Finite State Transducers for Information Extraction

TID 2009/10

Marek Schmidt

FIT BUT

2009-12-31

Introduction

We want to show that a practical Information Extraction system can be implemented as a Finite State Transducer. For that we need to...

1. Define a string representation of text with annotations.
2. Define a set of operators on FSTs.
3. Implement Information Extraction system using this set of operators.

Finite State Transducer

$$T = (Q, \Sigma, \Gamma, I, F, \delta) \tag{1}$$

$$Q \quad \ldots \quad \text{Finite set of states.} \tag{2}$$

$$\Sigma \quad \ldots \quad \text{Finite set, input alphabet.} \tag{3}$$

$$\Gamma \quad \ldots \quad \text{Finite set, output alphabet.} \tag{4}$$

$$I \subseteq Q \quad \ldots \quad \text{Initial states} \tag{5}$$

$$F \subseteq Q \quad \ldots \quad \text{Final states} \tag{6}$$

$$\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \times Q \quad \ldots \quad \text{Transitions} \tag{7}$$

$$x[T]y \subseteq \Sigma^* \times \Gamma^* \ldots \text{Behavior} \tag{8}$$

Operations on Finite State Transducers

Atom	x/y	$x[x/y]y$	(9)
Identity	R	$x[R]x$ iff $x \in R$	(10)
Union	$T \cup U$	$x[T \cup U]y$ iff $x[T]y$ or $x[U]y$	(11)
Intersection	$T \cap U$	$x[T \cap U]y$ iff $x[T]y$ and $x[U]y$	(12)
Concatenation	$T.U$	$xw[T.U]yz$ iff $x[T]y$ and $w[U]z$	(13)
Iteration	T^*	Minimal such as $\epsilon[T^*]\epsilon$ and $x[T^*]y \wedge w[T]z \Rightarrow xw[T^*]yz$.	(14)
Composition	$T \circ U$	$x[T \circ U]y$ iff $\exists z.x[T]z$ and $z[U]y$	(15)

Composition (ϵ-free)

$$T = (Q_T, \Sigma_T, \Gamma_T, I_T, F_T, \delta_T) \quad (16)$$

$$U = (Q_U, \Sigma_U, \Gamma_U, I_U, F_U, \delta_U) \quad (17)$$

$$T \circ U = (Q_V, \Sigma_V, \Gamma_V, I_V, F_V, \delta_V) \quad (18)$$

- $Q_V = Q_T \times Q_U$
- $\Sigma_V = \Sigma_T$
- $\Gamma_V = \Gamma_U$
- $I_V = \{(t,u) \in Q_V | t \in I_T \wedge u \in I_U\}$
- $F_V = \{(t,u) \in Q_V | t \in F_T \wedge u \in F_U\}$

$$\delta_V = \{((q_T, q_U), s_T, t_V, (r_T, r_U)) \in Q_V \times \Sigma_V \times \Gamma_V \times Q_V | \exists x.(q_T, s_T, x, r_T) \in \delta_T \wedge (q_U, x, t_V, r_U) \in \delta_V)\} \quad (19)$$

Information Extraction

- Extract structured information from unstructured natural language text.
- *Entity Recognition*
- *Relationship Extraction*

by the means of *patterns* on *annotated text*.

Annotation Model

	Oracle	has	acquired	Sun	Microsystems	for	7.4	billion.
POS	N	V	V	N	N	I	C	C
PHRASE	N	V		N		I	C	
ROLE	Recipient			Theme			Money	
EVENT	Acquisition							

Annotation Model

L . . . Finite set of *annotation labels*

1. *Coloured strings*
   - Extend alphabet $\Sigma' = \Sigma \times 2^L$
   - FST with Predicates and Identitites (Noord01)
2. *Markup*
   - Extend alphabet $\Sigma' = \Sigma \cup (L \times \{_{begin}, _{end}\})$

The *Markup* route seems more practical. . .

Operations on labels

$$\text{Match} \quad \langle x \rangle \quad \equiv x_{begin}.(\Sigma - \{x_{begin}, x_{end}\})^*.x_{end} \tag{20}$$

$$\text{Label} \quad T : x \quad \equiv \epsilon / x_{begin}.T.\epsilon / x_{end} \tag{21}$$

$$\text{Context} \quad U = T_M \quad \delta_U = \delta_T \cup \{(q, l, l, q) | \tag{22}$$

$$q \in Q_U, l \in (L - M) \times \{begin, end\}\}$$

- (see Ignore operator in (KaplanAndKay94))

Example

$$(\langle D \rangle.(\langle A \rangle)^*.(\langle N \rangle)^* \langle N \rangle) : NP \tag{23}$$

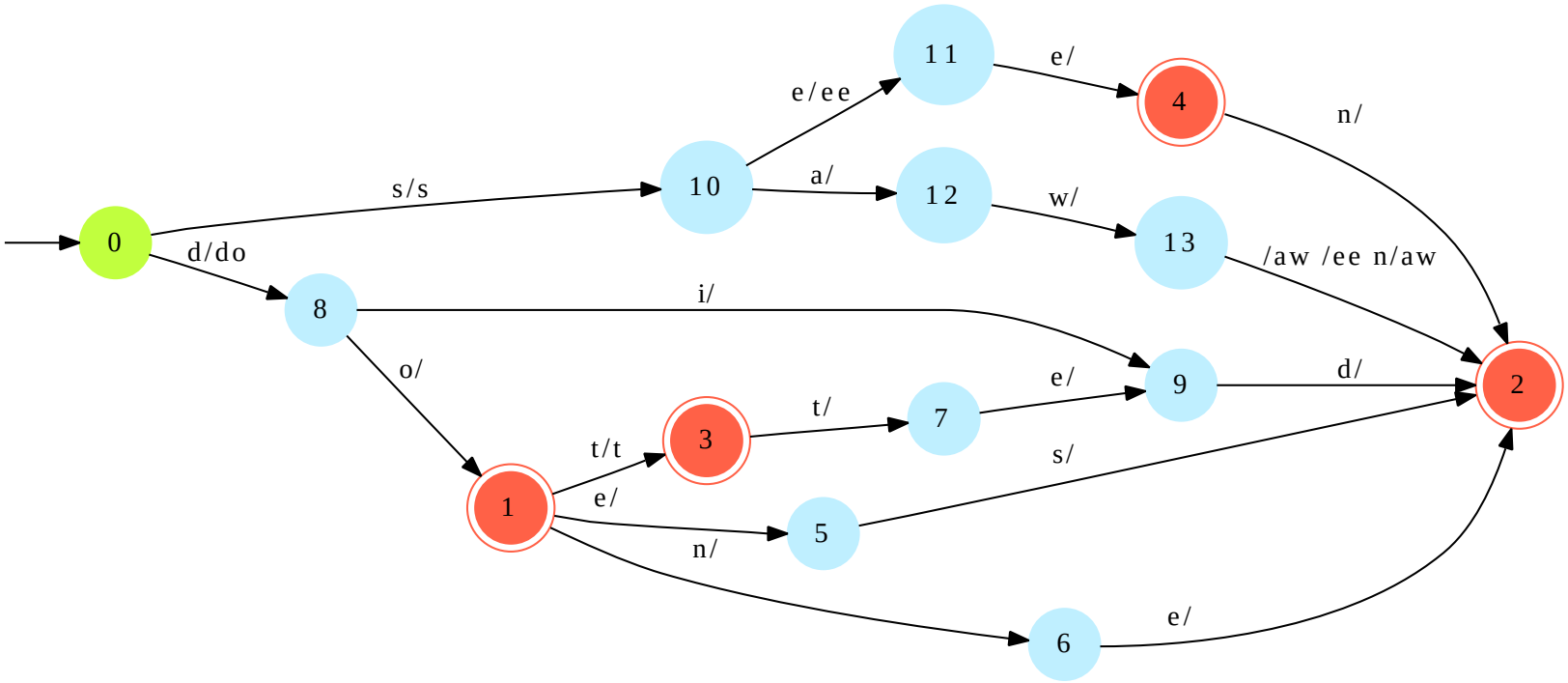
$$((\langle NP \rangle) : Recipient.(\text{has acquired})_{\emptyset}.(\langle NP \rangle) : Theme) : Acquisition \tag{24}$$

Information Extraction Pipeline

Composition of transducers.

1. Sentence splitting
   - Dummy label to ends of sentences to act as a sentinel.
2. Tokenization
   - w_{begin} lorem $w_{end} w_{begin}$ ipsum w_{end}
3. Morphology analysis
   - Lemmatization
   - Stemming
4. Part-of-speech tagging
5. Gazetteers
6. Shallow Parsing
7. Extraction Patterns

Morphology example



$$do/do \cup does/do \cup done/do \cup did/do \cup$$
$$see/see \cup saw/see \cup seen/see \cup$$
$$saw/saw \cup sawn/saw \cup$$
$$dot/dot \cup dotted/dot$$

Efficiency discussion

- Unlike FSM, no equivalent DFST for every NFST.
  - But we need nondeterminism in NLP anyway for ambiguities.
- *Subsequential transducers*
  - FST with deterministic input.
- *Finitely subsequential transducers*
  - deterministic transducers with finite set of final output strings.
- Finding equivalent finitely subsequential transducer is decidable (Mohri2003)
- *Label* operation in general leads to non-subsequentiable transducers.
- on-line implementation of composition operation.

Conclusion

- defined a set of operators on annotated text.
- outlined an implementation of all necessary components of an information extraction system.
- implemented a proof-of-concept on top FSA Utilities toolbox http://www.let.rug.nl/~vannoord/Fsa/ (Noord99)
- still more theoretical than practical.

References

- (KaplanAndKay94) R. Kaplan and M. Kay. 1994. Regular Models of Phonological Rule System, *Computational Linguistics*
- (Mohri2003) C. Allauzen and M. Mohri. 2003. Finitely Subsequential Transducers, *International Journal of Foundations of Computer Science*
- (Noord99) G. van Noord and D. Gerdemann. 1999. An Extendible Regular Expression Compiler for Finite-state Approaches in Natural Language Processing. *WIA 99*
- (Noord01) G. van Noord and D. Gerdemann. 2001. Finite State Transducers with Predicates and Identities. *Grammars*