Using Adaptive Formalisms to Describe Context-Dependencies in Natural Language

João José Neto and Miryam de Moraes

Lab. de Ling. e Tecnologias Adaptativas, Esc. Politécnica da Univ. de S. Paulo
Av. Prof. Luciano Gualberto tr. 3 n. 158, Cid. Universitária
05508-900 S.Paulo, Brazil
{joao.jose,miryam.moraes}@poli.usp.br
http://www.pcs.usp.br/~lta

Abstract. This text sketches a method based on adaptive technology for representing context-dependencies in NL processing. Based on a previous work [4] dedicated to syntactical ambiguities and non-determinisms in NL handling we extend it to consider context-dependencies not previously addressed. Although based on the powerful adaptive formalism [3], our method relies on adaptive structured pushdown automata [1] and grammars [2] – resulting simplicity, low-cost and efficiency.

1 Introduction

Since low-complexity language formalisms are too weak to handle NL, stronger formalisms are required, most of them resource demanding, hard to use or unpractical. Structured pushdown automata are excellent to represent regular and context-free aspects of NLs by allowing them to be split into a regular layer (implemented as finite-state machines) and a context-free one (represented by a pushdown store). Such device accepts deterministic context-free languages in linear time, and is suitable as an underlying mechanism for adaptive automata, allowing handling – without loss of simplicity and efficiency – languages more complex than context-free ones.

Classical grammars may describe non-trivial interdependencies between and inside sentences: attribute-, two-level-, evolving- and adaptive- grammars. Here, context dependency is handled with adaptive grammars (which may be converted [2] into structured pushdown adaptive automata [3]) by executing adaptive actions attached to the rule being used (stating self-modifications – rule addition and deletion – to be imposed). Upon a context-dependency is detected, one of such rules is applied, and the attached adaptive action learns to handle of the context dependency by adequately changing the underlying grammar. Starting from an initial grammar, the adaptive device follows its rules until some new context-dependency is handled. Thereafter, its operation follows the modified underlying grammar until either the sentence is fully derived or no matching rule is found. Complex languages, e.g. NLs, may be handled in this way, since adaptive grammars have type-0 power [1], [2]. By converting them into adaptive

N.J. Mamede et al. (Eds.): PROPOR 2003, LNAI 2721, pp. 94–97, 2003.

structured pushdown automata, simplicity and efficiency are achieved through piecewise-regular handling of the language, validating adaptive devices as practical and efficient for NL handling [5].

2 Illustrating Example

This example illustrates nominal agreement in Portuguese using an adaptive grammar [2] and considers: attractive agreement for adjectives placed before nouns coordinated with preposition “*e*” or comma (e.g. *As antigas mansões e parques*) and grammatical agreement for adjectives placed after such nouns (e.g. *Os parques e mansões restaurados*). Our adaptive grammar is defined as a 3-tuple (G^0, T, R^0) where:

T = finite set of adaptive functions
$G^0 = (V_N^0, V_T^0, V_C, P_L^0, P_D^0, S)$ initial grammar
V_N^0 = non-empty finite set of non-terminals
V_T = non-empty finite set of terminals
$V_N^0 \cap V_T = \emptyset$
V_C = finite set of context symbols
$V^0 = V_N^0 \cup V_T \cup V_C$
V_N^0, V_T, V_C are disjoint sets
$S \in V_N^0$ start symbol of the grammar
P_L^0 rules used in CF situations
P_D^0 rules used in CD situations
$R^0 = P_L^0 \cup P_D^0$
The example refers to $G = (G^0, T, R^0)$:
$G^0 = (V_N^0, V_T^0, V_C, P_L^0, P_D^0, S)$
$V_N^0 = \{C, C_1, C_2, C_3, C_4, C_5, C_6, C_7, D, A, S, C_{8a}, C_{8l}, ESM, ESF, EPM, EPF\}$
$V_C = \{sm, sf, pm, pf\}$
$V_T = \{as,\ e,\ antigas,\ mans\tilde{o}es,\ parques, restaurados, pra\c{c}as, \text{“,”}\}$

Context symbols sm, sf, pm, pf denote attributes: singular/plural masc./fem. D, A, S denote determinants, adjectives and nouns, respectively. Starting symbol is C. Adaptive functions dynamically handle optional elements: further nouns, a determinant, an adjective placed before/after the noun. A context-dependency is handled by an adaptive function when the noun is processed. It checks its agreement with the previous determinant and adjective. Another adaptive function enforces agreement between the adjective and multiple nouns.P_L^0 and $P_D^0$are:

$C \to \{A_1(C, C_1)\}DC_1$
$sfC_1 \to \{A_{1c}(C_1, C, ES, EF)\}ESF$
$pfC_1 \to \{A_{1c}(C_1, C, EP, EF)\}EPF$
$smC_1 \to \{A_{1c}(C_1, C, ES, EM)\}ESM$
$pmC_1 \to \{A_{1c}(C_1, C, EP, EM)\}EPM$
$C \to \{A_2(C, C_2, C_1)\}AC_2$
$sfC_2 \to \{A_{1c}(C_2, C, ES, EF)\}ESF$
$pfC_2 \to \{A_{1c}(C_2, C, EP, EF)\}ESF$
$smC_2 \to \{A_{1c}(C_2, C, ES, EM)\}ESM$
$pmC_2 \to \{A_{1c}(C_2, C, EP, EM)\}EPM$
$C \to SC_3$
$sfC_3 \to \{A_{3c}(C_6, C_1, C_2, ES, EF)\}ESF$
$smC_3 \to \{A_{3c}(C_6, C_1, C_2, ES, EM)\}ESM$
$ESM \to C \quad ESF \to C$
$EPM \to C \quad EPF \to C$
$pfC_3 \to \{A_{3c}(C_6, C_1, C_2, EP, EF)\}EPF$
$pmC_3 \to \{A_{3c}(C_6, C_1, C_2, EP, EM)\}EPM$
$C_7 \to \emptyset$
$C_3 \to \varepsilon$
$C_3 \to C_6$
$C_3 \to \text{“}e\text{”}C_4$
$C_3 \to \text{“,”}C_4$
$C_4 \to SC_5$
$smC_5 \to \{A_4(C_6, ES, EM)\}ESM$
$sfC_5 \to \{A_4(C_6, ES, EF)\}ESF$
$pmC_5 \to \{A_4(C_6, EP, EM)\}EPM$
$pfC_5 \to \{A_4(C_6, EP, EF)\}EPF$
$C_3 \to \{A_5(C_3, C_6), C_7, C_{8a}, C_{8l})\}AC_7$
$ESM \to C_3 \quad ESF \to C_3$
$EPF \to C_3 \quad EPM \to C_3$

Adaptive Actions:

$\mathbf{A_1}$(XC, XC1) =
/*remove extra determinant*/
$\{-[XC \to \{A_1(XC, XC_1)\}DXC1]\}$

$\mathbf{A_{1C}}$(x1,x2,xn,xg) =
/*Delete transitions with improper context symbols*/
$\{-[smx1 \to \{A_{1c}(x1, x2, xn, xg)\}ESM]$
$-[sfx1 \to \{A_{1c}(x1, x2, xn, xg)\}ESF]$
$-[pmx1 \to \{A_{1c}(x1, x2, xn, xg)\}EPM]$
$-[pfx1 \to \{A_{1c}(x1, x2, xn, xg)\}EPF]$
/*ATK: dummy adaptive action. It memorizes determinant attributes*/
$+[x1 \to \{ATK(xn, xg)\}x2]\}$

$\mathbf{A_2}$(XC, XC2, XC1) =
$\{-[XC \to \{A_1(XC, XC1)\}DXC1]$
/*disable further det. or adj.*/
$-[XC \to \{A_2(XC, XC2, XC1)\}AXC2]\}$

$\mathbf{A_{3c}}$(xc6, xc1,xc2,xc,xn,xg) =
$\{dn, dg, an, ag :$
/*memorize noun attributes*/
$A_4(xc6, xn, xg)$
/*fix inflexion of determinant and adjective before noun*/
$-[xc1 \to \{ATK(dn, dg)\}xc]$
$-[xc2 \to \{ATK(an, ag)\}xc]$
$+[xc1 \to \{ATK(xn, xg)\}xc]$
$+[xc2 \to \{ATK(xn, xg)\}xc]$

$\mathbf{A_4}$(xc6,xn,xg) =$\{x, s* :$
/*ATKS: dummy adaptive action. It memorizes noun attributes*/
$-[x \to xc6]$
$+[x \to \{ATKS(xn, xg)\}s]$
$+[s \to xc6]\}$

$\mathbf{A_5}$(xc3,xc6,xc7,xc8a,xc8l) =
$\{x, xsm, xsf, xpm, xpf, xn1, xn2,$
$xn3, xn4, xg1, xg2, xf1, xf2, x1, x2,$
$xaux1*, xaux2* :$
/* imposes attractive agreement*/
$?[x \to xc6]$
$?[xsm \to \{ATKS(ES, EM)\}x]$
$?[xsf \to \{ATKS(ES, EF)\}x]$
$?[xpf \to \{ATKS(EP, EF)\}x]$
$?[xpm \to \{ATKS(EP, EM)\}x]$
$+[smxc7 \to \{AT(xsm, xc7, xaux1)\}xaux1]$
$+[sfxc7 \to \{AT(xsf, xc7, xaux1)\}xaux1]$
$+[pmxc7 \to \{AT(xpm, xc7, xaux1)\}xaux1]$
$+[pfxc7 \to \{AT(xpf, xc7, xaux1)\}xaux1]$
/*initializes logical agreement */
$?[xc3 \to \{ATKS(xn1, EF)\}xf1]$
$?[xc3 \to \{ATKS(xn2, EM)\}xm1]$
$?[xf1 \to \{ATKS(xn3, xg1)\}x1]$
$?[xm1 \to \{ATKS(xn4, xg2)\}x2]$
$+[pfxc7 \to \{Z(xf1, x1, xaux2, xc8l)\}xaux2]$
$+[pmxc7 \to \{Z(xm1, x2, xaux2, xc8l)\}xaux2]$
$CL(xf1, xaux2, xc8l, xc7)\}$

CL(xf1,xaux2,xc8l,xc7) =
/* completes logical agreement*/
$\{xn1, xn2, xm, xf :$
$?[xf1 \to \{ATKS(xn1, EM)\}xm]$
$?[xf1 \to \{ATK(xn2, EF)\}xf]$
$+[pmxc7 \to \{Z(xm, xm, xaux2, xc8l)\}xaux2]$
$EliminaPF(xm, xc7, xaux2)$
$CL(xf, xaux2, xc8l, xc7)\}$

EliminaPF(xm,xc7,xaux2) =
/* removes the pl. fem. agreement*/
$\{x, y, z :$
$-[pfxc7 \to \{Z(x, y, xaux2, z)\}xaux2]\}$

Z(x,y,z,xc8l) =
/*inserts a transition to a final state*/
$\{+[z \to xc8l]\}$

AT(x,xc7,xaux1)=$\{xsmx, xsfx, xpmx, xpfx :$
/*performs transition self removal*/
$-[smxc7 \to \{AT(x, xc7, xaux1)\}xsmx]$
$-[sfxc7 \to \{AT(x, xc7, xaux1)\}xsfx]$
$-[pmxc7 \to \{AT(x, xc7, xaux1)\}xpmx]$
$-[pfxc7 \to \{AT(x, xc7, xaux1)\}xpfx]$
/*inserts an adequate transition*/
$+[smxc7 \to xsmx]$
$+[sfxc7 \to xsfx]$
$+[pmxc7 \to xpmx]$
$+[pfxc7 \to xpfx]\}$

ATK (xn,xg) = { }
ATKS (xn,xg) = { }

This simple example illustrates NL processing through adaptive formalisms. The following is a simplified derivation of the sentence "*as antigas praças, parques e mansões restaurados*" in our grammar.

C $\Rightarrow_0$ {A_1(C,C_1)}DC_1 $\Rightarrow_1$ *as* pf C_1 $\Rightarrow_1$ {A_{1c}(C_1, C, EP, EF)} EPF $\Rightarrow_2$ *as* C $\Rightarrow_2$ *as* {A_2(C,C_2,C_1)} AC_2 $\Rightarrow_3$ *as antigas* pf C_2 $\Rightarrow_3$ *as antigas* {A_{1c} (C_2, C, EP, EF) }EPF $\Rightarrow_4$ *as antigas* C $\Rightarrow_4$ *as antigas* S C_3 $\Rightarrow_4$ *as antigas praças* pf C_3 $\Rightarrow_4$ *as antigas praças* {A_{3c}(C_6,C_1,C_2,C,EP,EF)}EPF $\Rightarrow_5$ *as antigas praças* C_3 $\Rightarrow_5$ *as antigas praças,* C_4 $\Rightarrow_5$ *as antigas praças,* S C_5 $\Rightarrow_5$ *as antigas praças, parques* pm C_5 $\Rightarrow_5$ *as antigas praças, parques* {A_4(C_6,EP,EM)}EPM $\Rightarrow_6$ *as antigas praças, parques* C_3 $\Rightarrow_6$ *as antigas praças, parques e* C_4 $\Rightarrow_6$ *as antigas praças, parques e mansões* pf C_5 $\Rightarrow_6$ *as antigas praças, parques e mansões* {A_4(C_6, EP, EF) }EPF $\Rightarrow_7$ *as antigas praças, parques e mansões* C_3 $\Rightarrow_7$ *as antigas praças, parques e mansões* {A_5(C_3,C_6, C_7,C_{8a},C_{8l})} A C_7 $\Rightarrow$8 *as antigas praças, parques e mansões restaurados* pm C_7 $\Rightarrow_8$ *as antigas praças, parques e mansões restaurados* {Z(S2,S2,xaux2,C_{8l})} xaux21 $\Rightarrow_9$ *as antigas praças, parques e mansões restaurados* C_{8l} $\Rightarrow_9$ *as antigas praças, parques e mansões restaurados.*

Remark. "$\Rightarrow_i$", i ∈ N, denotes derivation over the rules $P_L{}^i \cup P_D{}^i$. They are available after the execution of the adaptive actions.

3 Conclusion

Many forms are reported in the literature [5] for the representation of NLs and their processing by a computer. Extending the results achieved in our previous works, this paper reports a proposal for the implementation of a method for modeling, representing and handling context-dependencies in NLs by means of adaptive devices [3]. The incremental dynamic nature of our device turns it into an attractive and low-cost option with good static and dynamic time and space performance.

References

[1] João José Neto: Contribuição à metodologia de construção de compiladores. São Paulo, 1993, 272p. Thesis (Livre-Docência) Escola Politécnica, Universidade de São Paulo.[In Portuguese]

[2] Iwai, M.K. Um formalismo gramatical adaptativo para linguagens dependentes de contexto. São Paulo 2000, 191p. Doctoral Thesis. Escola Politécnica, Universidade de São Paulo.

[3] Neto, J.J.: Adaptive rule-driven devices – general formulation and case study – CIAA'2001 Sixth International Conference on Implementation and Application of Automata. Lecture Notes in Computer Science, Springer-Verlag, Pretoria (2001)

[4] Neto, J.J., Moraes, M.: Formalismo adaptativo aplicado ao reconhecimento de linguagem natural – Anais da Conferencia Iberoamericana en Sistemas, Cibernética e Informática, 19–21 de Julio, 2002, Orlando, Florida (2002)

[5] Taniwaki, C.Y.O.: Formalismos Adaptativos na Análise Sintática de Linguagem Natural – MSc Dissertation, Escola Politécnica da Universidade de São Paulo (2002)