Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 1152–1157

Procedia Computer Science

www.elsevier.com/locate/procedia

International Workshop on Adaptive Technology (WAT 2017)

Adaptive Automata Applied to Natural Language Processing

Djalma Padovani*, João José Neto

School of Engineering of the University of São Paulo, Av. Prof. Luciano Gualberto, 380 - 05508-010 - São Paulo – SP, Brazil

Abstract

This work presents a brief review of the concepts of Adaptive Automata, detailing its mechanism of operation and main fields of application, highlighting the great potential of use in the field of natural language processing. Also, it is proposed Linguistico, an adaptive parser for Brazilian Portuguese, designed to dynamically modify its configuration, being able to insert and delete rules and change its behavior during execution. The effectiveness of the proposed framework is experimentally proven and the results obtained are comparable to the state of the art.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Self-Adaptive; Reconfiguration; Parsing; Grammars; Automata; Natural Language Processing; Computational Linguistics

1. Introduction

Adaptive Automaton is a state machine that successively changes its structure according to the application of adaptive actions associated with the rules of transitions performed by the automaton[1]. In this way, states and transitions can be eliminated or incorporated into the automaton as a result of each of the steps performed during the input analysis. In general, the Adaptive Automaton is formed by a conventional, non-adaptive device, and a set of adaptive mechanisms responsible for the self-modification of the system. The conventional device may be a grammar, an automaton, or any other device that respects a finite set of static rules. This device has a collection of rules, usually in the form of if-then clauses, which test the current situation against a specific configuration and take the device to its next situation. If no rule is applicable, an error condition is reported and operation of the device is

* Corresponding author. Tel.: +55-11-99909-8577.
E-mail address: djalma.padovani@ usp.br

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.386

discontinued. If there is a single rule applicable to the current situation, the next situation of the device is determined by the rule in question. If more than one rule adheres to the current situation of the device, all possible situations are handled in parallel and the device will display a non-deterministic operation. Adaptive mechanisms are formed by three types of elementary adaptive actions: consultation (inspection of the set of rules that define the device), exclusion (removal of some rule) and inclusion (addition of a new rule).

Natural Language Processing (NLP) requires the development of algorithms able to interpret the structure of the sentences at many levels of details, dealing with rules that are neither simple nor obvious, what makes computational processing complex. The text must be fractionated into lexical components and the syntactic role of these components should be determined, so that one can infer the semantics of the received text. There is no single way for natural language processing and the literature describes works using deterministic, statistical or heuristic approaches. One of the central problems is the need of high performance computing, due to the several possibilities of text interpretation and the need of answers in feasible time. Other main difficulty refers to the analysis of ambiguities and non-determinisms[2].

Adaptive Automaton has great potential of use in NLP due to the easiness with which it can represent complex linguistic situations such as ambiguities and non-determinisms. In addition, it can be implemented as recognition formalism, to preprocessing texts for a variety of scenarios, such as: syntax analysis, syntax checking, automatic translation processing, text interpretation and computer-aided language learning. The general form of the Adaptive Automaton allows dealing with the various classes of languages in the Chomsky's hierarchy[3,4]. Regular constructions are handled through finite state automata, without using push-down or adaptive actions. Context-free constructs are handled through push down automata. Context-dependent constructs are handled by adaptive actions that allow the model to change its topology, without the need of external elements to the model. In the more general case, the Adaptive Automaton is capable of dealing with recursively enumerable constructs (such as Analyzers), due to its equivalence with the Turing Machine[5,1].

2. Motivation

Regarding the field of morphologic-syntactic analysis, Menezes and Neto[6] present a method for the construction of a morphological tagger, which can be used in several languages. The authors say that all methods use three sources of linguistic information, extracted from a training corpus: word suffixes, as part of the inference process of the morphological tag of unknown words; a list of words associated with morphological categories (lexicon), to provide information on known words; and the context next to the lexical item that one wants to label (2 or 3 labels around), to refine the choice of its label. Thus, the proposed method first labels the words found in the lexicon; then uses heuristics applied to the suffixes to label the words not found in the lexicon; and finally does a refinement, according to the context.

Bonfante and Nunes[7] propose a probabilistic parser based on the notion of lexical nuclei, where, for each rule observed in the training set, non-core words are called modifiers, exerting influence on it. According to the authors, the great difficulty of specifying a grammar with descriptive power paved the way for empirical research. Thus, a set of syntactically annotated sentences is used, as training data, in a learning process to annotate unknown sentences. The formation of the syntactic structure of a sentence occurs through a bottom-up process commanded by the probability that a nucleus and a modifier come together to form a phrase.

Julia, Seabra and Semeghini-Siqueira[8] propose a parser that performs the syntactic and semantic analysis of statements about software specification expressed unrestrictedly in the natural language. The proposed parser corresponds to a structure that automatically generates semantic rules during the analysis, through a heuristic method. According to the authors, a structure is a system of transformations characterized by a set of rules. The syntactic part of grammar is expressed through rules, such as the rules of grammar proposed by Chomsky [4]. The parser is based on search algorithms that aim to find a path from the syntax tree to a leaf node that contains a category of meaning. The category of each word in the sentence will depend on the order in which it appears in the sentence.

Bick[9] presents a research developed by the VISL project - Visual Interactive Syntax Learning, based at the University of Southern Denmark, which also uses the deterministic approach in the development of the parser PALAVRAS, a reducing analyzer that selects labels based on constriction rules of the Constraint Grammar proposed by Karlson [10]. PALAVRAS seeks to disambiguate possible morphological interpretations through the application of rules that use contextual conditions to restrict the possible classifications, selecting, in the end, the most appropriate label. Bick explains that at the syntactic level, the parser works with productive and restrictive rules, the former map ambiguous tags and the latter reject labels based on context.

A statistical approach is presented by Marques and Lopes[11]. The authors argue that to carry out the morphological-syntactic tagging task it is required to manually annotate a large volume text, with hundreds of thousands of disambiguated words and they say there is no corpora with the necessary dimensions in Portuguese, nor the availability for the construction of a corpus with these characteristics. The authors also refute the construction of a rules-based system, as they also require the interference of a person with knowledge to provide rules so specific and dependent on the text that the system will process. The alternative they propose is the use of a neural network combined with a lexical analysis system and a manually labeled text.

This work proposes a computational model adaptable to the different approaches of natural language processing: deterministic, statistical or heuristic. Due to the scope of the theme, the model will be restricted to the stages of lexical-morphological recognition and syntactic recognition of texts written in the cultured pattern of the Portuguese Language of Brazil. Regional and temporal fluctuations, dialects and colloquial language will not be considered. The adaptive formalism was chosen as the underlying theoretical model due to its richness of representation and manipulation, which makes it consistent and flexible at the same time, providing the necessary foundation for the construction of the proposed computational model, without the need of auxiliary techniques.

3. Linguistico – An Adaptive Parser

The Laboratory of Adaptive Languages and Techniques has developed several works in natural language processing, specifically for Brazilian Portuguese [2, 12, 13]. Linguistico is a proposal of grammatical parser structured according the architecture previously described. It is composed by five sequential modules that carry out a specialized processing, working as a production line, in which each module sends the results obtained to the next one, until the text is completely analyzed (See Fig.1).

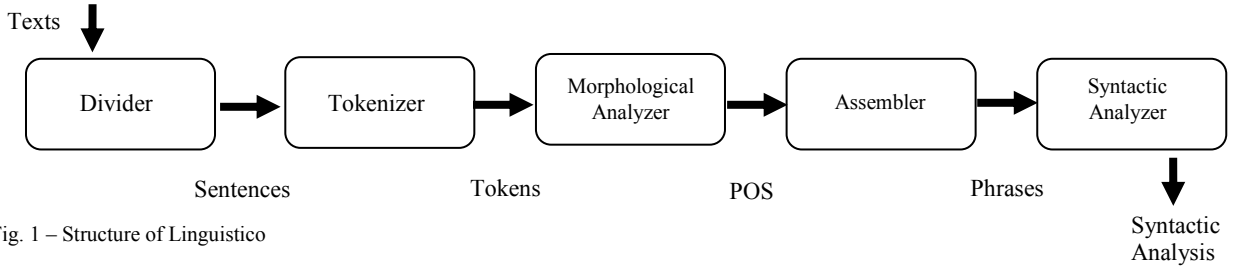
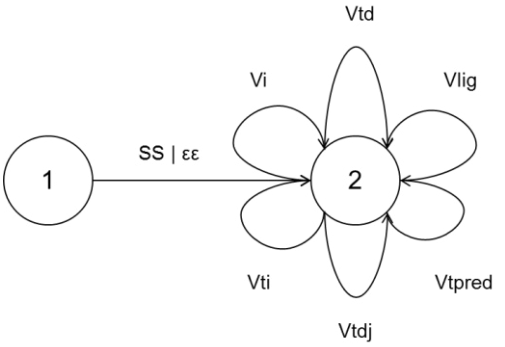


Fig. 1 – Structure of Linguistico

The first module, called Divider, receives a text as an input, identifies characters that can indicate end of sentence, abbreviations and compound words, and then divides the text into sentences. The second module, named Tokenizer, receives the sentences identified in the previous step and divides them into tokens, considering abbreviations, monetary values, hours, minutes, numerals, compound words, proper names, special characters and final punctuation. Tokens are stored in data structures (arrays) and sent one by one for analysis of the next module. The third module, called Morphological Analyzer, is composed by a Master Automaton and a set of specialized submachines that access databases with rules to identify morphological classifications. The priority is to obtain the classifications of known words in a lexicon; if the searched term is not found, the Morphological Analyzer searches for nouns, adjectives and verbs in the finite and infinite format (flexed and non-flexed) through a sub-machine specialized in the formation of words; finally, the Morphological Analyzer looks for invariant terms, that is, terms whose morphological classification is considered stable by linguists, such as conjunctions, prepositions and

pronouns. Another sub-machine is used to disambiguate morphological classifications, capturing the context of the tokens through trigrams, bigrams and unigrams and using a statistical approach to decide the better classification for the token. The fourth module, called Assembler, is composed by an automaton responsible for assembling the phrases from the part-of-speech obtained from the previous step. The Assembler put together the parts-of-speeches, identifying noun, verbal, prepositional, adjective and adverb phrases. The fifth and last module, called Syntactic Analyzer, receives the phrases from the previous module and looks for a valid production. If more than one is found, the Syntactic Analyzer chooses the most suitable for the context, comparing the probabilities of the productions. The Syntactic Analyzer uses an adaptive automaton to make recursive calls, storing the state and the chain of recognized tokens up to the time of the call in a stack structure. If the Syntactic Analyzer is unable to move from the received sentences, it generates an error and returns the pointer to the last recognized syntax, terminating the instance of the recursive automaton and returning the processing to the one that initialized it.



α (j): { o* : - [(j, Vlig)]
+ [(j, Vlig) :→ o, β (o)]}

β (o): { t*u*v*x*: + [(o, SP) :→ t]
+ [(o, Sadv) :→ u]
+ [(o, Sadj) :→ v]
+ [(o, SS) :→ x] }

Fig. 2 – Initial configuration of the Automaton.

Fig.2 presents an example of the initial configuration of the Automaton and the adaptive functions used to modify it when interpreting a Vlig token (linking verb). The adaptive function α (j) is called by the automaton before processing the token Vlig. It receives Vlig as parameter (j = Vlig), creates the state 11 and the transition that takes the state 2 to the state 11. The same function eliminates the transition to the state 2, when it receives Vlig. Then, the automaton calls the function β (o), passing the state 11 as parameter (o = 11). It creates the states 12, 13, 14 and 15 and the productions that interconnect state 11 to the new states. Fig.3 shows a fragment of the rules obtained.

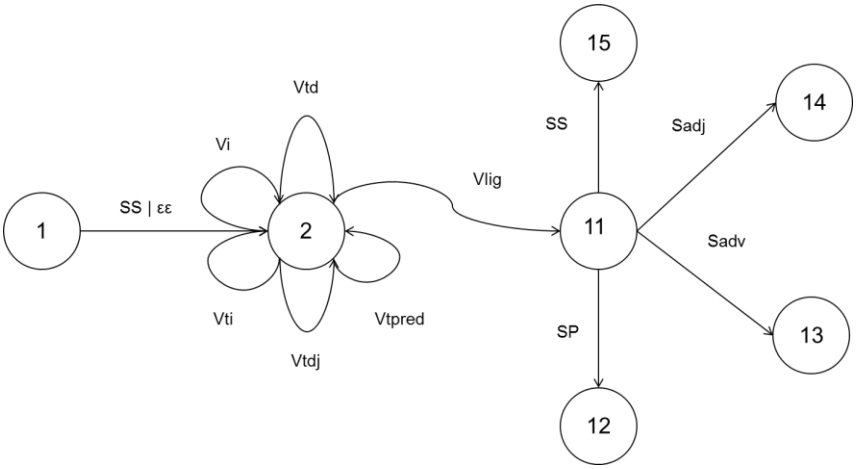


Fig. 3 – Configuration of the Automaton after interpreting the VLig token.

4. Experiments and Results

Some experiments were performed to test Linguistico and two modules, Morphological Analyzer and Syntactic Analyzer, were specifically evaluated because they were designed based on Adaptive Automaton approach. CINTIL-Treebank[14], a Portuguese Language corpus developed by the University of Lisbon, was chosen as grammar support, as it uses the same notation system of the University of Pennsylvania - Treebank, which is considered standard to natural language processing[15], and relies on a structural and non-descriptive grammar. The modules Divider, Tokenizer and Assembler were implemented exactly as described in the previous section.

The Morphological Analyzer was implemented according to the structure of Master Automaton and specialized submachines. A Viterbi algorithm was used to implement the disambiguation submachine, taking into consideration trigrams, bigrams and unigrams, and assuming the most common classification of the corpus - noun - when the classification of the token was not found. The Morphological Analyzer was trained with 90% of the CINTIL-Treebank corpus and tested with the remainder 10%. At the end, the Morphological Analyzer presented a precision of 91.41%. The results obtained were comparable to those of general purposed analyzers, but below than those presented by the specialized analyzers in Portuguese Language [15, 16], that reach 97,09 % of precision (See Table1). Nevertheless, there are opportunities for enhancement, since it is possible to increase the size of the training corpus and to include more contextual information for disambiguation.

Table 1. Comparison of Morphological Analyzers

Morphological Analyzers	Linguístico	TBL	TnT	MXPost	QTag
Precision	91,41%	97,09%	96,87%	97,08%	89,97%

The Syntactic Analyzer was implemented with production rules obtained from CINTIL-Treebank corpus and the X-Bar Theory[17], on which the corpus is based. A subset of 90% of CINTIL-Treebank corpus was used to identify the productions rules, which were enriched with the probabilities of their occurrences and adapted to avoid repetition. They were also configured with the encoding used in the Portuguese language (ISO-Latin-1). Fig.4 shows a fragment of the rules obtained.

```
AP -> A CONJP [0.00766]                VP -> V ADV [0.01065]
ADV_ -> PNT ADV_ PNT [0.05036]         A_ -> ADV A_ [0.01639]
S -> V_ AP [0.00019]                   AP -> AP CP [0.00128]
AP -> ADV_ A [0.00766]                 NP -> NP QNT [0.00003]
```

Fig. 4 – Fragment of Rules obtained from CINTIL– Treebank.

The results were measured using the Parseval method [18], considered standard to determine the degree of correction of the derivation trees generated by parsers. The tests were performed with the 10% of CINTIL-Treebank corpus. The FParseval index was 88.74%, above, therefore, than the results presented by the parsers Bikel and Stanford and bellow than Berkeley's, when applied to the same corpus, reaching levels similar to those obtained for the English language [15] (See Table 2).

Table 2. Comparison of Syntactic Analyzers

Syntactic Analyzers	Linguístico	Bikel	Stanford	Berkeley
FParseval	88,74 %	84,97%	88,07%	89,33%,

5. Conclusions

This article presented a brief review of the concepts of Adaptive Automaton, detailing its mechanism of operation and main fields of application, highlighting the great potential of use in the field of natural language processing. It

was also proposed Linguistico, a grammatical parser that uses Adaptive Automaton as underlying technology. The effectiveness of the proposed framework was experimentally proven and two modules were specifically evaluated: the Morphological Analyzer and the Syntactic Analyzer. The Morphological Analyzer obtained results comparable to those of general purpose analyzers, but below than those presented by the specialized analyzers in Portuguese Language. Improvements will likely, since it is possible to increase the size of the training corpus and to include more contextual information for disambiguation. The Syntactic Analyzer obtained better results than some of the parsers of the state of art applied to texts in Portuguese and reached levels similar to those presented by analyzes of texts in English. These initial results encourage us to go further in the investigation, implementing the identified improvements and testing Linguistico with other corpus. We also planned to widen the scope of the research, including aspects of context dependency, equivalences between different grammars, dialects and regionalisms. Finally it is our expectation to enhance the adaptive architecture, incorporating mechanisms of choice of computational models, evaluation criteria and transition rules.

References

1. Neto, J. J. Contribuições à metodologia de construção de compiladores. Tese de Livre Docência, EPUSP, São Paulo, 1993. *
2. Rocha, R. L. A. Tecnologia Adaptativa Aplicada ao Processamento Computacional de Língua Natural. Revista IEEE América Latina. Vol. 5, Num. 7, ISSN: 1548-0992, pp. 544-551, Novembro 2007. *
3. J. J. Neto. Adaptive Automata for Context -Sensitive Languages. ACM SIGPLAN NOTICES, Vol. 29, n. 9, pp. 115-124, September, 1994.
4. Chomsky, N. Three models for the description of language. IRE Transactions on Information Theory 2: 113-124, 1956.
5. R. L. A. Rocha e J. J. Neto. Autômato adaptativo, limites e complexidade em comparação com máquina de Turing. In: Proceedings of the second Congress of Logic Applied to Technology – LAPTEC'2000. São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, p. 33-48, 2001. *
6. C. E. D. Menezes e J. J. Neto. Um método para a construção de analisadores morfológicos, aplicado à língua portuguesa, baseado em autômatos adaptativos. V PROPOR, Encontro para o Processamento Computacional de Português Escrito e Falado, 2000, Atibaia, Brasil. *
7. Bonfante, A. G.; Nunes, M. G. V. Parsing Probabilístico para o Português do Brasil. In: I Workshop de Teses e Dissertações em Inteligência Artificial (I WTDIA), 2002, Porto de Galinhas - Recife. I Workshop de Teses e Dissertações em Inteligência Artificial (I WTDIA), 2002. *
8. Julia, R. M. S.; Seabra, J. R.; Semeghini-Siqueira, I. An Intelligent Parser that Automatically Generates Semantic Rules during Syntatic and Semantic Analysis. In: IEEE International Conference on Systems, Man and Cybernetics, 1995, Vancouver. v. i. p. 806-811.
9. Bick, E. The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework, 2000.PhD. Thesis, Arhus University.
10. KARLSON, F. Constraint Grammar as a Framework for Parsing Running Text. 13o. International Conference on Computational Linguistics,1990, Helsinki (Vol.3, p.168-173).
11. Marques, N.M.C.; Lopes, J.G.P. Redes Neuronais e um Léxico na Etiquetagem Morfossintática para o Estudo da Subcategorização Verbal. In: Sardinha, T. B. A Língua Portuguesa no Computador. 295p. Mercado de Letras, 2005. P. 71-90.*
12. J. J. Neto, and M. Moraes. Using Adaptive Formalisms to Describe Context-Dependencies in Natural Language. Lecture Notes in Artificial Intelligence. N.J. Mamede, J. Baptista, I. Trancoso, M. das Graças, V. Nunes (Eds.): Computational Processing of the Portuguese Language 6th International Workshop, PROPOR 2003, Volume 2721, Faro, Portugal, June 26-27, Springer-Verlag, 2003, pp 94-97.
13. J. J. Neto, e M. Moraes. Formalismo adaptativo aplicado ao reconhecimento de linguagem natural. Anais da Conferencia Iberoamericana en Sistemas, Cibernética e Informática, 19-21 de Julio, 2002, Orlando, Florida. *
14. Branco, António, Francisco Costa, João Silva, Sara Silveira, Sérgio Castro, Mariana Avelãs, Clara Pinto and João Graça, 2010, "Developing a Deep Linguistic Databank Supporting a Collection of Treebanks: the CINTIL DeepGramBank ", In Proceedings, LREC2010 - The 7th international conference on Language Resources and Evaluation, La Valleta, Malta, May 19-21, 2010.
15.Branco, A., Silva, J.: Evaluating solutions for the rapid development of state-of-the-art POS taggers for Portuguese. In: Proceedings of the 4th Language Resources and Evaluation Conference (LREC). (2004) 507–510.
16.Silva, J.: Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, University of Lisbon (2007) Published as Technical Report DI-FCUL-TR-07-16.
17.Mioto C., Silva M. C. F., Lopes, R. Novo Manual de Sintaxe. Ed. Contexto. 2013. *
18. Black, E., Abney, S., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Marcus, M., Santorini, B.: A procedure for quantitatively comparing the syntactic coverage of English grammars. In: Proceedings of the Workshop on the Evaluation of Parsing Systems. (1991) 306–311.
* In Portuguese.