A Note on Sequential Rule-Based POS Tagging

S. Schmitz

LSV, ENS Cachan & CNRS

FSMNLP 2011, Blois, July 15, 2011

Outline

Brill Tagger

a rule-based POS tagger

this talk

a simple constructive proof of sequentiality based on (Simon, 1994)

contents

Brill Tagger

(Brill, 1992)

1. *lexical tagger*: most probable tag for each word
2. *unknown word tagger*
3. *contextual tagger*: correct tag assignments

Contextual Tagger

Example

*Chapman/NNP killed/VBN John/NNP Lennon/NNP

*John/NNP Lennon/NNP was/VBD shot/VBD by/IN Chapman/NNP

He/PRP witnessed/VBD Lennon/NNP killed/VBN by/IN Chapman/NNP

Contextual Tagger

Example

*Chapman/NNP killed/VBN John/NNP Lennon/NNP

*John/NNP Lennon/NNP was/VBD shot/VBD by/IN Chapman/NNP

He/PRP witnessed/VBD Lennon/NNP killed/VBN by/IN Chapman/NNP

$$\text{NNP VBN} \rightarrow \text{NNP VBD}$$

Contextual Tagger

Example

Chapman/NNP killed/VBD John/NNP Lennon/NNP

*John/NNP Lennon/NNP was/VBD shot/VBD by/IN Chapman/NNP

*He/PRP witnessed/VBD Lennon/NNP killed/VBD by/IN Chapman/NNP

$$\text{NNP VBN} \rightarrow \text{NNP VBD}$$

Contextual Tagger

Example

Chapman/NNP killed/VBD John/NNP Lennon/NNP

*John/NNP Lennon/NNP was/VBD shot/VBD by/IN Chapman/NNP

*He/PRP witnessed/VBD Lennon/NNP killed/VBD by/IN Chapman/NNP

$$\text{NNP VBN} \rightarrow \text{NNP VBD}$$

$$\text{VBD IN} \rightarrow \text{VBN IN}$$

Contextual Tagger

Example

Chapman/NNP killed/VBD John/NNP Lennon/NNP

John/NNP Lennon/NNP was/VBD shot/VBN by/IN Chapman/NNP

He/PRP witnessed/VBD Lennon/NNP killed/VBN by/IN Chapman/NNP

$$\text{NNP VBN} \rightarrow \text{NNP VBD}$$

$$\text{VBD IN} \rightarrow \text{VBN IN}$$

Contextual Tagger

Formally

- sequence of rewrite rules $\mathcal{C} = r_1 r_2 \cdots r_n$
- rules $r_i = u_i \to v_i$ in $\Sigma^* \times \Sigma^*$
- leftmost applications
- $[\![r_i]\!]$ is a total function $\Sigma^* \to \Sigma^*$ (identity over $\Sigma^* \setminus \Sigma^* u_i \Sigma^*$)
- $[\![\mathcal{C}]\!] = [\![r_1]\!] \,;\, [\![r_2]\!] \,;\, \cdots \,;\, [\![r_n]\!]$

Contextual Tagger

Formally

- sequence of rewrite rules $\mathcal{C} = r_1 r_2 \cdots r_n$
- rules $r_i = u_i \to v_i$ in $\Sigma^* \times \Sigma^*$
- leftmost applications
- $[\![r_i]\!]$ is a total function $\Sigma^* \to \Sigma^*$ (identity over $\Sigma^* \backslash \Sigma^* u_i \Sigma^*$)
- $[\![\mathcal{C}]\!] = [\![r_1]\!] \fatsemi [\![r_2]\!] \fatsemi \cdots \fatsemi [\![r_n]\!]$

Contextual Tagger

Algorithmically

- naive algorithm: try to match each rule at each position of input w:

$$O\left(|w| \cdot \sum_i |u_i|\right)$$

- Roche and Schabes (1995): construct a sequential transducer $\mathcal{T}$ and run over w:

$$O\left(|w| + |\mathcal{T}|\right)$$

Contextual Tagger

Algorithmically

- but in (Roche and Schabes, 1995)

$$|\mathcal{T}| \sim O(2^{\prod_{i=1}^{n} 2^{|u_i|}})$$

- here:

$$|\mathcal{T}| = O(\prod_{i=1}^{n} |u_i|)$$

Sequential Transducers

(Schützenberger, 1977)

- $\mathcal{T} = \langle Q, \Sigma, \Delta, q_0, \delta, \eta, \iota, \rho \rangle$:
  - $\delta : Q \times \Sigma \to Q$ partial transition function
  - $\eta : Q \times \Sigma \to \Delta^*$ partial transition output function with $\mathrm{dom}(\delta) = \mathrm{dom}(\eta)$
  - $\iota \in \Delta^*$ initial output string
  - $\rho : Q \to \Delta^*$ partial final output function
- defines a *sequential function* $[\![\mathcal{T}]\!] : \Sigma^* \to \Delta^*$
- existence of a canonical minimal sequential transducer

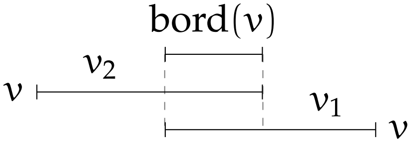
Sequential Transducers

(Schützenberger, 1977)

- $\mathcal{T} = \langle Q, \Sigma, \Delta, q_0, \delta, \eta, \iota, \rho \rangle$:
  - $\delta : Q \times \Sigma \to Q$ partial transition function
  - $\eta : Q \times \Sigma \to \Delta^*$ partial transition output function with $\mathrm{dom}(\delta) = \mathrm{dom}(\eta)$
  - $\iota \in \Delta^*$ initial output string
  - $\rho : Q \to \Delta^*$ partial final output function
- defines a *sequential function* $[\![\mathcal{T}]\!] : \Sigma^* \to \Delta^*$
- existence of a canonical minimal sequential transducer

Borders

- u is a *border* of a v iff $\exists v_1, v_2, v = uv_1 = v_2u$
- if $v \neq \varepsilon$, $\mathrm{bord}(v)$ is the longest border of v different from v



Example

$$\mathrm{bord}(b) = \varepsilon \qquad \mathrm{bord}(aa) = a$$
$$\mathrm{bord}(abb) = \varepsilon \qquad \mathrm{bord}(abaa) = a$$
$$\mathrm{bord}(ababa) = aba$$

Sequential Transducer of $\mathtt{ababb} \rightarrow \mathtt{abbbb}$

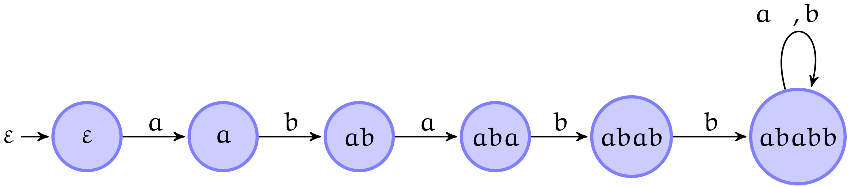
$$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$$

$$\varepsilon \rightarrow \varepsilon \xrightarrow{a} a \xrightarrow{b} ab \xrightarrow{a} aba \xrightarrow{b} abab \xrightarrow{b} ababb$$

$$\delta(w, a) = \begin{cases} wa & \text{if } wa \leqslant_{\text{pref}} u \\ w & \text{if } w = u \\ \mathrm{bord}(wa) & \text{otherwise} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

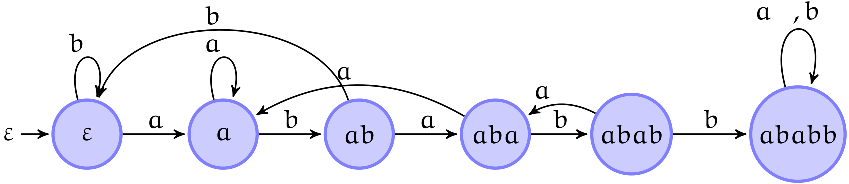
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\delta(w, a) = \begin{cases} wa & \text{if } wa \leqslant_{\text{pref}} u \\ w & \text{if } w = u \\ \mathrm{bord}(wa) & \text{otherwise} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \rightarrow \mathtt{abbbb}$

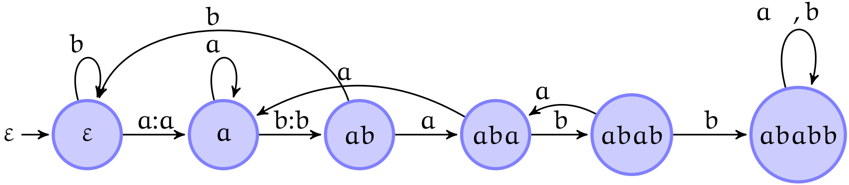
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\delta(w, a) = \begin{cases} wa & \text{if } wa \leqslant_{\mathrm{pref}} u \\ w & \text{if } w = u \\ \mathrm{bord}(wa) & \text{otherwise} \end{cases}$$

Sequential Transducer of ababb → abbbb

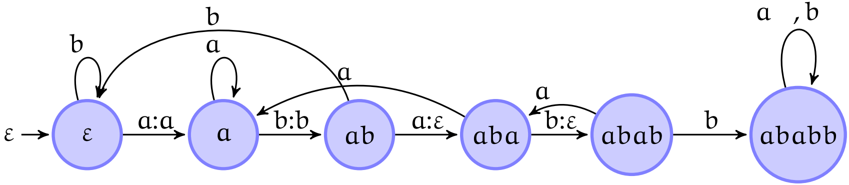
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\eta(w, a) = \begin{cases} a & \text{if } wa \leqslant_{\mathrm{pref}} (u \wedge v) \\ \varepsilon & \text{if } (u \wedge v) <_{\mathrm{pref}} wa <_{\mathrm{pref}} u \\ (u \wedge v)^{-1} \cdot v & \text{if } wa = u \\ a & \text{if } w = u \\ \rho(w)a \cdot \rho(\mathrm{bord}(wa))^{-1} & \text{otherwise.} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

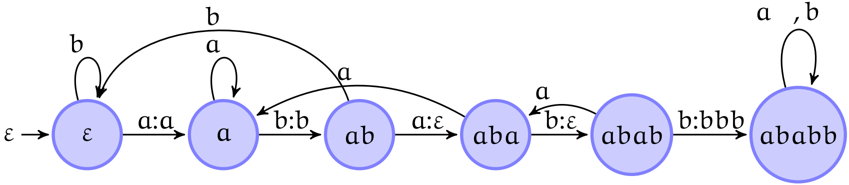
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\eta(w, a) = \begin{cases} a & \text{if } wa \leqslant_{\mathrm{pref}} (u \wedge v) \\ \varepsilon & \text{if } (u \wedge v) <_{\mathrm{pref}} wa <_{\mathrm{pref}} u \\ (u \wedge v)^{-1} \cdot v & \text{if } wa = u \\ a & \text{if } w = u \\ \rho(w)a \cdot \rho(\mathrm{bord}(wa))^{-1} & \text{otherwise.} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

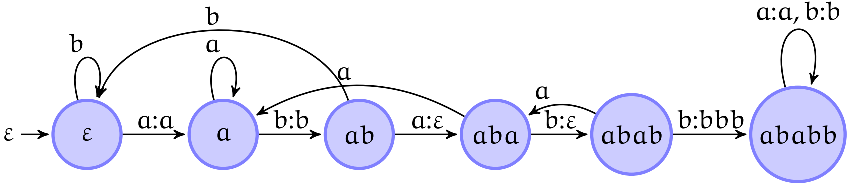
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\eta(w, a) = \begin{cases} a & \text{if } wa \leqslant_{\text{pref}} (u \wedge v) \\ \varepsilon & \text{if } (u \wedge v) <_{\text{pref}} wa <_{\text{pref}} u \\ (u \wedge v)^{-1} \cdot v & \text{if } wa = u \\ a & \text{if } w = u \\ \rho(w)a \cdot \rho(\mathrm{bord}(wa))^{-1} & \text{otherwise.} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

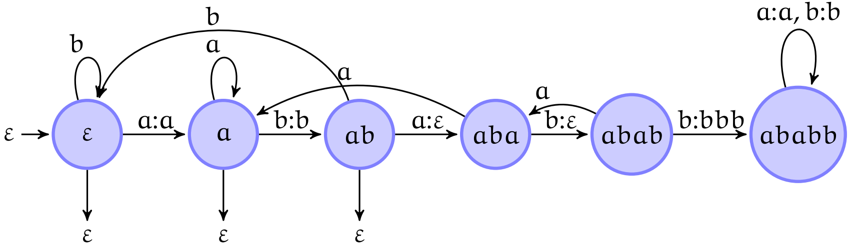
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\eta(w, a) = \begin{cases} a & \text{if } wa \leqslant_{\text{pref}} (u \wedge v) \\ \varepsilon & \text{if } (u \wedge v) <_{\text{pref}} wa <_{\text{pref}} u \\ (u \wedge v)^{-1} \cdot v & \text{if } wa = u \\ a & \text{if } w = u \\ \rho(w)a \cdot \rho(\mathrm{bord}(wa))^{-1} & \text{otherwise.} \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

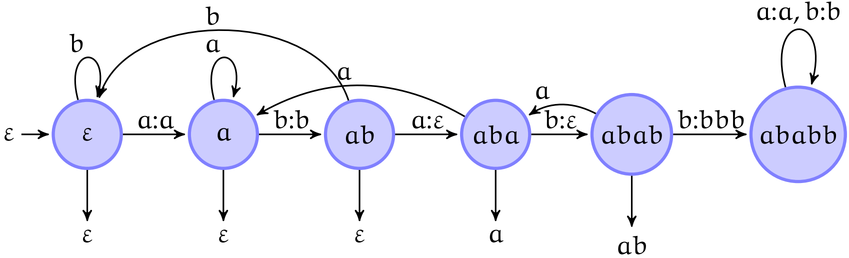
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\rho(w) = \begin{cases} \varepsilon & \text{if } w \leqslant_{\text{pref}} (u \wedge v) \\ (u \wedge v)^{-1} \cdot w & \text{if } (u \wedge v) <_{\text{pref}} w <_{\text{pref}} u \\ \varepsilon & \text{otherwise, i.e. if } w = u \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \rightarrow \mathtt{abbbb}$

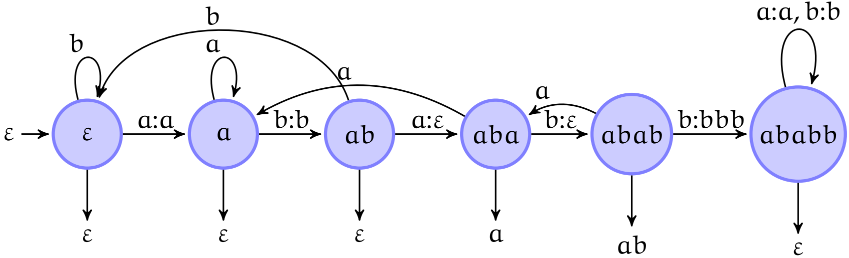
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\rho(w) = \begin{cases} \varepsilon & \text{if } w \leqslant_{\mathrm{pref}} (u \wedge v) \\ (u \wedge v)^{-1} \cdot w & \text{if } (u \wedge v) <_{\mathrm{pref}} w <_{\mathrm{pref}} u \\ \varepsilon & \text{otherwise, i.e. if } w = u \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

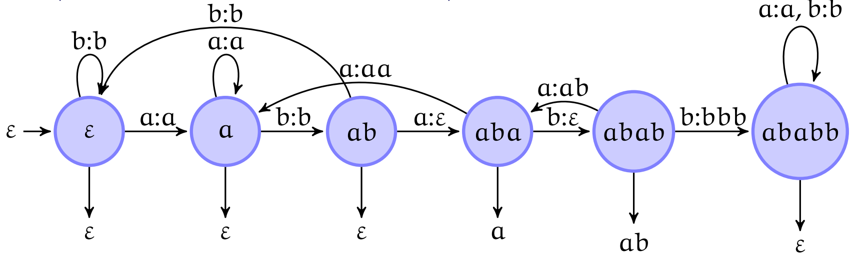
$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\rho(w) = \begin{cases} \varepsilon & \text{if } w \leqslant_{\mathrm{pref}} (u \wedge v) \\ (u \wedge v)^{-1} \cdot w & \text{if } (u \wedge v) <_{\mathrm{pref}} w <_{\mathrm{pref}} u \\ \varepsilon & \text{otherwise, i.e. if } w = u \end{cases}$$

Sequential Transducer of $\mathtt{ababb} \to \mathtt{abbbb}$

$\mathcal{T}_r = \langle \mathrm{Pref}(u), \Sigma, \Sigma, \varepsilon, \delta, \eta, \varepsilon, \rho \rangle$



$$\eta(w,a) = \begin{cases} a & \text{if } wa \leqslant_{\mathrm{pref}} (u \wedge v) \\ \varepsilon & \text{if } (u \wedge v) <_{\mathrm{pref}} wa <_{\mathrm{pref}} u \\ (u \wedge v)^{-1} \cdot v & \text{if } wa = u \\ a & \text{if } w = u \\ \rho(w)a \cdot \rho(\mathrm{bord}(wa))^{-1} & \text{otherwise.} \end{cases}$$

Result

See paper

- correctness: $[\![\mathcal{T}_r]\!] = [\![r]\!]$
- $\mathcal{T}_r$ is normalized
- $\mathcal{T}_r$ is minimal
- $|\mathcal{T}_r| = |u| + 1$ (state complexity)

Remaining Issues: Composition

Is there a smart way of composing the $\mathcal{T}_{r_i}$ to obtain $\mathcal{T}_{\mathcal{C}}$?

PTime-completeness of word problem for a sequence of length-preserving rewrite rules might indicate: NO

Remaining Issues: Semantics of Rules

- in the paper: single leftmost application
- in (Roche and Schabes, 1995): iterated left-to-right leftmost application according to construction
- in (Brill, 1992): iterated left-to-right leftmost application with possible overlap according to source code...

Remaining Issues: Semantics of Rules

- in the paper: single leftmost application
- in (Roche and Schabes, 1995): iterated left-to-right leftmost application according to construction
- in (Brill, 1992): iterated left-to-right leftmost application with possible overlap according to source code...

Remaining Issues: Semantics of Rules

- in the paper: single leftmost application
- in (Roche and Schabes, 1995): iterated left-to-right leftmost application according to construction
- in (Brill, 1992): iterated left-to-right leftmost application with possible overlap according to source code...

Remaining Issues: Semantics of Rules

- in the paper: single leftmost application
- in (Roche and Schabes, 1995): iterated left-to-right leftmost application according to construction
- in (Brill, 1992): iterated left-to-right leftmost application with possible overlap according to source code...

Remaining Issues: Semantics of Rules

- in the paper: single leftmost application
- in (Roche and Schabes, 1995): iterated left-to-right leftmost application according to construction
- in (Brill, 1992): iterated left-to-right leftmost application with possible overlap according to source code...

References

Brill, E., 1992. A simple rule-based part of speech tagger. In *ANLP '92*, pages 152–155. ACL Press. doi:10.3115/974499.974526.

Roche, E. and Schabes, Y., 1995. Deterministic part-of-speech tagging with finite-state transducers. *Comput. Linguist.*, 21(2):227–253. http://www.aclweb.org/anthology/J95-2004.pdf.

Schützenberger, M.P., 1977. Sur une variante des fonctions séquentielles. *Theor. Comput. Sci.*, 4(1):47–57. doi:0.1016/0304-3975(77)90055-X.

Simon, I., 1994. String matching algorithms and automata. In Karhumäki, J., Maurer, H., and Rozenberg, G., editors, *Results and Trends in Theoretical Computer Science: Colloquium in Honor of Arto Salomaa*, volume 812 of *LNCS*, pages 386–395. Springer. ISBN 978-3-540-58131-4. doi:10.1007/3-540-58131-6_61.