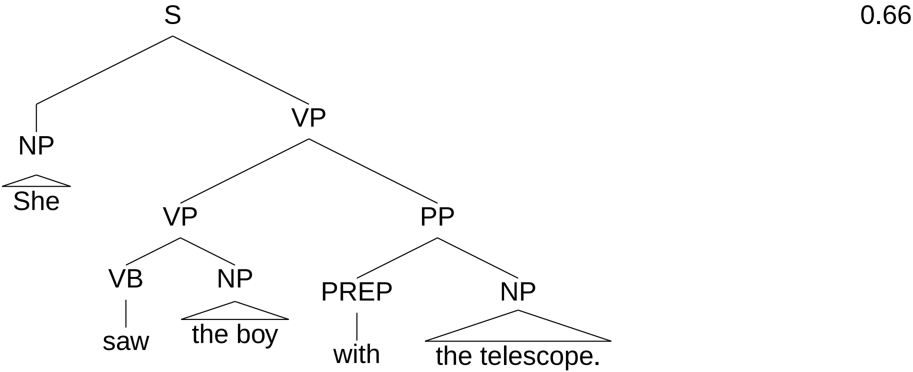


more likely

Why Syntax?

Example

She saw the boy with the telescope.



Why Syntax?

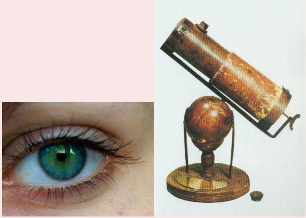
Example

She saw the boy with the telescope.

Benefits

- Parsing ambiguities do not affect translation model
- Translation can use syntactic structure

Remaining problem



vs.

Translation Model

Criteria

(a) Generalize FST; in particular, epsilon-transitions
(b) Efficient training
(c) Handles rotation
(d) Closed under composition

Existing Models

Model \ Criterion	(a)	(b)	(c)	(d)
Top-down tree transducer	–	x	–	x
Quasi-alphabetic tree bimorphism	–	?	–	x
Synchronous context-free grammar	x	x	–	x
Synchronous tree substitution grammar	x	x	x	–
Synchronous tree adjoining grammar	x	x	x	–

Table of Contents

Syntax

Definition

$(Q, \Sigma, \Delta, I, R)$ xtt

- Q finite set of states
- Σ and Δ ranked alphabets
- $I \subseteq Q$ initial states
- R finite set of rules of the form $q(t) \to u$ where $q \in Q$, $t \in T_\Sigma(X)$, and $u \in T_\Delta(Q(\text{var}(t)))$

References

- Arnold, Dauchet: Bi-transductions de forêts. ICALP 1976
- Graehl, Knight: Training Tree Transducers. HLT-NAACL 2004

Syntax

Definition

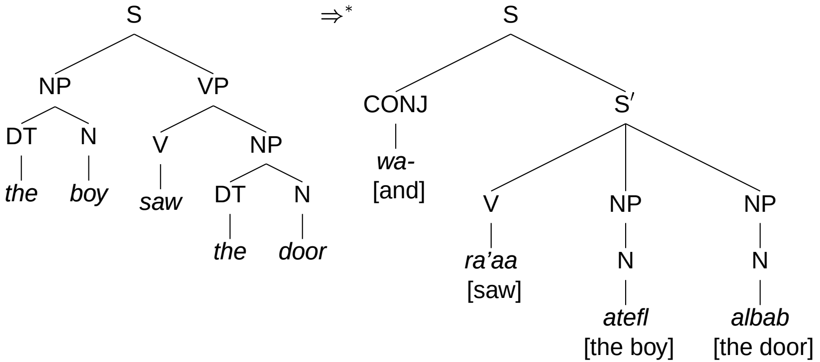
$(Q, \Sigma, \Delta, I, R)$ xtt

- Q finite set of states
- Σ and Δ ranked alphabets
- $I \subseteq Q$ initial states
- R finite set of rules of the form $q(t) \to u$ where $q \in Q$, $t \in T_\Sigma(X)$, and $u \in T_\Delta(Q(\mathrm{var}(t)))$

References

- Arnold, Dauchet: Bi-transductions de forêts. ICALP 1976
- Graehl, Knight: Training Tree Transducers. HLT-NAACL 2004

Syntax — Example



Question

How to implement this English $\rightarrow$ Arabic translation using xtt?

Syntax — Example (cont'd)

Example

States $\{q, q_{\mathrm{S}}, q_{\mathrm{V}}, q_{\mathrm{NP}}\}$ of which only q is initial

$$
\begin{aligned}
q(x_1) &\to q_{\mathrm{S}}(x_1) && (r_1) \\
q(x_1) &\to \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), q_{\mathrm{S}}(x_1)) && (r_2) \\
q_{\mathrm{S}}(\mathrm{S}(x_1, \mathrm{VP}(x_2, x_3))) &\to \mathrm{S}'(q_{\mathrm{V}}(x_2), q_{\mathrm{NP}}(x_1), q_{\mathrm{NP}}(x_3)) && (r_3) \\
q_{\mathrm{V}}(\mathrm{V}(\textit{saw})) &\to \mathrm{V}(\textit{ra'aa}) && (r_4) \\
q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{boy}))) &\to \mathrm{NP}(\mathrm{N}(\textit{atefl})) && (r_5) \\
q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))) &\to \mathrm{NP}(\mathrm{N}(\textit{albab})) && (r_6)
\end{aligned}
$$

Syntax — Example (cont'd)

Example

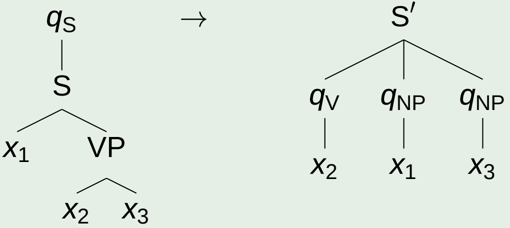
1. Nondeterminism and epsilon rules (rules r_1 and r_2)

$$\begin{array}{c} q \\ | \\ x_1 \end{array} \;\rightarrow\; \begin{array}{c} q_{\mathrm{S}} \\ | \\ x_1 \end{array} \qquad \text{and} \qquad \begin{array}{c} q \\ | \\ x_1 \end{array} \;\rightarrow\; \begin{array}{c} \mathrm{S} \\ \diagup \quad \diagdown \\ \begin{array}{cc} \mathrm{CONJ} & q_{\mathrm{S}} \\ | & | \\ \textit{wa-} & x_1 \end{array} \end{array}$$

Syntax — Example (cont'd)

Example

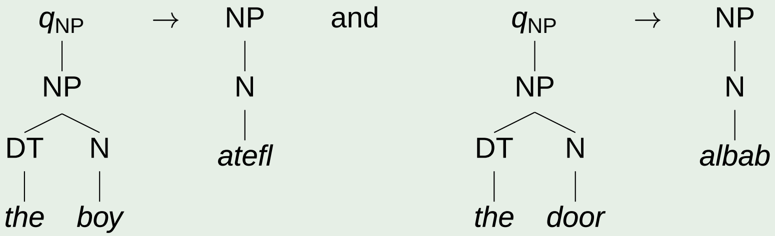
1. Nondeterminism and epsilon rules (rules r_1 and r_2)
2. Deep attachment of variables (rule r_3)



Syntax — Example (cont'd)

Example

1. Nondeterminism and epsilon rules (rules r_1 and r_2)
2. Deep attachment of variables (rule r_3)
3. Finite look-ahead (rules r_4 and r_5)



Semantics

$M = (Q, \Sigma, \Delta, I, R)$

Definition

Let $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$. Then $\xi \Rightarrow_M \zeta$ if there exist

1. a rule $q(t) \to u$
2. a substitution $\theta \colon X \to T_\Sigma$
3. a position $w \in \text{pos}(\xi)$

such that $\xi|_w = q(t\theta)$ and $\zeta = \xi[u\theta]_w$

Definition

Computed transformation:

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

Semantics

$M = (Q, \Sigma, \Delta, I, R)$

Definition

Let $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$. Then $\xi \Rightarrow_M \zeta$ if there exist

1. a rule $q(t) \to u$
2. a substitution $\theta \colon X \to T_\Sigma$
3. a position $w \in \mathrm{pos}(\xi)$

such that $\xi|_w = q(t\theta)$ and $\zeta = \xi[u\theta]_w$

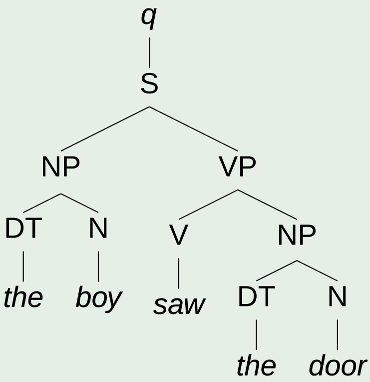
Definition

Computed transformation:

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

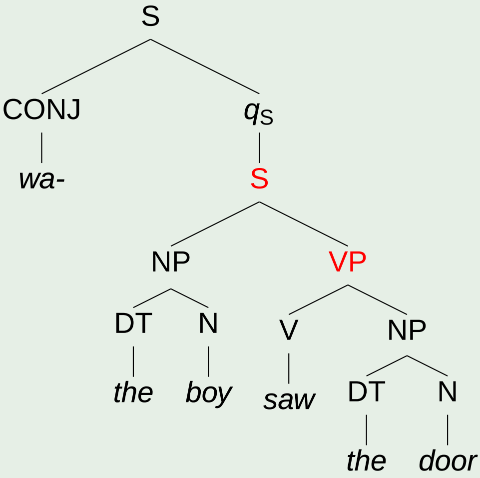
Semantics — Example

Example



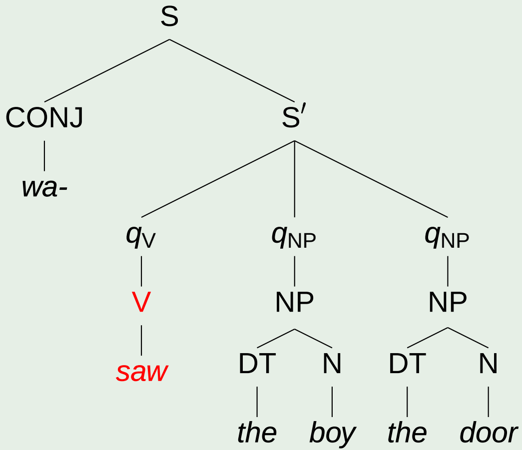
Semantics — Example

Example



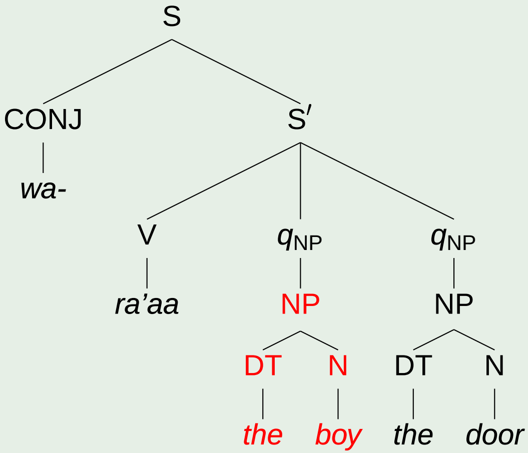
Semantics — Example

Example



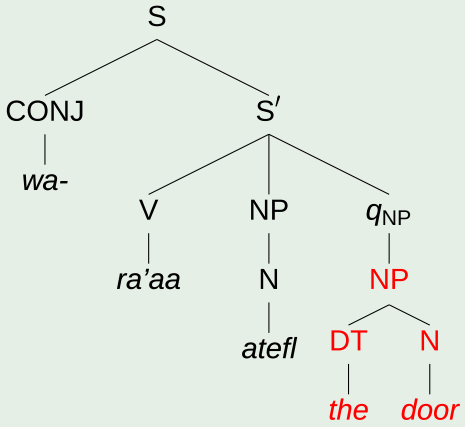
Semantics — Example

Example



Semantics — Example

Example



Semantics — Example

Example

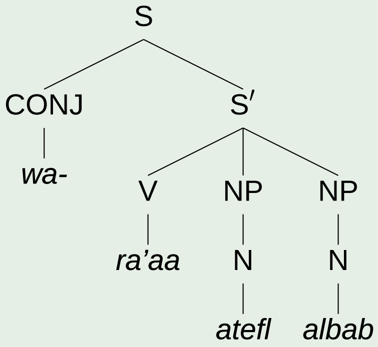


Table of Contents

Wanted Expressivity

Criteria

1. Generalize FST including epsilon rules (ln-tdtt: no, ln-xtt: yes)
2. Efficiently trainable (ln-tdtt: yes, ln-xtt: yes)
3. Can handle rotations (ln-tdtt: no, ln-xtt: yes)

4. Can handle flattenings (ln-tdtt: no, ln-xtt: yes)

Wanted Expressivity

Criteria

1. Generalize FST including epsilon rules (ln-tdtt: no, ln-xtt: yes)
2. Efficiently trainable (ln-tdtt: yes, ln-xtt: yes)
3. Can handle rotations (ln-tdtt: no, ln-xtt: yes)

4. Can handle flattenings (ln-tdtt: no, ln-xtt: yes)

Wanted Expressivity

Criteria

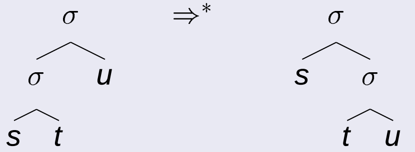
1. Generalize FST including epsilon rules (ln-tdtt: no, ln-xtt: yes)
2. Efficiently trainable (ln-tdtt: yes, ln-xtt: yes)
3. Can handle rotations (ln-tdtt: no, ln-xtt: yes)

4. Can handle flattenings (ln-tdtt: no, ln-xtt: yes)

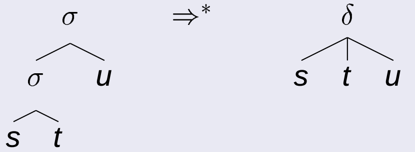
Wanted Expressivity

Criteria

1. Generalize FST including epsilon rules (ln-tdtt: no, ln-xtt: yes)
2. Efficiently trainable (ln-tdtt: yes, ln-xtt: yes)
3. Can handle rotations (ln-tdtt: no, ln-xtt: yes)



4. Can handle flattenings (ln-tdtt: no, ln-xtt: yes)



Wanted Expressivity (Cont'd)

Criteria

1. Preservation of Recognizability (ln-tdtt: yes, ln-xtt: yes)
2. Closure under composition (ln-tdtt: yes, ln-xtt: no)

Definition

- linear: no right-hand side contains a duplicate variable
- non-deleting: all right-hand sides contain all variables of their left-hand side
- epsilon-free: no rules of the form $q(x) \to u$

Wanted Expressivity (Cont'd)

Criteria

1. Preservation of Recognizability (ln-tdtt: yes, ln-xtt: yes)
2. Closure under composition (ln-tdtt: yes, ln-xtt: no)

Definition

- linear: no right-hand side contains a duplicate variable
- non-deleting: all right-hand sides contain all variables of their left-hand side
- epsilon-free: no rules of the form $q(x) \to u$

Features of xtt

Discriminative features

- Finite look-ahead
- Epsilon rules
- Deep attachment of variables

Features of xtt

Discriminative features

- Finite look-ahead
- Epsilon rules
- Deep attachment of variables

Features of xtt

Discriminative features

- Finite look-ahead
- Epsilon rules
- Deep attachment of variables

Hasse Diagram

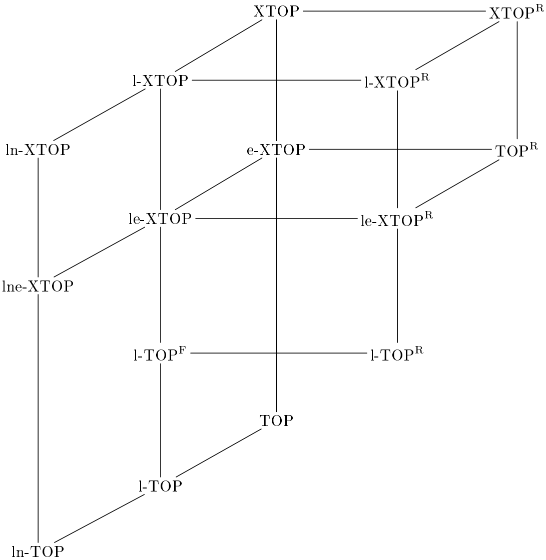


Table of Contents

Closure under Composition

Theorem

Every $\mathrm{l\text{-}TOP} \subseteq \mathcal{L} \subseteq \mathrm{XTOP}$ *is not closed under composition.*

Proof.

Composition closure of $\mathrm{l\text{-}TOP}$ is $\mathrm{l\text{-}TOP}^{\mathsf{R}}$. By the diagram, $\mathrm{l\text{-}TOP}^{\mathsf{R}} \not\subseteq \mathrm{XTOP}$. □

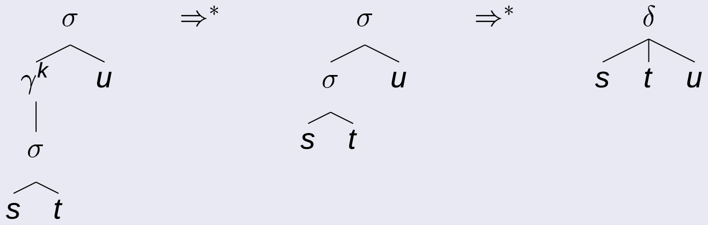
Closure under Composition (Cont'd)

Theorem

Every $\text{ln-TOP} \subseteq \mathcal{L} \subseteq \text{l-XTOP}^R$ *that contains rotations or flattenings is not closed under composition.*

Proof.

Prove $\text{ln-TOP} \,;\, \{\tau_{\text{flat}}\} \not\subseteq \text{l-XTOP}^R$ using, e.g.,



□

Closure under Composition (Cont'd)

Theorem

XTOP^R *is not closed under composition.*

Proof.

Follow classical proof for TOP^R. □

Conclusion or Bad news

No (mentioned) class of xtt computes a closed class of transformation.

Closure under Composition (Cont'd)

Theorem

XTOP^R *is not closed under composition.*

Proof.

Follow classical proof for TOP^R. □

Conclusion or Bad news

No (mentioned) class of xtt computes a closed class of transformation.

Compositions

Problem

Compositions are extremely important (e.g., for a framework)!

Questions

1. Identify a suitable subclasses that is closed under composition (expressive vs. closure)
2. Determine whether two given l-xtt can be composed
3. What is the composition closure of l-xtt
4. Identify superclasses that are closed under composition and still preserve recognizability (preservation vs. closure)
5. What is the effect of epsilon rules on composition closure

Table of Contents

Tiburon

Features

- Implements xtt (and tree automata; everything also weighted)
- Framework with command-line interface
- Optimized for machine translation

Algorithms

- Application of xtt to input tree
- Application of xtt to input language
- Backward application of xtt to output language
- Composition (for some xtt)
- ...

Example in Tiburon

Example

```
q
q.x0:                          -> qS.x0
q.x0:                          -> S(CONJ(wa-) qS.x0)
qS.S(x0: VP(x1: x2:))          -> S’(qV.x1 qNP.x0 qNP.x2)
qV.V(saw)                      -> V(ra’aa)
qNP.NP(DT(the) N(boy))         -> NP(N(atefl))
qNP.NP(DT(the) N(door))        -> NP(N(albab))
```

References

- Arnold, Dauchet: Bi-transductions de forêts. ICALP 1976
- Baker: Composition of top-down and bottom-up tree transducers. *Inform. Control 41*. 1979
- Engelfriet: Bottom-up and top-down tree transformations—a comparison. *Math. Syst. Theory 9*. 1975
- Engelfriet: Top-down tree transducers with regular look-ahead. *Math. Syst. Theory 10*. 1976
- May, Knight: Tiburon: A Weighted Tree Automata Toolkit. CIAA 2006
- M., Graehl, Hopkins, Knight: The power of extended top-down tree transducers. 2008

Thank You for your attention!