Parsing and Translation Algorithms based on Weighted Extended Tree Transducers

Andreas Maletti[1] and Giorgio Satta[2]

[1] Universitat Rovira i Virgili, Tarragona, Spain

[2] University of Padua, Padua, Italy

email: andreas.maletti@urv.cat

Uppsala, Sweden — July 16, 2010

Background

Recall

Weighted tree transducers of considerable interest nowadays in statistical, syntax-based machine translation

But

- parsing and translation with tree transducers traditionally defined for input trees
- whereas NLP applications typically process strings

Goals

Theorem

BAR-HILLEL (1964) showed that the intersection of a context-free language with a regular language is context-free.

Use

Foundation of tabular parsing algorithms for CFGs

Our goals

- extension to weighted tree transducers
  ($\rightarrow$ string alignment/parsing and string translation algorithms)
- computing interesting statistical parameters
  (maximum-likelihood unsupervised probabilities, partition function)
- asymptotically improve computational complexity of the algorithm

Goals

Theorem

BAR-HILLEL (1964) showed that the intersection of a context-free language with a regular language is context-free.

Use

Foundation of tabular parsing algorithms for CFGs

Our goals

- extension to weighted tree transducers
  ($\rightarrow$ string alignment/parsing and string translation algorithms)
- computing interesting statistical parameters
  (maximum-likelihood unsupervised probabilities, partition function)
- asymptotically improve computational complexity of the algorithm

Goals

Theorem

BAR-HILLEL *(1964) showed that the intersection of a context-free language with a regular language is context-free.*

Use

Foundation of tabular parsing algorithms for CFGs

Our goals

- extension to weighted tree transducers
  ($\rightarrow$ string alignment/parsing and string translation algorithms)
- computing interesting statistical parameters
  (maximum-likelihood unsupervised probabilities, partition function)
- asymptotically improve computational complexity of the algorithm

Table of Contents

Extended Tree Transducer

References

- ARNOLD, DAUCHET: Bi-transductions de forêts. ICALP 1976
- ENGELFRIET: Bottom-up and top-down tree transformations—a comparison. *Math. Syst. Theory 9*, 1975
- GRAEHL, KNIGHT, MAY: Training tree transducers. *Computational Linguistics 34*, 2008
- $\sim$, GRAEHL, HOPKINS, KNIGHT: The power of extended top-down tree transducers. *SIAM J. Comput. 39*, 2009

Syntax

Definition

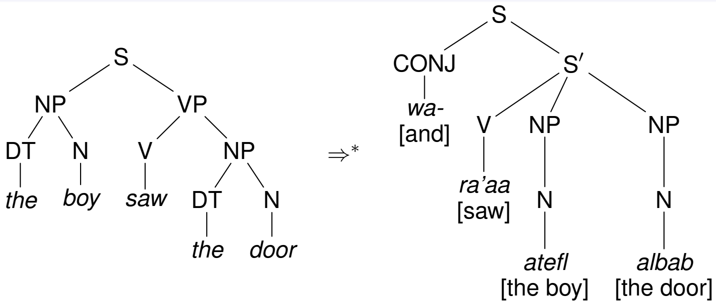
$M = (Q, \Sigma, \Delta, I, R)$ extended tree transducer (xtt)

- Q finite set of *states*
- Σ and Δ ranked alphabets
- $I \colon Q \to \mathbb{R}$ *initial weight* distribution
- $R \colon \bigcup_{k \geq 0} Q \times C_\Sigma(X_k) \times Q^k \times C_\Delta(X_k) \to \mathbb{R}$ is a *rule weight assignment* such that
  - $\mathrm{supp}(R)$ is finite and
  - $\{l, r\} \not\subseteq X$ for every $(q, l, w, r) \in \mathrm{supp}(R)$.

References

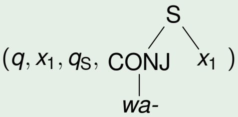
- Arnold, Dauchet: Bi-transductions de forêts. ICALP 1976
- Graehl, Knight: Training tree transducers. HLT-NAACL 2004

Syntax — Example

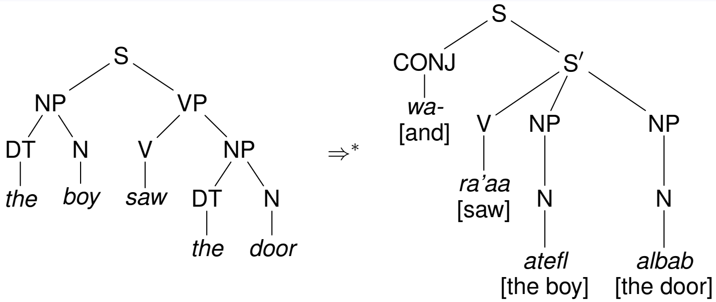


Example

States $\{q, q_S, q_V, q_{NP}\}$ of which only q is initial

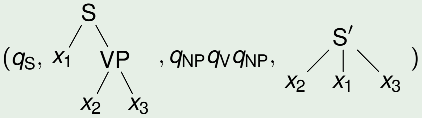


Syntax — Example

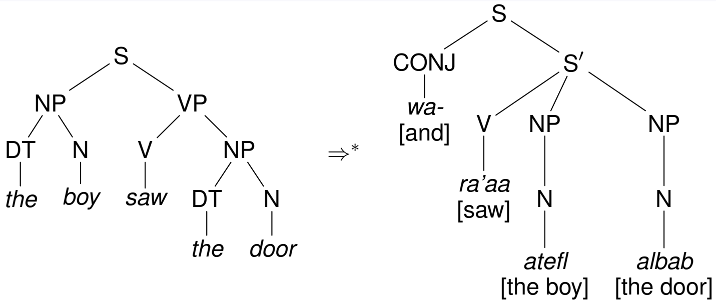


Example

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



Syntax — Example

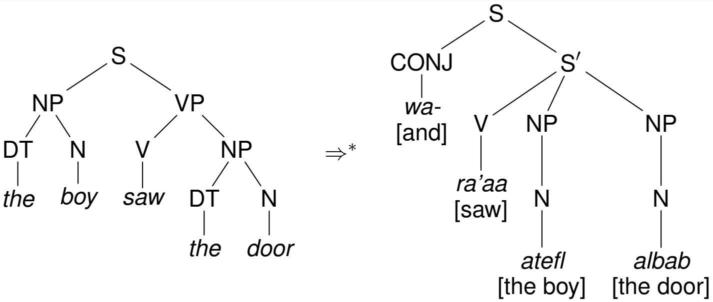


Example

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

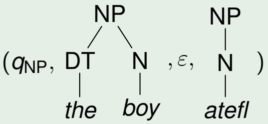
$$\left(q_{\mathrm{V}},\ \begin{array}{c}\mathrm{V}\\ |\\ \textit{saw}\end{array},\ \varepsilon,\ \begin{array}{c}\mathrm{V}\\ |\\ \textit{ra'aa}\end{array}\right)$$

Syntax — Example

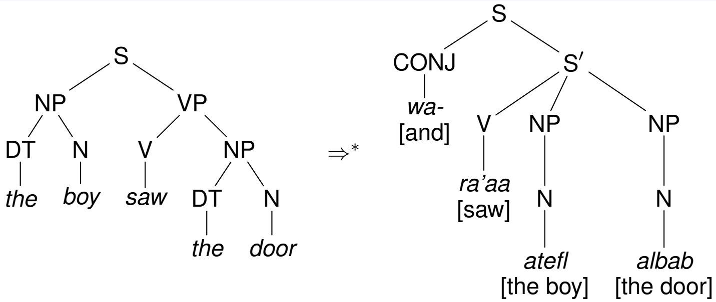


Example

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

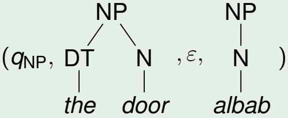


Syntax — Example



Example

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



Semantics

Definition

match(t) is

$$\{(l, t_1, \ldots, t_k) \mid l \in C_\Sigma(X_k), t_1, \ldots, t_k \in T_\Sigma(X) \colon t = l[t_1, \ldots, t_k]\}$$

Semantics

Definition

Let $q, p_1, \dots, p_n \in Q$, $t \in T_\Sigma(X_n)$, and $u \in T_\Delta(X_n)$.

$$M_{p_i}^{p_1 \cdots p_n}(x_i, x_i) = 1$$

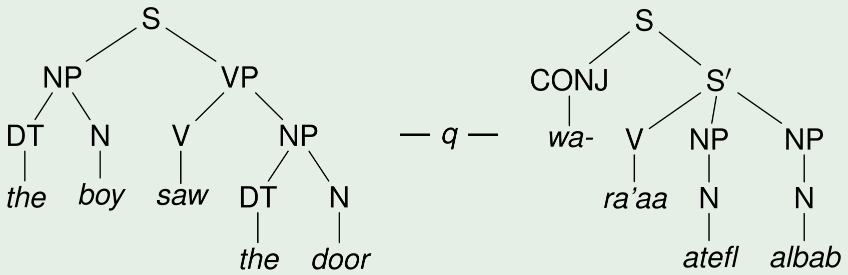
$$M_q^{p_1 \cdots p_n}(t, u) = \sum_{\substack{(l, t_1, \dots, t_k) \in \mathrm{match}(t) \\ (r, u_1, \dots, u_k) \in \mathrm{match}(u) \\ q_1, \dots, q_k \in Q}} R(q, l, q_1 \cdots q_k, r) \cdot \prod_{i=1}^{k} M_{q_i}^{p_1 \cdots p_n}(t_i, u_i)$$

Then

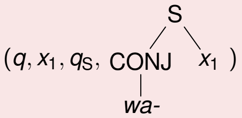
$$M(t, u) = \sum_{q \in Q} I(q) \cdot M_q(t, u)$$

Semantics — Example

Example

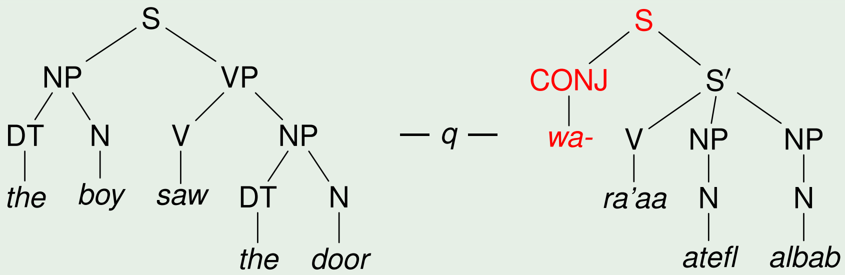


Used rule

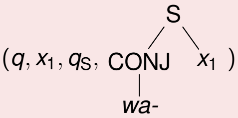


Semantics — Example

Example

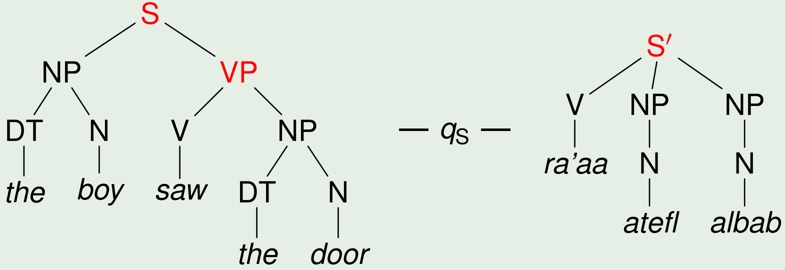


Used rule

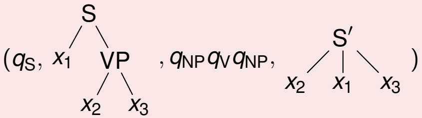


Semantics — Example

Example

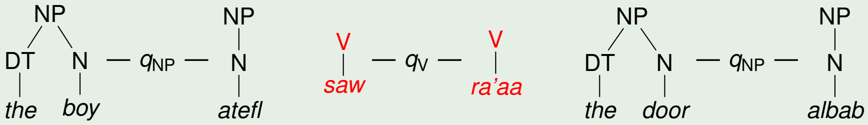


Used rule



Semantics — Example

Example

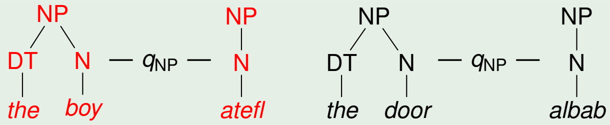


Used rule

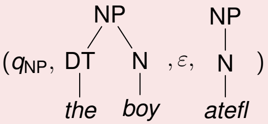
$$\left(q_V,\ \begin{array}{c}\mathrm{V}\\ |\\ \textit{saw}\end{array},\ \varepsilon,\ \begin{array}{c}\mathrm{V}\\ |\\ \textit{ra'aa}\end{array}\right)$$

Semantics — Example

Example

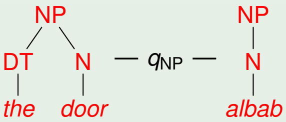


Used rule



Semantics — Example

Example



Used rule

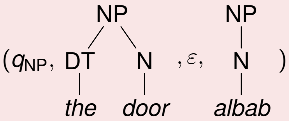


Table of Contents

Definition

Definition

Let $T\colon T_\Sigma \times T_\Delta \to \mathbb{R}$ and $L\colon T_\Sigma \to \mathbb{R}$.

$$(L \triangleleft T)(t, u) = L(t) \cdot T(t, u)$$

Application

- parsing
- forward and backward application

Input Restriction

References

- BAR-HILLEL, PERLES, SHAMIR: On formal properties of simple phrase structure grammars. *Language and Information: Selected Essays on their Theory and Application*, Addison Wesley. 1964 [for CFG]
- NEDERHOF AND GIORGIO SATTA: Probabilistic parsing as intersection. IWPT 2003 [for SCFG]
- $\sim$, SATTA: Parsing algorithms based on tree automata. IWPT 2009 [for RTG]
- NEDERHOF: Weighted parsing of trees. IWPT 2009 [for STAG]
- $\sim$: Input products for weighted extended top-down tree transducers. DLT 2010 [for general xtt]

Weighted String Automaton

Definition

weighted string automaton (wsa) $A = (P, \Gamma, J, \mu, F)$ where

- P finite set of *states*
- Γ alphabet of *input symbols*
- $J, F \colon P \to \mathbb{R}$ *initial* and *final distribution*
- $\mu \colon \Gamma \to \mathbb{R}^{P \times P}$ *transition weights*

Definition (Semantics)

- Extend μ to a homomorphism $\mu^* \colon \Gamma^* \to \mathbb{R}^{P \times P}$
- $A(w) = J\mu^*(w)F$
- $\mu^*(w)$ can be computed in time $O(|w| \cdot |P|^3)$

Construction

Input rule

$$R\left(q_{\mathrm{NP}},\ \begin{array}{c}\mathrm{NP}\\ \diagup\quad\diagdown\\ \mathrm{DT}\qquad\mathrm{N}\\ |\qquad\quad|\\ \textit{the}\quad\textit{door}\end{array},\ \varepsilon,\ \begin{array}{c}\mathrm{NP}\\ |\\ \mathrm{N}\\ |\\ \textit{albab}\end{array}\right) = c$$

Constructed rule

$$R'\left(\ {\color{red}\langle p, q_{\mathrm{NP}}, p'\rangle}\ ,\ \begin{array}{c}\mathrm{NP}\\ \diagup\quad\diagdown\\ \mathrm{DT}\qquad\mathrm{N}\\ |\qquad\quad|\\ \textit{the}\quad\textit{door}\end{array},\ \varepsilon,\ \begin{array}{c}\mathrm{NP}\\ |\\ \mathrm{N}\\ |\\ \textit{albab}\end{array}\right) = c \cdot \mu^*(\textit{the door})_{p,p'}$$

Result

Theorem

For every wsa $A = (P, \Gamma, J, \mu, F)$ *and xtt* $M = (Q, \Sigma, \Delta, I, R)$ *with* $\Gamma = \Sigma_0$*, the input restriction* $A \triangleleft M$ *can be constructed in time*

$$O(|\mathrm{supp}(R)| \cdot \mathrm{len}(M) \cdot |P|^{2\,\mathrm{rk}(M)+5})$$

- $\mathrm{len}(M)$ *longest span in* l *of a rule* $(q, l, w, r) \in \mathrm{supp}(R)$
- $\mathrm{rk}(M)$ *largest* k *such that*

$$\mathrm{supp}(R) \cap (Q \times C_\Sigma(X_k) \times Q^k \times C_\Delta(X_k)) \neq \emptyset$$

Universitat Rovira i Virgili
La universitat pública de Tarragona

Result (cont'd)

Notes

- no direct correspondence between runs
  [some runs of the wsa can be collapsed]
- but consistent with *k* best run extraction
  [the best run of the restriction is better than the product of the best individual runs]
- direct correspondence between runs obtainable at expense of additional states

Result (cont'd)

Notes

- no direct correspondence between runs
  [some runs of the wsa can be collapsed]
- but consistent with *k* best run extraction
  [the best run of the restriction is better than the product of the best individual runs]
- direct correspondence between runs obtainable at expense of additional states

Result (cont'd)

Notes

- no direct correspondence between runs
  [some runs of the wsa can be collapsed]
- but consistent with *k* best run extraction
  [the best run of the restriction is better than the product of the best individual runs]
- direct correspondence between runs obtainable at expense of additional states

Table of Contents

Factorization

Motivation

- rk(M) essential part in the complexity of xtt operations [like input restriction]

$$O(|\text{supp}(R)| \cdot \text{len}(M) \cdot |P|^{2\,\text{rk}(M)+5})$$

Factorization

- reduce rk(M) by decomposing rules
- maximal decomposition prefered

Factorization (cont'd)

References

- ZHANG, HUANG, GILDEA, KNIGHT: Synchronous binarization for machine translation. HLT-NAACL 2006 [for SCFG]
- GILDEA, SATTA, ZHANG: Factoring synchronous grammars by sorting. CoLing/ACL 2006 [for SCFG]
- NESSON, SATTA, SHIEBER: Optimal *k*-arization of synchronous tree-adjoining grammar. ACL 2008 [for STAG]
- GILDEA: Optimal parsing strategies for linear context-free rewriting systems. HLT-NAACL 2010 [for LCFRS]

Maximal Factorization

Common advertisement slogan

- Optimal k-arization
- Optimal parsing strategy

But

- factorization is just one way to reduce rk(M)
- obtained "optimal" rank is not optimal for the given transformation
- rk(M) is just one parameter to the parsing complexity

Conclusion

- optimal k-arization $\neq$ maximal factorization
- optimal parsing strategy $\neq$ maximal factorization

Maximal Factorization

Common advertisement slogan

- Optimal k-arization
- Optimal parsing strategy

But

- factorization is just one way to reduce rk(M)
- obtained “optimal” rank is not optimal for the given transformation
- rk(M) is just one parameter to the parsing complexity

Conclusion

- optimal k-arization $\neq$ maximal factorization
- optimal parsing strategy $\neq$ maximal factorization

Main Definition

Definition

xtt $M' = (Q', \Sigma, \Delta, I', R')$ is a factorization of the xtt $M = (Q, \Sigma, \Delta, I, R)$ if

- $I'(q) = I(q)$ for every $q \in Q$
- $I'(q) = 0$ for every $q \in Q' \setminus Q$
- $(M')_q^{p_1 \cdots p_n}(t, u) = M_q^{p_1 \cdots p_n}(t, u)$ for every $q, p_1, \ldots, p_n \in Q$, $t \in T_\Sigma(X_n)$, and $u \in T_\Delta(X_n)$.

Theorem

If M' is a factorization of M, then M' and M are equivalent.

Main Definition

Definition

xtt $M' = (Q', \Sigma, \Delta, I', R')$ is a factorization of the xtt $M = (Q, \Sigma, \Delta, I, R)$ if

- $I'(q) = I(q)$ for every $q \in Q$
- $I'(q) = 0$ for every $q \in Q' \setminus Q$
- $(M')_q^{p_1 \cdots p_n}(t, u) = M_q^{p_1 \cdots p_n}(t, u)$ for every $q, p_1, \ldots, p_n \in Q$, $t \in T_\Sigma(X_n)$, and $u \in T_\Delta(X_n)$.

Theorem

If M' is a factorization of M, then M' and M are equivalent.

Main Definition (cont'd)

Theorem

The relation *'is a factorization of'* *is a pre-order. Moreover, in an additively cancellative semiring it is a partial order.*

Note

- maximality and optimality are now wrt. this pre-order
- then: maximal factorization $=$ optimal k-arization

Mind

- not binarizable: there exists no equivalent binary xtt
- optimal xtt not binary: there exists no binary factorization

Main Definition (cont'd)

Theorem

The relation 'is a factorization of' is a pre-order. Moreover, in an additively cancellative semiring it is a partial order.

Note

- maximality and optimality are now wrt. this pre-order
- then: maximal factorization $=$ optimal k-arization

Mind

- not binarizable: there exists no equivalent binary xtt
- optimal xtt not binary: there exists no binary factorization

Main Definition (cont'd)

Theorem

The relation 'is a factorization of' is a pre-order. Moreover, in an additively cancellative semiring it is a partial order.

Note

- maximality and optimality are now wrt. this pre-order
- then: maximal factorization $=$ optimal k-arization

Mind

- not binarizable: there exists no equivalent binary xtt
- optimal xtt not binary: there exists no binary factorization

Construction

Example

$$\left(q,\ \sigma(x_1, \sigma(x_3, x_2)),\ q_1 q_2 q_3,\ \gamma(\sigma(x_1, \gamma(\sigma(x_2, x_3))))\right)$$

Constructed rules

Construction

Example

$$\left(q,\ \sigma\big(x_1, {\color{red}\sigma(x_3, x_2)}\big),\ q_1 q_2 q_3,\ \gamma\Big(\sigma\big(x_1, {\color{red}\gamma(\sigma(x_2, x_3))}\big)\Big)\right)$$

Constructed rules

Construction

Example

$$\left(q,\ \sigma(x_1, \sigma(x_3, x_2)),\ q_1 q_2 q_3,\ \gamma(\sigma(x_1, \gamma(\sigma(x_2, x_3))))\right)$$

Constructed rules

$$\left(q,\ \sigma(x_1, x_2),\ q_1 \star,\ \gamma(\sigma(x_1, x_2))\right) \qquad \left(\star,\ \sigma(x_2, x_1),\ q_2 q_3,\ \gamma(\sigma(x_1, x_2))\right)$$

Construction (cont'd)

Notes

- full construction in the paper
- runs in linear time
- returns maximal (meaningful) factorization
- returned xtt is rank-optimal (among all factorizations)

Thank You for your attention!