Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 1102–1107

Procedia Computer Science

www.elsevier.com/locate/procedia

9th International Conference on Ambient Systems, Networks and Technologies, ANT-2018 and
the 8th International Conference on Sustainable Energy Information Technology,
SEIT 2018, 8-11 May, 2018, Porto, Portugal

Syntactic analysis of natural language sentences based on rewriting systems and adaptivity

Paulo Roberto Massa Cereda[a], Newton Kiyotaka Miura[a,b,*], João José Neto[a]

[a] *Escola Politécnica, Departamento de Engenharia de Computação e Sistemas Digitais, Universidade de São Paulo*
Av. Prof. Luciano Gualberto, s/n, Travessa 3, 158, CEP: 05508-900 – São Paulo, SP – Brasil
[b] *Olos Tecnologia e Sistemas Ltda., Av. Luiz Dumont Villares, 1160, 10º andar, CEP: 02046-070 – São Paulo, SP – Brasil*

Abstract

The intricate, dependent structures found in natural language pose as a challenge for computational processing. Existing approaches resort to either probabilistic models or case-oriented syntactic mappings, leading to unsatisfactory or excessively convoluted grammatical rules. As a means to reduce complexity and offer an incremental, hierarchical approach to the phenomenon of context sensitivity, this paper presents a rule-based rewriting system using adaptive technology for syntactic analysis of sentences in natural language. We provide a detailed description of a sentence with dependent constructs being decomposed into a syntactic tree through successive reductions as a proof of concept.

Keywords: rewriting systems, adaptive technology, natural language processing

1. Introduction

Researchers have been motivated to create automatic systems that artificially reproduce human abilities with the goal of substituting humans in performing activities like communicating with another human. Half a century ago, Weizenbaum[1] developed a text-based "chat bot", i.e, a computer program that simulated a conversation of a human being, implementing algorithms to analyze text sentences created by humans and synthesize responses.

Natural language modelling can be achieved through grammatical rules[2]; such initiative analyzes texts by decomposing them into smaller elements. Texts are formed by sentences, and each sentence is composed by words organized according to syntactic rules. Each word has also its own formation rules. The meaning attributed to words and the syntactic construction that organizes their occurrence in a sentence help us understand the text as a whole, coherent structure. Natural language understanding requires the analysis of lexical, syntactic and semantic information available in a text.

[*] Corresponding author, +55 11 3091-5402.
E-mail addresses: paulo.cereda@usp.br (Paulo Roberto Massa Cereda)., nkmiura@usp.br (Newton Kiyotaka Miura)., jjneto@usp.br (João José Neto).

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.164

In this paper we apply a rule-based rewriting system to syntacticly analyze texts sentences in natural language. Rewriting systems offer conveniences for handling different levels of abstraction, such as the ones encountered in natural language processing. Lexical, syntactic and semantic structures can be handled one at a time, based on which rewriting level is currently in operation.

The remainder of this paper is presented as follows. Section 2 introduces the background concepts, namely natural language processing, rewriting systems and adaptivity, as a means to support the proposed system, formally introduced later in Section 3. A sample sentence written in Portuguese is analyzed in Section 4. Finally, Section 5 presents the final remarks and identifies future directions.

2. Background

In order to support our approach of incremental syntactic analysis of context-sensitive sentences, presented later on Section 3, this section briefly introduces the basic concepts of natural language processing, rewriting systems and adaptivity.

2.1. Natural language processing

Rule-based description of natural language is an approach to analyze rule-governed linguistic behavior in computational natural language processing (also known as NLP)[3]. The hierarchy of language classes proposed by Chomsky[4] with classes of equivalent grammars and recognizers can be used in both formal and natural language modeling. In this paper we focus solely on symbolic processing that considers the rules of language formation.

When considering the rules of language formation, a sentence is represented as a sequence of symbols (or characters) that belongs to an finite alphabet. Finite sequences of symbols of a sentence can be grouped into words, such that a sentence to be also described as a finite sequence of words.

Each word has a set of grammatical properties besides its textual representation. Such set includes the part of speech (POS) category it belongs to, such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection and article[3]. Observe that a single word can belong to several POS categories at once. Depending on the POS category, a word can have further associated properties; for instance, in Portuguese, a noun can have inflections indicating its gender (masculine or feminine) and number (plural or singular)[5,6].

By grouping a sequence of words with a hierarchical structure, the linguistic unit called *syntagm* can be constructed. Syntactic analysis of a sentence aims at getting the equivalent representation describing the inner relation among its composing elements. Processing these elements presents several challenges arising from problems caused by the inherent characteristics of ambiguity, both in words and sentences, non-determinism and context-sensitivity[3].

Context-sensitivity in natural language can occur either inside a syntagm, in the sentence, or in a sequence of sentences. In Portuguese, for example, all words that constitutes a noun syntagm, i.e, a syntagm that has characteristics of a noun, must have the same inflection regarding number and gender[5,6]. In a sentence with a transitive verb, an noun syntagm must be provided as an object. Luft[6] presents several syntactic rules for Brazilian Portuguese in a way that can be mapped into a formal language representation with little effort.

In this paper, the hierarchical relation among the elements in a sentence is modeled by a hierarchically organized rewriting system. Such system features structure processing in a sentence with characteristics of regular, context-free and context-sensitive languages in the Chomsky hierarchy, one at a time.

2.2. Adaptivity

Adaptivity is the term used to denote a phenomenon in which a rule-driven device spontaneously modifies its inner workings in order to accommodate planned yet unexpected situations, without external interference[7,8]. A device is called *adaptive* if such characteristic is present. For instance, a pawn that advances all the way to the opposite side of the chess board triggers a rule which promotes it to another piece of that player's choice; the pawn now has its own behavior changed to another piece and may act like so[9].

A rule-driven device $AD = (ND_0, AM)$, such that ND_0 is a device and AM is an adaptive mechanism, is said to be *adaptive* when, for all operation steps $k \geq 0$ (k is the value of an internal counter T starting in zero and incremented by one each time a non-null adaptive action is executed), AD follows the behavior of an underlying device ND_k until the start of an operation step $k + 1$ triggered by a non-null adaptive action, modifying the current rule set; in short, the execution of a non-null adaptive action in an operation step $k \geq 0$ makes the adaptive device AD evolve from an underlying device ND_k to ND_{k+1}[8]. Three types of elementary adaptive actions are defined in order to perform tests on the rule set or modify existing rules, namely: *(a)* inspection, for querying rules that match a certain pattern, *(b)* removal, for eliminating rules from the current rule set, and *(c)* insertion, for adding rules to the rule set[7,8].

According to Cereda and José Neto[10], adaptivity provides mechanisms for expressing abstractions more conveniently. As a direct consequence, several model improvements are made possible and practically viable, such as complexity reduction, problem partitioning, and hierarchical solving, available at almost no sensible cost to the user.

2.3. Rewriting systems

Rewriting systems (also referred as *reduction systems*) employ term transformations according to a set of substitution rules (also known as rewriting rules)[11]. Such systems are widely applied on several areas, including computability theory, word problem decidability, and theorem proving[12].

Generally, an *abstract reduction system* R is defined as $R = (A, I)$, such that A is the set of elements and I is a sequence of binary relations $\rightarrow_\alpha$ over A, also known as reduction (rewriting) relations. An abstract reduction system with only one reduction relation is known as *substitution system* or *transformation system*. If $a, b \in A$ and $(a, b) \in\rightarrow_\alpha$, such reduction relation is written as $a \rightarrow_\alpha b$ and b is said to be a (one step) α-reduction of a. Similarly, $a \rightarrow^*_\alpha b$, being $\rightarrow^*_\alpha$ the reflexive and transitive closure of $\rightarrow_\alpha$, if there exists a finite, potentially empty, sequence of reduction steps $a \equiv a_0 \rightarrow_\alpha a_1 \rightarrow_\alpha \ldots \rightarrow_\alpha a_n \equiv b$, such that $\equiv$ denotes the element identity of A. A term that cannot be rewritten is said to be in the *normal form*.

Rewriting termination (i.e, the situation in which there is no more rewriting rules to be applied in the term sequence) is, in general, a undecidable problem[13]. However, there are ongoing efforts towards applying rewriting restrictions such that a system is able to properly terminate (i.e, all terms in the sequence are in their corresponding normal forms). In particular, macros are a special type of rewriting system in which terms are checked against syntactic patterns[14]; when a certain rule pattern matches, the corresponding transformation is applied. Macro substitutions can be purely *symbolic* (the replacement sequence is applied ipsis litteris over the match) or *algorithmic* (interpretation and evaluation are required as part of the process)[15]. Besides, the metalanguage specification defines the operational semantics of macro expansion (e.g, recursion handling and pattern unfolding)[15,14].

Macros realize the concept of *textual abstraction*, in which certain text fragments can be removed from their contexts and replaced by a description of their common structure (and thus a higher abstraction level is achieved)[14]. This feature is desirable mainly when the sequence processing requires multiple views on each term (hierarchical problem solving).

3. Rewriting-based syntactic analysis

In order to handle context sensitivity in NLP while preserving model compaction, we introduce a syntactic analysis based on hierarchical rewriting. The analysis itself is partitioned into abstraction levels, such that each level aims at solving a specific subproblem. The functional composition of all levels, disposed in an hierarchical fashion, results in the expected, constructively described, syntactic analysis. Macro-based rewriting allows better correspondence and representation of all intermediate steps towards the solution.

The first level applies rewriting rules towards word segmentation, i.e, a lexical task of dividing a string into its component words. The rules for word segmentation usually follow a regular pattern and are highly dependent on the underlying language, such as using space as word divider and later removal of punctuation symbols for English and other languages using certain variations of the Latin alphabet. In this level, punctuation symbols and word dividers are explicitly removed by applying a symbolic transformation that reduces such elements to an empty symbol (namely, ϵ). Similarly, words are rewritten as a sequence of `ID` elements. An `ID` element holds the word value (i.e, the textual reference itself) in a set of properties (namely, a map). From an implementation point of view, regular patterns are easily covered by finite automata.

The second level, applies rewriting rules towards morphological analysis, i.e, categorization of `ID` elements (words) obtained in the previous level based on their linguistic structures. A single word may hold several categories according to the underlying language. All categories associated with a word are carried over to the next level for potential disambiguation (i.e, the proper identification of which meaning is used in context). As a direct impact, multiple sequences may arise from the rewriting process (more precisely, the result is a cartesian product of all word categories in the sequence). In this level, an algorithmic transformation is applied to each `ID` element obtained from the previous step (regular pattern matching), resulting in a sequence of categorized words. The transformation consists of a dictionary lookup based on the underlying language. From an implementation point of view, the regular pattern is easily implemented covered by a finite automaton and the dictionary lookup is implemented as a function call.

The third and final level applies contextual rewriting rules towards syntactic analysis, i.e, understanding of form, function and syntactic relationship of words in a sentence. Rewriting rules group specific language patterns into higher structures, such as noun syntagms, and are highly dependent on the underlying language. Observe that, in this level, patterns are most likely context-sensitive, as the syntactic elements are bound by their properties (e.g, a verb must be conjugated according to the subject). It is important to observe that some initiatives incorrectly label this purely syntactic scenario as statically semantic. Adaptivity plays a major role in identifying context-sensitive patterns, as the rewriting rules are able to group the corresponding syntactic elements accordingly. When a potential context is identified, adaptive actions establish the syntactic relationship among elements such that their property maps must (entirely or partially) match. The ultimate goal is the reduction of the entire sentence to a single `SENTENCE` element; otherwise, the sentence is either not properly addressed given the current rewriting rules (potential insufficient coverage) or simply wrong according to the underlying language.

It is important to note that our approach allows the direct representation of the applied macros as a tree structure, such that the matched patterns (left side) become leaves. Similarly, the transformation elements (right side) become subtree roots. Tree structures that, after successive rewriting transformations, do not reduce to `SENTENCE` as their main root are deemed invalid and thus discarded from the model. Formally, let there be a rewriting system $R = (A, M)$ such that A is the set of elements and M is the set of all available macros. For the sake of organization, $M = M_1 \cup M_2 \cup M_3$, such that M_i denotes the subset of macros available at level i. Additionally, $M_1 \cap M_2 \cap M_3 = \emptyset$, i.e, macros are restricted to their corresponding levels, such that model encapsulation and hierarchy are ensured. A rewriting-based syntactic analysis α over a sequence s is defined as an application of R on s, $\alpha(R, s) = s'$, such that all elements of s' are in the normal form. If $s' = \texttt{SENTENCE}$, the syntactic analysis was successfully achieved.

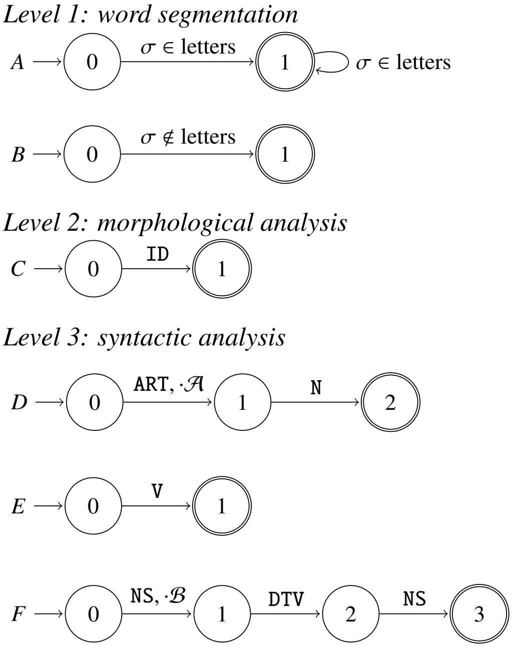
4. Experiments

Once the rewriting-based system is formally introced, we now proceed to perform a syntactic analysis of the sentence $s = \langle$ `o aluno respeita o professor` $\rangle$ in Portuguese. This example, taken from Luft[6], translates to $\langle$ `the student respects the teacher` $\rangle$ in English. Consider the macro subset for syntactic analysis of Portuguese sentences used in our experiments, presented in Figure 1.

The analysis begins with macros $A\downarrow$ and $B\downarrow$ (level 1 of Figure 1) being applied to s. Such macros perform word segmentation as the first level of the hierarchical processing of the input sequence of symbols. Figure 2 shows the resulting sentence s' composed by a sequence of `ID`s with their associated values containing the words.

In the next step, macro $C\downarrow$ (Figure 1, level 2) performs the morphological analysis by rewriting `ID`s from the previous level. The reduction consists of a dictionary lookup of the associated words, bringing their classification and other relevant properties. Observe that, as the word $\langle$ `o` $\rangle$ can be categorized as article or pronoun, all combinations from each category must be generated; due to space constraints, Figure 3 shows 2 possible output sentences s' resulting from the categorical ambiguity of $\langle$ `o` $\rangle$ (other trees were deliberatelly omitted).

Each possible sequence obtained from level 2 (Figure 3) is now subjected to macros $D\downarrow$, $E\downarrow$ and $F\downarrow$ in level 3 (Figure 1, level 3). First of all, macros $D\downarrow$ and $E\downarrow$ are applied in order to get the candidate syntagms; finally, macro F is applied to reduce the corresponding sentence to a single `SENTENCE` element, thus concluding the syntactic analysis (Figure 4). Macros $D\downarrow$ and $F\downarrow$ are constructed to handle context-sensitive grammar rules. The adaptive function $\mathcal{A}$ in macro $D\downarrow$ assures that the associated article (`ART`) and noun (`N`) that compose the noun syntagm (`NS`) agrees in singular number (represented by tag *sing*) and masculine gender (represented by tag *masc*). Similarly, the adaptive function $\mathcal{B}$ in macro $F\downarrow$ verifies if the direct transitive verb (`DTV`) morphology agrees with the properties of person (represented


Level 1: word segmentation
A
0
σ ∈ letters
1
σ ∈ letters
B
0
σ ∉ letters
1
Level 2: morphological analysis
C
0
ID
1
Level 3: syntactic analysis
D
0
ART, ·𝒜
1
N
2
E
0
V
1
F
0
NS, ·ℬ
1
DTV
2
NS
3

Syntactic pattern:
$w \mid w \in L(A) \Rightarrow \texttt{ID} : [\texttt{value} : w]$

Syntactic pattern:
$w \mid w \in L(B) \Rightarrow \epsilon$

Syntactic pattern:
$w \mid w \in L(C) \Rightarrow \texttt{dictionary}(w)$

Syntactic pattern:
$w \mid w \in L(D) \Rightarrow \texttt{NS} : [\texttt{value} : w, \texttt{properties}(w)]$

$\mathcal{A} = \{\ ?(0, \texttt{ART} : [?x], 1),\ -(1, \texttt{N}, 2),\ ?(1, \texttt{N} : [?x], 2)\ \}$

Syntactic pattern:
$w \mid w \in L(E) \Rightarrow \texttt{classify}(w)$

Syntactic pattern:
$w \mid w \in L(F) \Rightarrow \texttt{SENTENCE} : [\texttt{value} : w]$

$\mathcal{B} = \{\ ?(0, \texttt{NS} : [?x], 1),\ -(1, \texttt{DTV}, 2),\ ?(1, \texttt{DTV} : [?x], 2)\ \}$

Fig. 1. Macro subset for syntactic analysis of Portuguese sentences used in our experiments. Regular and context-sensitive patterns are implemented as finite and adaptive automata, respectively. Let $X\downarrow$ be the macro implementing the syntactic pattern represented by a device X.

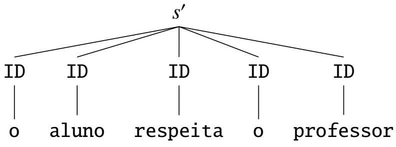


Fig. 2. Tree representation of s' after the first level rewriting. Observe that word dividers and punctuation symbols were discarded.

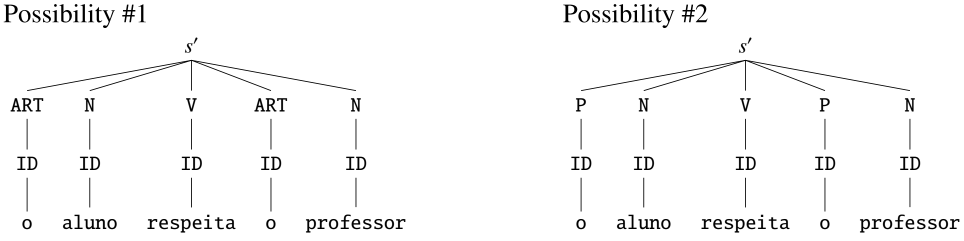


Fig. 3. Tree representations of s' after the second level rewriting. Observe that the word categorization resulted in multiple sequences.

by tag *per*) and number (represented by tag *num*) of NS provided that this information was obtained while processing the macro $D\downarrow$. The process of property retrieval for syntagms is beyond the scope of our paper.

As seen in Figure 4, since s' is now reduced to a single SENTENCE element, the original sentence s is said to be syntactically correct according to the Portuguese grammatical rules. Observe that the tree #2 from Figure 3 is discarded since there are no matches for macros $D\downarrow$, $E\downarrow$ and $F\downarrow$ from level 3, so SENTENCE is never obtained.

5. Final remarks

In this paper we presented a rule-based rewriting system implemented with adaptive technology for syntactic analysis of natural language sentences. The example presented in Section 4 demonstrated the flexibility of macro

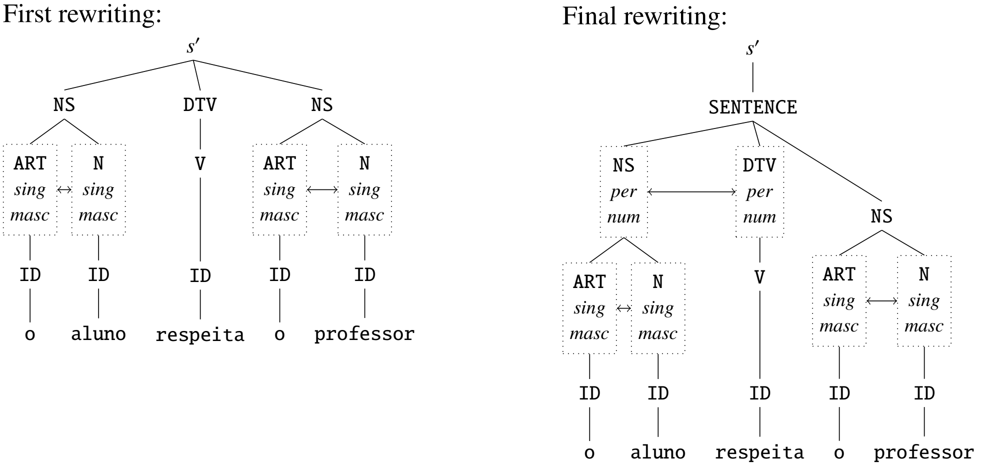


Fig. 4. Tree representation of s' after the third level rewriting. Observe that context-sensitive structures were correctly grouped. Since s' is now reduced to SENTENCE, the original sentence s is said to be syntactically correct according to the Portuguese grammatical rules.

usage for obtaining the corresponding syntax tree nodes while handling context sensitivity. Through adaptivity, the syntactic pattern was kept to a minimum, reducing verbosity and therefore improving legibility.

This approach can be further expanded to comprehensively cover other syntactic constructs by adding more macros to the current rewriting system. Additionally, the syntagm macros can be further enhanced in order to get additional properties from their elements, based on the underlying language structure.

Investigation can also be conducted towards handling potential combinatorial explosion when dealing with words with multiple categories (as seen in Figure 3, a single word with 2 categories would generate at least 4 trees at that point). Our hypothesis is that a context-sensitive macro (or a set of macros) located at a lower level might prevent associations that are not likely to occur based on the language formation rules, thus reducing the category candidates for a word. Further studies are needed.

Each level described in Section 3 reduces previously obtained elements to simpler yet meaningful representations. Since macros are restricted to their corresponding levels, model encapsulation and hierarchy are ensured. Moreover, the rewriting scheme has potential to be used in grammar-based compression for texts with repetitive syntactic patterns and restricted set of words.

References

1. J. Weizenbaum, ELIZA: a computer program for the study of natural language communication between man and machine, Communications of the ACM 9 (1) (1966) 36–45.
2. E. Lieberman, J.-B. Michel, J. Jackson, T. Tang, M. A. Nowak, Quantifying the evolutionary dynamics of language, Nature 449 (7163) (2007) 713–716.
3. D. Jurafsky, J. H. Martin, Speech and Language Processing, 2nd Edition, Prentice Hall, Inc., Upper Saddle River, NJ, USA, 2008.
4. M. Sipser, Introduction to the theory of computation, Thomson Course Technology, Boston, MA, USA, 2006.
5. E. Bechara, Moderna gramática portuguesa, 37th Edition, Nova Fronteira, Rio de Janeiro, RJ, Brazil, 2009, in Portuguese.
6. C. P. Luft, Moderna gramática brasileira: edição revista e atualizada, Editora Globo, São Paulo, SP, Brazil, 2002, in Portuguese.
7. J. José Neto, Adaptive automata for context -sensitive languages, SIGPLAN Notices 29 (9) (1994) 115–124.
8. J. José Neto, Adaptive rule-driven devices: general formulation and case study, in: International Conference on Implementation and Application of Automata, 2001.
9. P. R. M. Cereda, J. José Neto, A recommendation engine based on adaptive automata, in: Proceedings of the 17th International Conference on Enterprise Information Systems, Vol. 2, Barcelona, Spain, 2015, pp. 594–599.
10. P. R. M. Cereda, J. José Neto, A middleware architecture for adaptive devices, Procedia Computer Science 109 (2017) 1158–1163.
11. F. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, New York, NY, USA, 1998.
12. H. Friedman, M. Sheard, Elementary descent recursion and proof theory, Annals of Pure and Applied Logic 71 (1) (1995) 1–45.
13. M. Hermann, C. Kirchner, H. Kirchner, Implementations of term rewriting systems, The Computer Journal 34 (1) (1991) 20–33.
14. E. Kohlbecker, Syntactic extensions in the programming language LISP, Ph.D. thesis, Indiana University, Bloomington, IN, USA (1986).
15. C. Brabrand, M. I. Schwartzbach, Growing languages with metamorphic syntax macros, SIGPLAN Notices 37 (3) (2002) 31–40.