Weighted Tree Transducers in Natural Language Processing

Andreas Maletti

Universitat Rovira i Virgili
Tarragona, Spain

email: andreas.maletti@urv.cat

Wrocław — May 17, 2010

Collaborators

Joint work with

- JOOST ENGELFRIET, LIACS, Leiden, The Netherlands
- ZOLTÁN FÜLÖP, University of Szeged, Hungary
- JONATHAN GRAEHL, USC, Los Angeles, CA, USA
- MARK HOPKINS, Language Weaver Inc., Los Angeles, CA, USA
- KEVIN KNIGHT, USC, Los Angeles, CA, USA
- ERIC LILIN, Université de Lille, France
- GIORGIO SATTA, University of Padua, Italy
- HEIKO VOGLER, TU Dresden, Germany

Motivation

Example (Input in Catalan)

Benvolguda i benvolgut membre de la comunitat universitària,
Avui dilluns es duu a terme el darrer Consell de Govern del meu mandat com a rector; el proper dia 6 de maig, com correspon, hi haurà una nova elecció on tota la comunitat universitària podrà escollir nou rector o rectora.
Aquest darrer consell té, naturalment, un caràcter marcadament tècnic; l'ordre del dia complet el trobaràs adjunt al final d'aquest text. A continuació et comento només els punts que, al meu parer, poden ser més del teu interès.

Translation (Google Translate) to English

Dear and beloved member of the university community,
Today is Monday carried out by the Governing Council last of my term as rector, the next day, May 6, as appropriate, there will be another election where the entire university community can choose new rector.
This last advice is, of course, a markedly technician complete agenda can be found attached to the end of this text. Then I said only the points that I believe may be of interest.

Motivation

Example (Input in Catalan)

Benvolguda i benvolgut membre de la comunitat universitària,
Avui dilluns es duu a terme el darrer Consell de Govern del meu mandat com a rector; el proper dia 6 de maig, com correspon, hi haurà una nova elecció on tota la comunitat universitària podrà escollir nou rector o rectora.
Aquest darrer consell té, naturalment, un caràcter marcadament tècnic; l'ordre del dia complet el trobaràs adjunt al final d'aquest text. A continuació et comento només els punts que, al meu parer, poden ser més del teu interès.

Translation (GOOGLE TRANSLATE) to English

Dear and beloved member of the university community,
Today is Monday carried out by the Governing Council last of my term as rector, the next day, May 6, as appropriate, there will be another election where the entire university community can choose new rector.
This last advice is, of course, a markedly technician complete agenda can be found attached to the end of this text. Then I said only the points that I believe may be of interest.

Machine Translation System

Input sentence (*Benvolguda i benvolgut ...*)

⇓

Translation system

⇓

Output sentence (*Dear and beloved ...*)

Machine Translation System

Input sentence (*Benvolguda i benvolgut ...*) **f**

$\Downarrow$

Translation system

$\Downarrow$

Output sentence (*Dear and beloved ...*) **e**

Statistical translation system

$$\mathbf{e} = \operatorname*{argmax}_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{f})$$

Noisy Channel Viewpoint

Input sentence (*Benvolguda i benvolgut ...*) **f**

⇓

Identity translation	⇐ Error signal (Noise)

⇓

Output sentence (*Dear and beloved ...*) **e**

Noisy Channel Viewpoint

Input sentence (*Benvolguda i benvolgut ...*) $\mathbf{f}$

$\Downarrow$

Identity translation

$\Leftarrow$ Error signal (Noise)

$\Downarrow$

Output sentence (*Dear and beloved ...*) $\mathbf{e}$

Bayes' theorem

$$\mathbf{e} = \operatorname*{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \operatorname*{argmax}_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e}) \cdot p(\mathbf{e})}{p(\mathbf{f})} = \operatorname*{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}) \cdot p(\mathbf{e})$$

Components

Optimization problem

$$\mathbf{e} = \underset{\mathbf{e}}{\operatorname{argmax}}\, p(\mathbf{f}|\mathbf{e}) \cdot p(\mathbf{e})$$

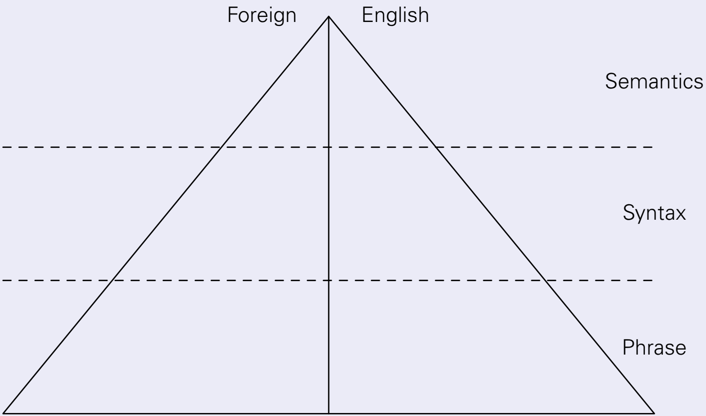
Required models

- $p(\mathbf{e})$ — language model
- $p(\mathbf{f}|\mathbf{e})$ — translation model

Input Sentence $\mathbf{f}$ $\Leftarrow$ Translation model $p(\mathbf{f}|\mathbf{e})$ $\Leftarrow$ Language model $p(\mathbf{e})$ $\Leftarrow$ Output sentence $\mathbf{e}$

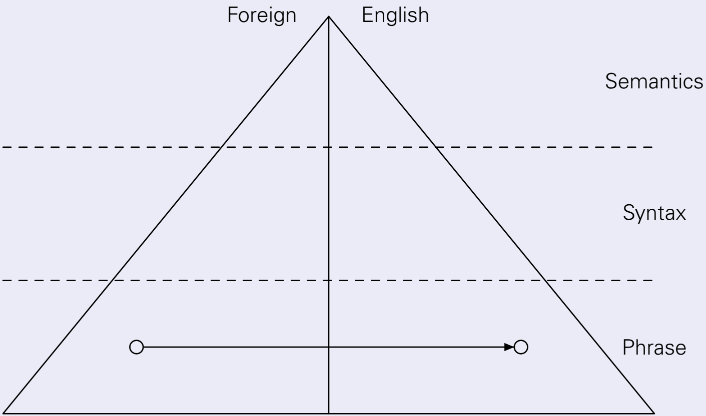
Translation Approach

Overview



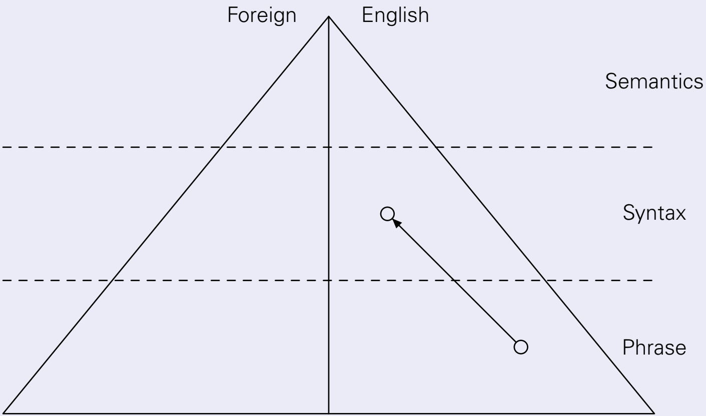
Translation Approach

Overview



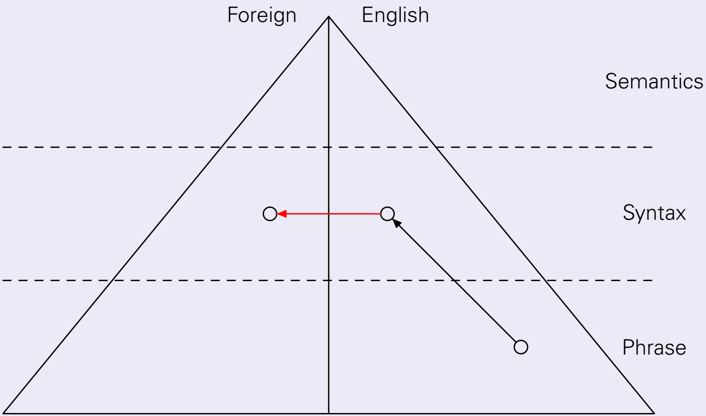
Translation Approach

Overview



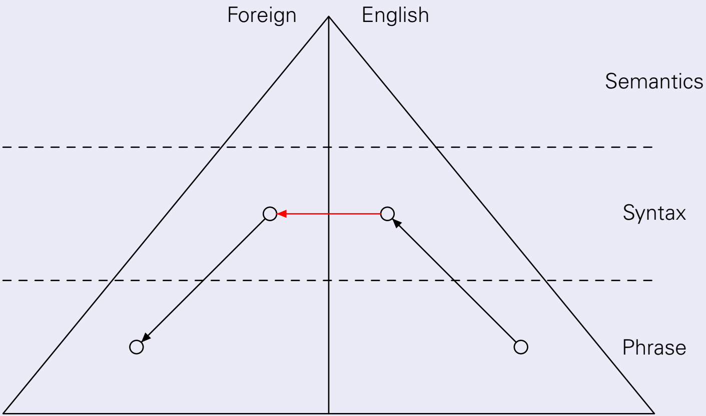
Translation Approach

Overview



Translation Approach

Overview



Why Syntax?

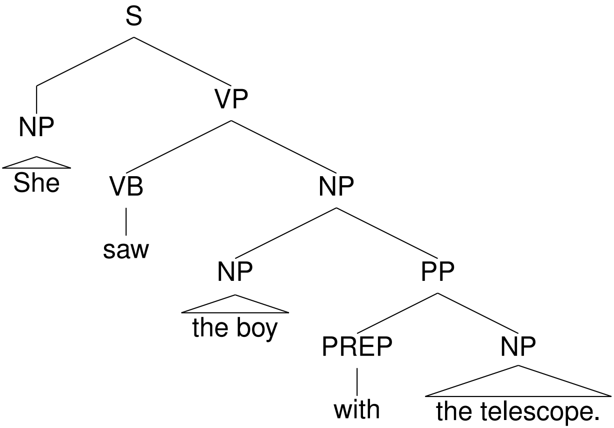
Example

She saw the boy with the telescope.

Why Syntax?

Example

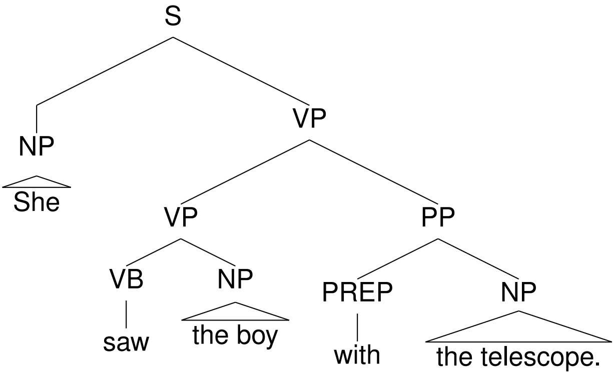
She saw the boy with the telescope.



Why Syntax?

Example

She saw the boy with the telescope.

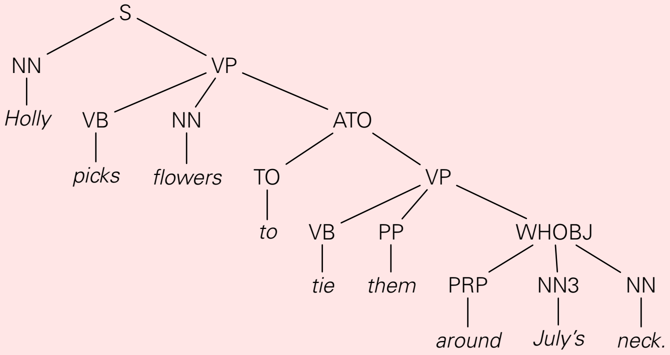


Syntactic Analysis

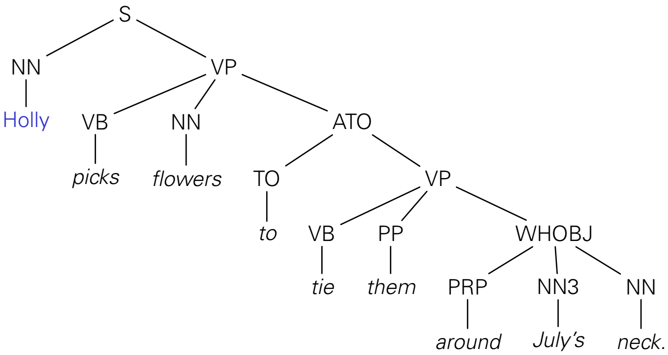
Output sentence

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

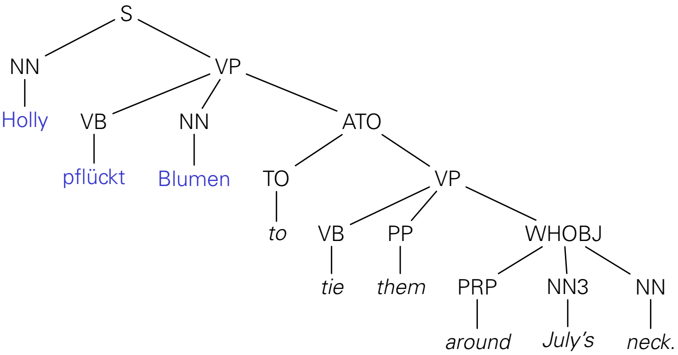
Parser output



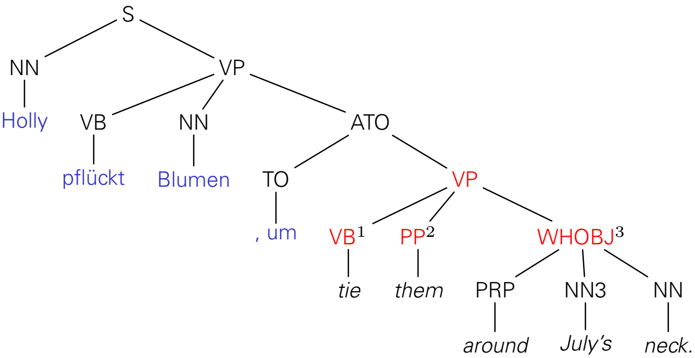
Syntax-based Machine Translation



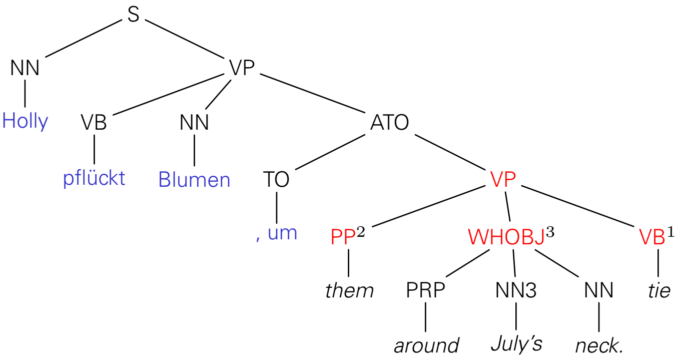
Syntax-based Machine Translation



Syntax-based Machine Translation



Syntax-based Machine Translation



Syntax-based Machine Translation

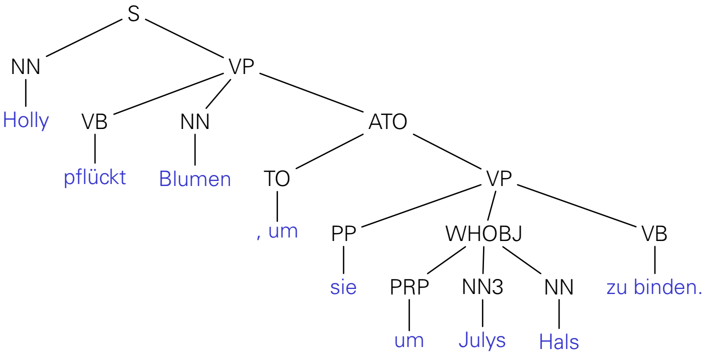


Table of Contents

Weight Structure

Definition

$(A, +, \cdot, 0, 1)$ is a (commutative) semiring if

- $(A, +, 0)$ and $(A, \cdot, 1)$ commutative monoids,
- $\cdot$ distributes over $+$, and
- $a \cdot 0 = 0$ for every $a \in A$.

Example

- $(\{0, 1\}, \max, \min, 0, 1)$ BOOLEAN semiring
- $(\mathbb{R}, +, \cdot, 0, 1)$ semiring of real numbers
- $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- any field, ring, etc.

Syntax

Definition

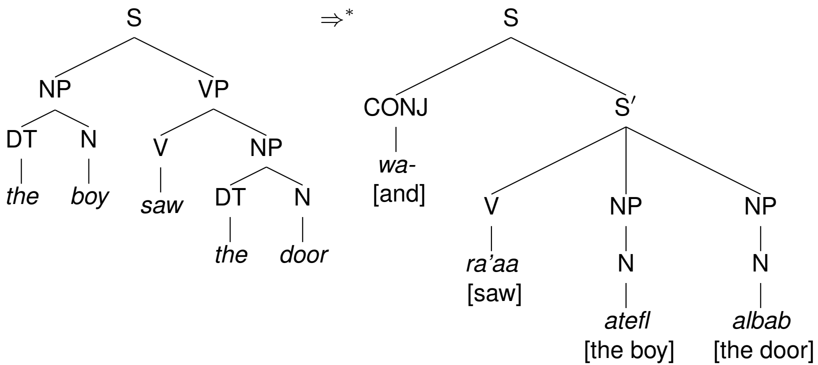
$(Q, \Sigma, \Delta, I, R)$ (weighted) extended (top-down) tree transducer (xtt)

- Q finite set of *states* (considered unary)
- Σ and Δ ranked alphabets
- $I\colon Q \to A$ *initial weight* distribution
- $R\colon Q(T_\Sigma(X)) \times T_\Delta(Q(X)) \to A$ *rule weight* assignment s.t.
  - $\mathrm{supp}(R)$ is finite
  - for every $(l, r) \in \mathrm{supp}(R)$ there is $k \in \mathbb{N}$ such that $l \in Q(C_\Sigma(X_k))$ and $r \in T_\Delta(Q(X_k))$
  - $\{l, r\} \not\subseteq Q(X)$ for every $(l, r) \in \mathrm{supp}(R)$

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 → 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

$$q(x_1) \to q_{\mathrm{S}}(x_1) \qquad (r_1)$$

$$q(x_1) \to \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), q_{\mathrm{S}}(x_1)) \qquad (r_2)$$

$$q_{\mathrm{S}}(\mathrm{S}(x_1, \mathrm{VP}(x_2, x_3))) \to \mathrm{S}'(q_{\mathrm{V}}(x_2), q_{\mathrm{NP}}(x_1), q_{\mathrm{NP}}(x_3)) \qquad (r_3)$$

$$q_{\mathrm{V}}(\mathrm{V}(\textit{saw})) \to \mathrm{V}(\textit{ra'aa}) \qquad (r_4)$$

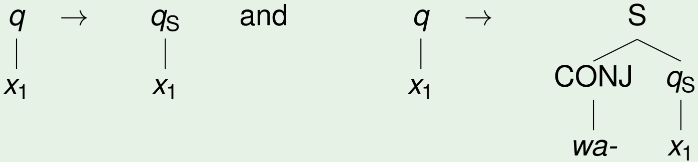
$$q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{boy}))) \to \mathrm{NP}(\mathrm{N}(\textit{atefl})) \qquad (r_5)$$

$$q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))) \to \mathrm{NP}(\mathrm{N}(\textit{albab})) \qquad (r_6)$$

Syntax — Example (cont'd)

Example

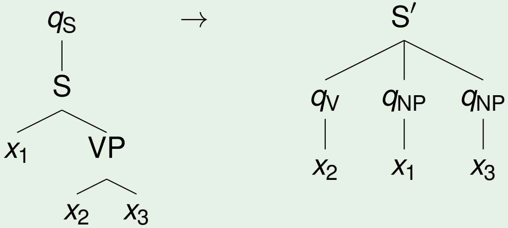
1. Nondeterminism and epsilon rules (rules r_1 and r_2)



Syntax — Example (cont'd)

Example

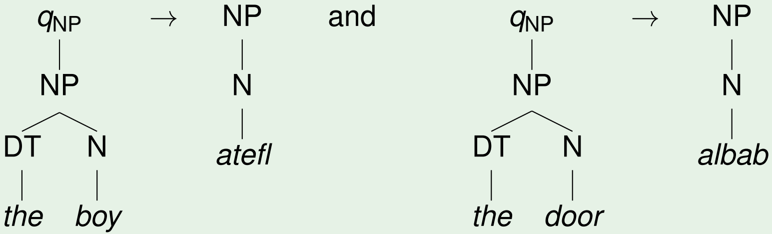
1. Nondeterminism and epsilon rules (rules r_1 and r_2)
2. Deep attachment of variables (rule r_3)



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)



Definition

Let $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$. Then $\xi \overset{a}{\Longrightarrow}_M \zeta$ if there exist

1. a rule $R(q(t), u) = a \neq 0$
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 ($t \in T_\Sigma$ and $u \in T_\Delta$):

$$\tau_M(t, u) = \sum_{\substack{q \in Q \\ q(t) \overset{a_1}{\Longrightarrow} \cdots \overset{a_n}{\Longrightarrow} u \\ \text{left-most derivation}}} I(q) \cdot a_1 \cdot \ldots \cdot a_n$$

Semantics

Definition

Let $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$. Then $\xi \stackrel{a}{\Longrightarrow}_M \zeta$ if there exist

1. a rule $R(q(t), u) = a \neq 0$
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 ($t \in T_\Sigma$ and $u \in T_\Delta$):

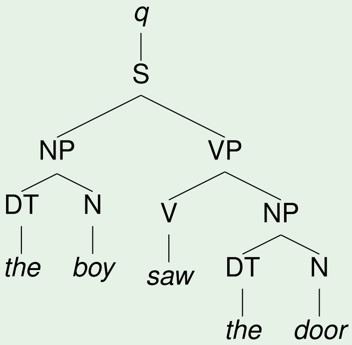
$$\tau_M(t, u) = \sum_{\substack{q \in Q \\ q(t) \stackrel{a_1}{\Longrightarrow} \cdots \stackrel{a_n}{\Longrightarrow} u \\ \text{left-most derivation}}} I(q) \cdot a_1 \cdot \ldots \cdot a_n$$

Semantics — Example

Rule

$q(x_1) \to \mathrm{S}(\mathrm{CONJ}(\textit{wa-}), q_{\mathrm{S}}(x_1))$

Example

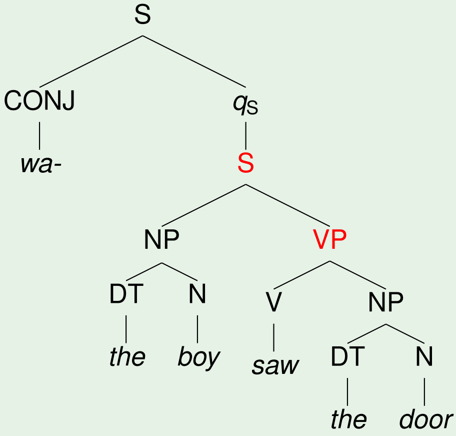


Semantics — Example

Rule

$$q_S(S(x_1, VP(x_2, x_3))) \to S'(q_V(x_2), q_{NP}(x_1), q_{NP}(x_3))$$

Example

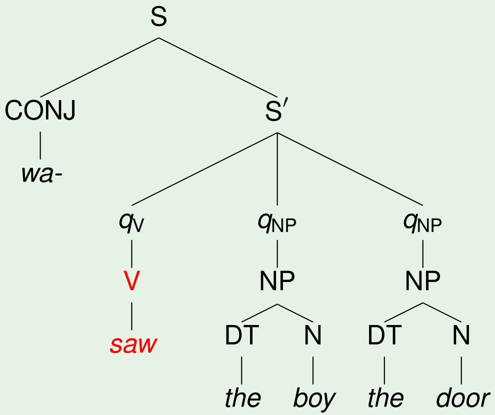


Semantics — Example

Rule

$q_{\mathrm{V}}(\mathrm{V}(\textit{saw})) \to \mathrm{V}(\textit{ra'aa})$

Example

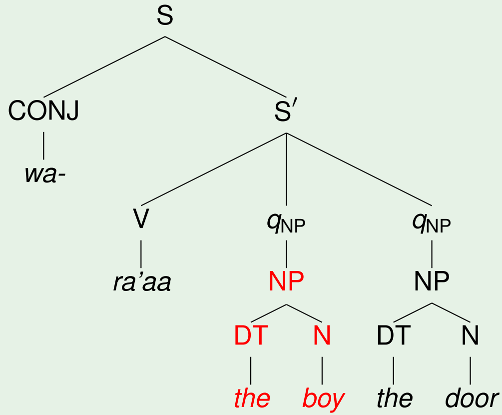


Semantics — Example

Rule

$q_{\text{NP}}(\text{NP}(\text{DT}(\textit{the}), \text{N}(\textit{boy}))) \to \text{NP}(\text{N}(\textit{atefl}))$

Example

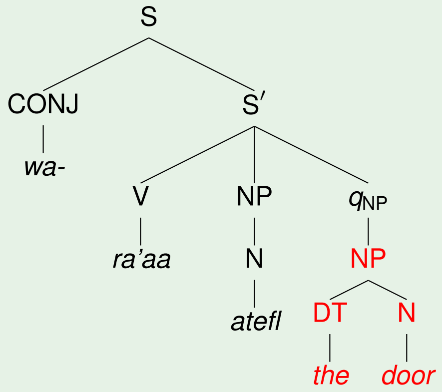


Semantics — Example

Rule

$q_{\mathrm{NP}}(\mathrm{NP}(\mathrm{DT}(\textit{the}), \mathrm{N}(\textit{door}))) \to \mathrm{NP}(\mathrm{N}(\textit{albab}))$

Example



Semantics — Example

Rule

Example

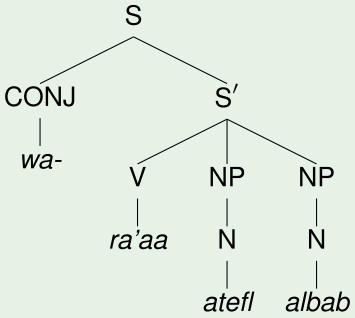


Table of Contents

Syntactic Restrictions

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 $q(x) \to u$

Definition

- tdtt: every left-hand side is of the form $q(\sigma(x_1, \ldots, x_k))$

Abbreviations

- ln-tdtt: linear non-deleting tdtt
- ln-xtt: linear non-deleting xtt

Syntactic Restrictions

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 $q(x) \to u$

Definition

- tdtt: every left-hand side is of the form $q(\sigma(x_1, \ldots, x_k))$

Abbreviations

- ln-tdtt: linear non-deleting tdtt
- ln-xtt: linear non-deleting xtt

Syntactic Restrictions

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 $q(x) \to u$

Definition

- tdtt: every left-hand side is of the form $q(\sigma(x_1, \ldots, x_k))$

Abbreviations

- ln-tdtt: linear non-deleting tdtt
- ln-xtt: linear non-deleting xtt

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

Wanted Expressivity

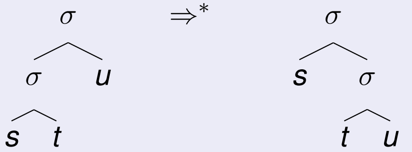
Criteria

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

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)

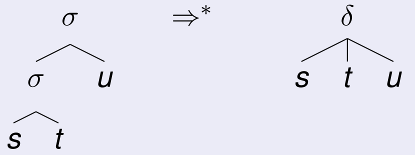


4. Can handle flattenings (ln-tdtt: no, ln-xtt: yes)
5. Preservation of Recognizability (ln-tdtt: yes, ln-xtt: yes)
6. Closure under composition (ln-tdtt: yes, ln-xtt: no)

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)



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

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

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

Criterion \ Model	ln-tdtt	ln-xtt
FST generalization	–	x
trainable	x	x
rotations	–	x
flattenings	–	x
pres. recog.	x	x
composition	x	–

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 (if the weight structure is not a ring)

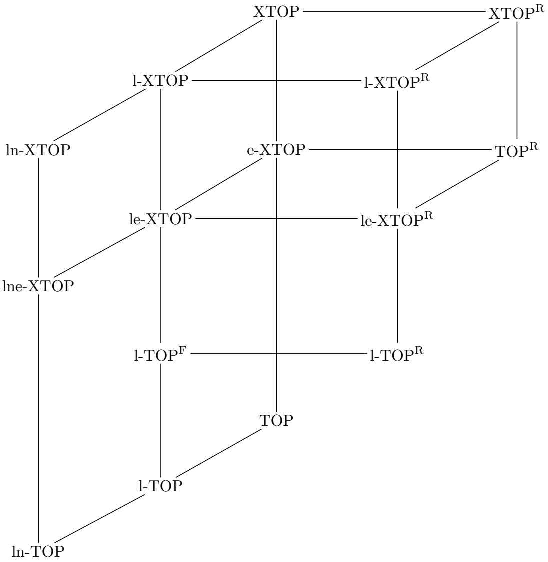
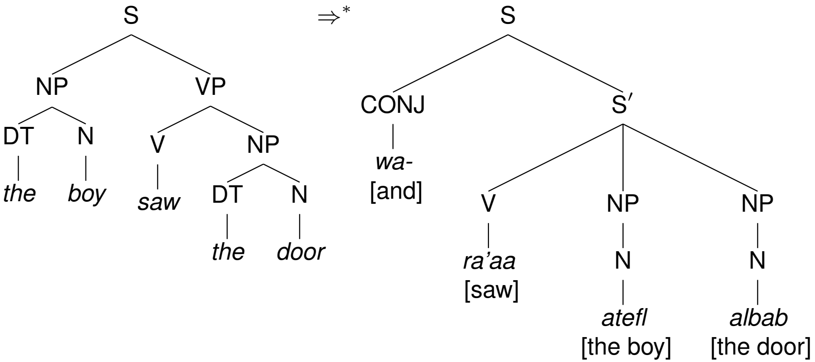


Table of Contents

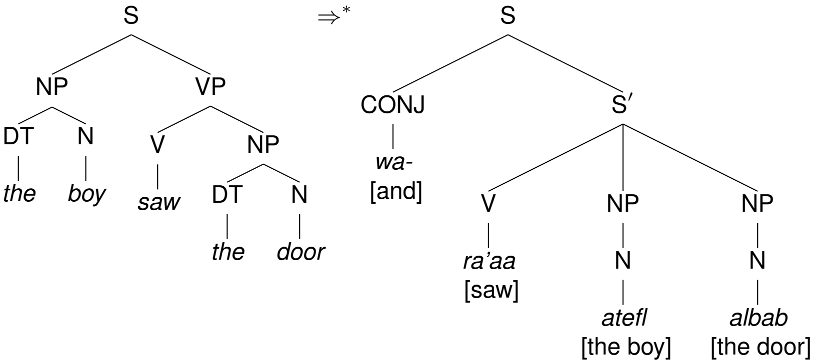
Training



References

- Galley, Hopkins, Knight, Marcu: What's in a translation rule? HLT-NAACL 2004
- Graehl, Knight, May: Training tree transducers. Comput. Ling. 34, 2008

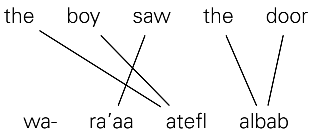
Training



References

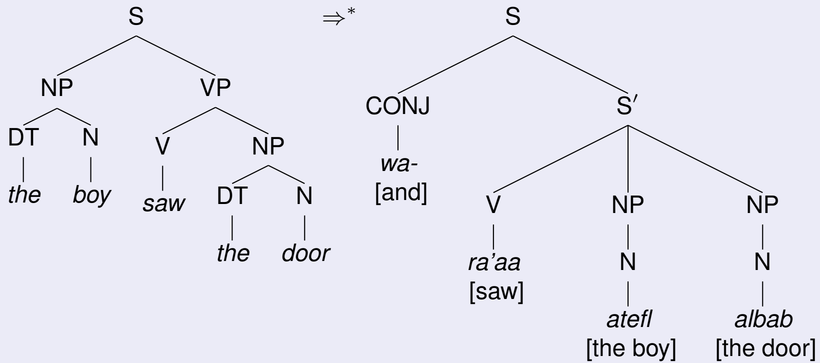
- GALLEY, HOPKINS, KNIGHT, MARCU: What's in a translation rule? HLT-NAACL 2004
- GRAEHL, KNIGHT, MAY: Training tree transducers. Comput. Ling. 34, 2008

Training (Cont'd)



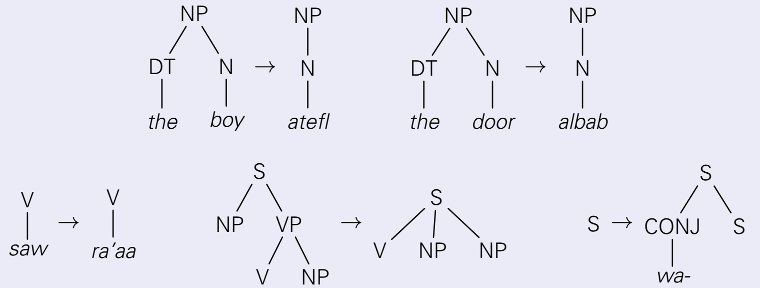
Alignment

Generate rules



Training (Cont'd)

Generated STSG rules

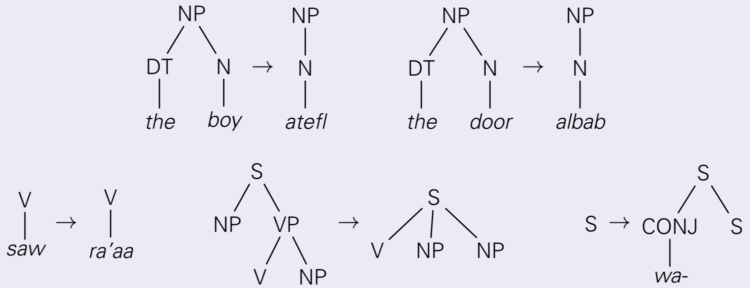


Conclusion

- In-xtt efficiently trainable
- Can we use states? Nonlinearity? Deletion? ...

Training (Cont'd)

Generated STSG rules



Conclusion

- In-xtt efficiently trainable
- Can we use states? Nonlinearity? Deletion? ...

Training (Cont'd)

Setting the weights

- count how often an obtained rule is used in a corpus (absolute frequency)
- set rule weight to relative frequency

Optimize the weights

- EM algorithm [Dempster, Laird, Rubin 1977]
- Gradient descent
- etc.

Composition

Theorem

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

Proof.

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

Reference

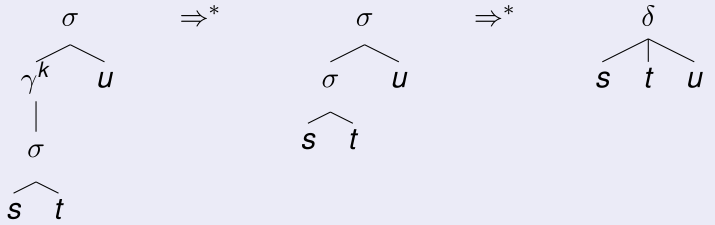
- ARNOLD, DAUCHET: Morphismes et bimorphismes d'arbres. Theoret. Comput. Sci. 20, 1982

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



□

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.

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.

Composition (Cont'd)

Problem

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

Questions

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

Reference

- ∼, Graehl, Hopkins, Knight: The power of extended top-down tree transducers. Siam J. Comput. 39, 2009

Composition (Cont'd)

Problem

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

Questions

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

Reference

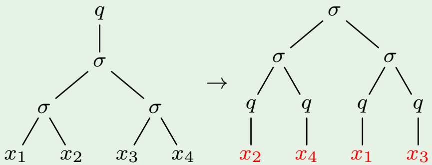
- $\sim$, GRAEHL, HOPKINS, KNIGHT: The power of extended top-down tree transducers. SIAM J. Comput. 39, 2009

Binarization

Definition

A xtt is binarized if there are at most 3 states per rule.

Example



Conclusions

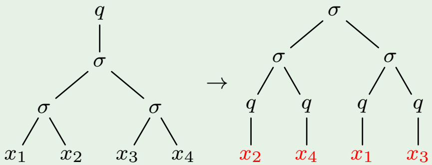
- linear xtt are not binarizable [AHO, ULLMAN 1972]
- What about non-linear xtt?

Binarization

Definition

A xtt is binarized if there are at most 3 states per rule.

Example

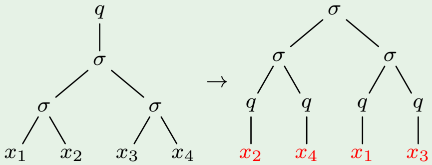


Conclusions

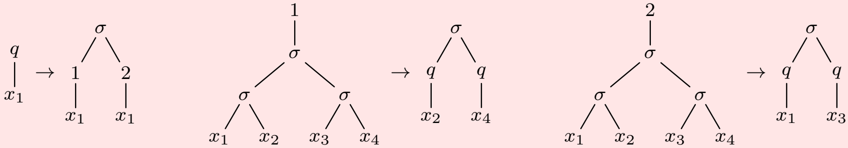
- linear xtt are not binarizable [Aho, Ullman 1972]
- What about non-linear xtt?

Binarization (Cont'd)

Example



Binarization



$\Rightarrow$ xtt can be binarized using non-linearity

Input Product

Definition

Given $\tau \colon T_\Sigma \times T_\Delta \to A$ and $\varphi \colon T_\Sigma \to A$, let $\varphi \lhd \tau \colon T_\Sigma \times T_\Delta \to A$

$$(\varphi \lhd \tau)(t, u) = \varphi(t) \cdot \tau(t, u)$$

Theorem

$\varphi \lhd \tau \in$ n-XTOP *for every* $\varphi \in$ Rec *and* $\tau \in$ n-XTOP

Input Product (Cont'd)

Parsing complexity

In-xtt M and input word w: $O(|M| \cdot |w|^{2\,\mathrm{rk}(M)+5})$

References

- ∼, SATTA: Parsing and translation algorithms based on weighted extended tree transducers. ATANLP 2010
- ∼: Why synchronous tree substitution grammars? HLT-NAACL 2010

Input Product (Cont'd)

Deleting xtt

How to obtain input products for deleting xtt?

Partial solutions

- for idempotent semirings
- for rings

but they do not work for the (non-linear) xtt obtained from binarization

References

- ∼: Input products for weighted extended top-down tree transducers. DLT 2010

Input Product (Cont'd)

Deleting xtt

How to obtain input products for deleting xtt?

Partial solutions

- for idempotent semirings
- for rings

but they do not work for the (non-linear) xtt obtained from binarization

References

- $\sim$: Input products for weighted extended top-down tree transducers. DLT 2010

Preservation of Recognizability

Definition

Given $\tau\colon T_\Sigma \times T_\Delta \to A$ and $\varphi\colon T_\Sigma \to A$, let $\tau(\varphi), \text{range}(\tau)\colon T_\Delta \to A$

$$\big(\tau(\varphi)\big)(u) = \sum_{t \in T_\Sigma} \varphi(t) \cdot \tau(t, u)$$

$$\big(\text{range}(\tau)\big)(u) = \sum_{t \in T_\Sigma} \tau(t, u)$$

References

- FÜLÖP, ∼, VOGLER: Backward and forward application of extended tree series transformations. WATA 2010
- MAY, KNIGHT, VOGLER: Efficient inference through cascades of weighted tree transducers. ACL 2010

Preservation of Recognizability (Cont'd)

Theorem

Given $\tau \colon T_\Sigma \times T_\Delta \to A$ *and* $\varphi \colon T_\Sigma \to A$

$$\tau(\varphi) = \mathrm{range}(\varphi \triangleleft \tau)$$

Proof.

$$\begin{aligned}\big(\tau(\varphi)\big)(u) &= \sum_{t \in T_\Sigma} \varphi(t) \cdot \tau(t, u) \\ &= \sum_{t \in T_\Sigma} (\varphi \triangleleft \tau)(t, u) \\ &= \big(\mathrm{range}(\varphi \triangleleft \tau)\big)(u)\end{aligned}$$

□

Table of Contents

Tiburon

Features

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

Algorithms

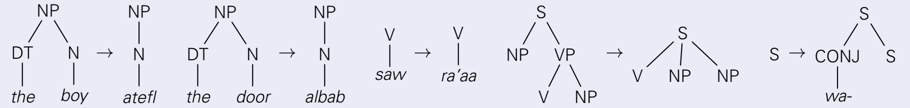
- Application of xtt to input tree/language
- Backward application of xtt to output language
- Composition (for some xtt)
- ...

Reference

- MAY, KNIGHT: Tiburon: A Weighted Tree Automata Toolkit. CIAA 2006

Tiburon (Cont'd)

Generated STSG rules



Example

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

Summary

Criteria

(a) Generalize FST; in particular, epsilon-transitions

(b) Efficient training

(c) Handles rotation

(d) Closed under composition

(e) Preserves recognizability

Models

Model \ Criterion	(a)	(b)	(c)	(d)	(e)
Top-down tree transducer	–	x	–	x	x
Synchronous context-free grammar	x	x	–	x	x
Synchronous tree substitution grammar	x	x	x	–	x
Synchronous tree adjoining grammar	x	x	x	–	–
Multi bottom-up tree transducer	x	?	x	x	–

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
- $\sim$, GRAEHL, HOPKINS, KNIGHT: The power of extended top-down tree transducers. SIAM J. Comput. 2009

Thank You for your attention!