TECHNISCHE UNIVERSITÄT DRESDEN

Learning Deterministically Recognizable Tree Series — Revisited

Andreas Maletti

Θεσσαλονίκη, 22 May 2007

00 Motivation

Goal

- Given $\psi : T_\Sigma \to A$ with $(A, +, \cdot, 0, 1)$ semifield
- Learn finite representation (here: deterministic wta) of ψ, if possible
- Access to ψ is granted by a certain form of teacher (oracle)

00 Table of Contents

Contents

01 Notation

Trees

- T_Σ: trees over ranked alphabet Σ
- C_Σ: contexts (trees with exactly one occurrence of $\Box$) over Σ
- $\mathrm{size}(t)$: number of nodes of a tree t

Tree series

- tree series: mapping of type $T_\Sigma \to A$
- we write (ψ, t) for $\psi(t)$ with $\psi \colon T_\Sigma \to A$
- $A\langle\langle T_\Sigma \rangle\rangle$: set of all mappings of type $T_\Sigma \to A$

02 Table of Contents

02 Syntax

Definition (Borchardt and Vogler ’03)

$(Q, \Sigma, \mathcal{A}, \mu, F)$ is a weighted tree automaton (wta)

- Q is a finite nonempty set (states)
- Σ is a ranked alphabet (of input symbols)
- $\mathcal{A} = (A, +, \cdot, 0, 1)$ is a semifield (of weights)
- $\mu = (\mu_k)_{k \geq 0}$ with $\mu_k : \Sigma^{(k)} \to A^{Q^k \times Q}$ (called tree representation)
- $F \subseteq Q$ (final states)

Definition

wta $(Q, \Sigma, \mathcal{A}, \mu, F)$ is deterministic if for every $\sigma \in \Sigma^{(k)}$ and $w \in Q^k$ there exists at most one $q \in Q$ such that $\mu_k(\sigma)_{w,q} \neq 0$.

02 Example wta

$Q = \{\text{S}, \text{VP}, \text{NP}, \text{NN}, \text{ADJ}, \text{VB}\}$ and $F = \{\text{S}\}$

$\text{Alice} \xrightarrow{0.5} \text{NN}$ $\quad$ $\text{Bob} \xrightarrow{0.5} \text{NN}$ $\quad$ $\text{loves} \xrightarrow{0.5} \text{VB}$ $\quad$ $\text{hates} \xrightarrow{0.5} \text{VB}$

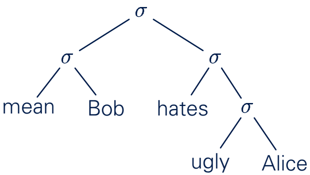
$\text{ugly} \xrightarrow{0.25} \text{ADJ}$ $\quad$ $\text{nice} \xrightarrow{0.25} \text{ADJ}$ $\quad$ $\text{mean} \xrightarrow{0.25} \text{ADJ}$ $\quad$ $\text{tall} \xrightarrow{0.25} \text{ADJ}$

$\sigma(\text{NN}, \text{VP}) \xrightarrow{0.5} \text{S}$ $\quad$ $\sigma(\text{NP}, \text{VP}) \xrightarrow{0.5} \text{S}$ $\quad$ $\sigma(\text{VB}, \text{NN}) \xrightarrow{0.5} \text{VP}$

$\sigma(\text{VB}, \text{NP}) \xrightarrow{0.5} \text{VP}$ $\quad$ $\sigma(\text{ADJ}, \text{NN}) \xrightarrow{0.5} \text{NP}$ $\quad$ $\sigma(\text{ADJ}, \text{NP}) \xrightarrow{0.5} \text{NP}$

02 Computation using wta

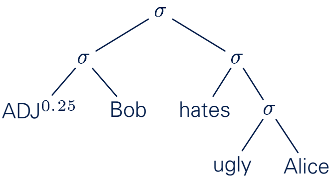
Example



mean $\xrightarrow{0.25}$ ADJ

02 Computation using wta

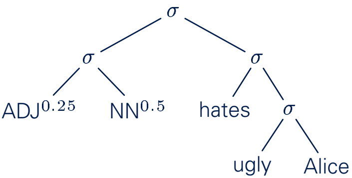
Example

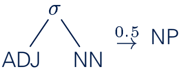


Bob $\xrightarrow{0.5}$ NN

02 Computation using wta

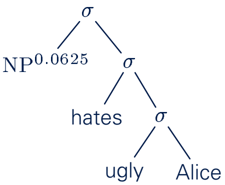
Example





02 Computation using wta

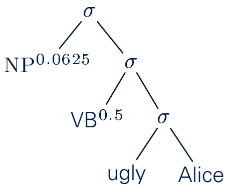
Example



hates $\xrightarrow{0.5}$ VB

02 Computation using wta

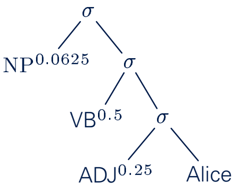
Example



ugly $\xrightarrow{0.25}$ ADJ

02 Computation using wta

Example



$\text{Alice} \xrightarrow{0.5} \text{NN}$

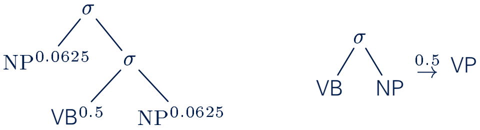
02 Computation using wta

Example

TECHNISCHE UNIVERSITÄT DRESDEN

02 Computation using wta

Example



02 Computation using wta

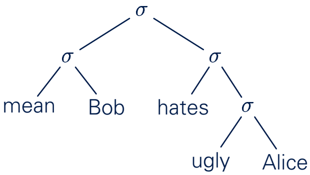
Example

02 Computation using wta

Example

$$S^{0.000244140625}$$

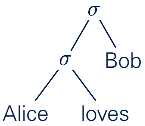
So the tree



is accepted with weight 0.000244140625.

02 Computation using wta (cont'd)

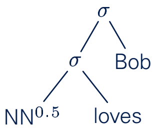
Example



$$\text{Alice} \xrightarrow{0.5} \text{NN}$$

02 Computation using wta (cont'd)

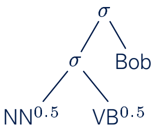
Example



$$\text{loves} \overset{0.5}{\rightarrow} \text{VB}$$

02 Computation using wta (cont'd)

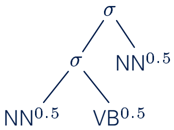
Example



$\text{Bob} \stackrel{0.5}{\to} \text{NN}$

02 Computation using wta (cont'd)

Example

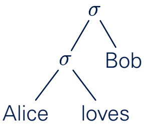


?

02 Computation using wta (cont'd)

Example

So the tree



is rejected (accepted with weight 0).

02 Deterministically recognizable

Definition

A tree series $\psi \in A\langle\langle T_\Sigma \rangle\rangle$ is deterministically recognizable if there exists a deterministic wta accepting ψ.

03 Table of Contents

03 Definition

In the sequel, let $\psi \in A\langle\langle T_\Sigma \rangle\rangle$ with $A = (A, +, \cdot, 0, 1)$ a semifield.

Definition (Borchardt '03)

Two trees $t, u \in T_\Sigma$ are equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C_\Sigma$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

This equivalence relation is denoted by $\equiv$.

03 Definition

In the sequel, let $\psi \in A\langle\langle T_\Sigma \rangle\rangle$ with $A = (A, +, \cdot, 0, 1)$ a semifield.

Definition (Borchardt '03)

Two trees $t, u \in T_\Sigma$ are equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C_\Sigma$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

This equivalence relation is denoted by $\equiv$.

Example

Let $\psi = \text{size}$ and $\mathcal{A} = (\mathbb{Z} \cup \{\infty\}, \min, +, \infty, 0)$. Then $t \equiv u$ for every $t, u \in T_\Sigma$ because with $a = \text{size}(u) - \text{size}(t)$

$$a + \text{size}(c[t]) = a + \text{size}(c) - 1 + \text{size}(t) = \text{size}(u) + \text{size}(c) - 1 = \text{size}(c[u])$$

03 Myhill-Nerode theorem

Lemma (Borchardt '03)

$\equiv$ *is a congruence on* (T_Σ, Σ).

03 Myhill-Nerode theorem

Lemma (Borchardt '03)

$\equiv$ is a congruence on (T_Σ, Σ).

Theorem (Borchardt '03)

The following are equivalent:

- *ψ is deterministically recognizable.*
- *$\equiv$ has finite index.*

Note: The implementation of $\equiv$ yields a minimal deterministic wta accepting ψ.

03 Approximating the Myhill-Nerode relation

Definition

Let $C \subseteq C_\Sigma$. Two trees $t, u \in T_\Sigma$ are C-equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

The C-equivalence relation is denoted by $\equiv_C$.

03 Approximating the Myhill-Nerode relation

Definition

Let $C \subseteq C_\Sigma$. Two trees $t, u \in T_\Sigma$ are C-equivalent if there exists $a \in A \setminus \{0\}$ such that for every context $c \in C$

$$a \cdot (\psi, c[t]) = (\psi, c[u]) \ .$$

The C-equivalence relation is denoted by $\equiv_C$.

Lemma

- *$\equiv$ and $\equiv_{C_\Sigma}$ coincide.*
- *If $\equiv$ has finite index, then there exists finite $C \subseteq C_\Sigma$ such that $\equiv$ and $\equiv_C$ coincide.*

04 Table of Contents

04 Supervised learning

Goal

Learn a small $C \subseteq C_\Sigma$ such that $\equiv$ and $\equiv_C$ coincide.

Definition (Drewes and Vogler '07)

A maximally adequate teacher answers two types of queries:

- Coefficient query: Given $t \in T_\Sigma$ the teacher supplies (ψ, t).
- Equivalence query: Given wta M the teacher supplies either
  - $\perp$ if $S(M) = \psi$; or
  - some $t \in T_\Sigma$ such that $(S(M), t) \neq (\psi, t)$.

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\square \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C \colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\Box \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C \colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

$T \cap L_C = \emptyset$ means: “No dead states”

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\Box \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C \colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

$\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$ means: "No equivalent states"

04 Main data structure

Definition

(E, T, C) is an observation table if

- E and T are finite subsets of T_Σ; C is a finite subset of C_Σ
- $E \subseteq T \subseteq \Sigma(E) = \{\sigma(t_1, \ldots, t_k) \mid \sigma \in \Sigma^{(k)}, t_1, \ldots, t_k \in E\}$
- $\square \in C$
- $T \cap L_C = \emptyset$ where $L_C = \{t \in T_\Sigma \mid \forall c \in C \colon (\psi, c[t]) = 0\}$
- $\mathrm{card}(E) = \mathrm{card}(E/_{\equiv_C})$

Definition

observation table (E, T, C) complete if $\mathrm{card}(E) = \mathrm{card}(T/_{\equiv_C})$.

04 Completing an observation table

Algorithm: COMPLETE

Require: an observation table (E, T, C)

Ensure: return a complete observation table (E', T, C) such that $E \subseteq E'$

for all $t \in T$ **do**
  if $t \not\equiv_C e$ for every $e \in E$ **then**
    $E \leftarrow E \cup \{t\}$
return (E, T, C)

04 Construction of the wta

Definition

Let $\mathcal{T} = (E, T, C)$ complete observation table. Construct $(Q, \Sigma, \mathcal{A}, \mu, F)$

- $Q = E$
- $F = \{e \in E \mid (\psi, e) \neq 0\}$
- for every $\sigma \in \Sigma_k$ and $e_1, \ldots, e_k \in E$ such that $t = \sigma(e_1, \ldots, e_k) \in T$

$$\mu_k(\sigma)_{e_1 \cdots e_k, \mathcal{T}(t)} = (\psi, t) \cdot \prod_{i=1}^{k} (\psi, e_i)^{-1}$$

- all remaining entries are 0

04 Construction of the wta

Definition

Let $\mathcal{T} = (E, T, C)$ complete observation table. Construct $(Q, \Sigma, \mathcal{A}, \mu, F)$

- $Q = E$
- $F = \{e \in E \mid (\psi, e) \neq 0\}$
- for every $\sigma \in \Sigma_k$ and $e_1, \ldots, e_k \in E$ such that $t = \sigma(e_1, \ldots, e_k) \in T$

$$\mu_k(\sigma)_{e_1 \cdots e_k, \mathcal{T}(t)} = (\psi(\mathcal{T}), t) \cdot \prod_{i=1}^{k} (\psi(\mathcal{T}), e_i)^{-1}$$

- all remaining entries are 0

04 Construction of the wta (cont'd)

Definition

Let $\mathcal{T} = (E, T, C)$ complete observation table. Define $\psi(\mathcal{T})\colon T_\Sigma \to A \setminus \{0\}$ by

- if $(\psi, t) \neq 0$

$$(\psi(\mathcal{T}), t) = (\psi, t)$$

04 Construction of the wta (cont'd)

Definition

Let $\mathcal{T} = (E, T, C)$ complete observation table. Define $\psi(\mathcal{T})\colon T_\Sigma \to A \setminus \{0\}$ by

- if $(\psi, t) \neq 0$
$$(\psi(\mathcal{T}), t) = (\psi, t)$$
- if $(\psi, t) = 0$ and $t \in T$, then let $c \in C$ be such that $(\psi, c[t]) \neq 0$
$$(\psi(\mathcal{T}), t) = (\psi, c[t]) \cdot (\psi, c[\mathcal{T}(t)])^{-1}$$

04 Construction of the wta (cont'd)

Definition

Let $\mathcal{T} = (E, T, C)$ complete observation table. Define $\psi(\mathcal{T}) \colon T_\Sigma \to A \setminus \{0\}$ by

- if $(\psi, t) \neq 0$
$$(\psi(\mathcal{T}), t) = (\psi, t)$$
- if $(\psi, t) = 0$ and $t \in T$, then let $c \in C$ be such that $(\psi, c[t]) \neq 0$
$$(\psi(\mathcal{T}), t) = (\psi, c[t]) \cdot (\psi, c[\mathcal{T}(t)])^{-1}$$
- All remaining entries are 1

04 The outer structure

Algorithm: MAIN

$\mathcal{T} \leftarrow (\emptyset, \emptyset, \{\Box\})$ {initial observation table}
loop
  $M \leftarrow \mathcal{M}(\mathcal{T})$ {construct new wta}
  $t \leftarrow$ EQUAL?(M) {ask equivalence query}
  if $t = \bot$ **then**
    return M {return the approved wta}
  else
    $\mathcal{T} \leftarrow$ EXTEND$(\mathcal{T}, t)$ {extend the observation table}

04 The workhorse: Extend

Algorithm: Extend

Require: a complete observation table $\mathcal{T} = (E, T, C)$ and a counterexample $t \in T_\Sigma$

Ensure: return a complete observation table $\mathcal{T}' = (E', T', C')$ such that $E \subseteq E'$ and $T \subseteq T'$ and one inclusion is strict

Decompose t into $t = c[u]$ where $c \in C_\Sigma$ and $u \in \Sigma(E) \setminus E$

if $u \in T$ and $u \equiv_{C \cup \{c\}} \mathcal{T}(u)$ **then**
  return Extend$(\mathcal{T}, c[\mathcal{T}(u)])$ {normalize and continue}
else
  return Complete$(E, T \cup \{u\}, C \cup \{c\})$ {add u and c to table}

TECHNISCHE UNIVERSITÄT DRESDEN

05 Table of Contents

05 Input wta

$Q = \{\mathsf{S}, \mathsf{VP}, \mathsf{NP}, \mathsf{NN}, \mathsf{ADJ}, \mathsf{VB}\}$ and $F = \{\mathsf{S}\}$

$\text{Alice} \xrightarrow{0.5} \mathsf{NN}$ $\text{Bob} \xrightarrow{0.5} \mathsf{NN}$ $\text{loves} \xrightarrow{0.5} \mathsf{VB}$ $\text{hates} \xrightarrow{0.5} \mathsf{VB}$

$\text{ugly} \xrightarrow{0.25} \mathsf{ADJ}$ $\text{nice} \xrightarrow{0.25} \mathsf{ADJ}$ $\text{mean} \xrightarrow{0.25} \mathsf{ADJ}$ $\text{tall} \xrightarrow{0.25} \mathsf{ADJ}$

$\sigma(\mathsf{NN}, \mathsf{VP}) \xrightarrow{0.5} \mathsf{S}$ $\sigma(\mathsf{NP}, \mathsf{VP}) \xrightarrow{0.5} \mathsf{S}$ $\sigma(\mathsf{VB}, \mathsf{NN}) \xrightarrow{0.5} \mathsf{VP}$

$\sigma(\mathsf{VB}, \mathsf{NP}) \xrightarrow{0.5} \mathsf{VP}$ $\sigma(\mathsf{ADJ}, \mathsf{NN}) \xrightarrow{0.5} \mathsf{NP}$ $\sigma(\mathsf{ADJ}, \mathsf{NP}) \xrightarrow{0.5} \mathsf{NP}$

05 Learned wta

$Q = \{\mathsf{NP}, \mathsf{VB}, \mathsf{VP}, \mathsf{S}, \mathsf{ADJ}\}$ and $F = \{\mathsf{S}\}$

$\mathsf{Alice} \xrightarrow{1} \mathsf{NP} \qquad \mathsf{Bob} \xrightarrow{1} \mathsf{NP} \qquad \mathsf{loves} \xrightarrow{1} \mathsf{VB} \qquad \mathsf{hates} \xrightarrow{1} \mathsf{VB}$

$\mathsf{ugly} \xrightarrow{1} \mathsf{ADJ} \qquad \mathsf{nice} \xrightarrow{1} \mathsf{ADJ} \qquad \mathsf{mean} \xrightarrow{1} \mathsf{ADJ} \qquad \mathsf{tall} \xrightarrow{1} \mathsf{ADJ}$

$\sigma(\mathsf{NP}, \mathsf{VP}) \xrightarrow{0.03125} \mathsf{S} \qquad \sigma(\mathsf{VB}, \mathsf{NP}) \xrightarrow{1} \mathsf{VP} \qquad \sigma(\mathsf{ADJ}, \mathsf{NP}) \xrightarrow{0.125} \mathsf{NP}$

TECHNISCHE UNIVERSITÄT DRESDEN

05 Thank you for your attention!

References

- Borchardt, Vogler: Determinization of finite state weighted tree automata. J. Autom. Lang. Combin. 8(3):417–463, 2003
- Borchardt: The Myhill-Nerode theorem for recognizable tree series. Proc. 7th Int. Conf. Developments in Language Theory, LNCS 2710, p. 146–158, Springer 2003
- Drewes, Vogler: Learning deterministically recognizable tree series. J. Autom. Lang. Combin., to appear, 2007