Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2569

SPECIFICATION OF THE UNL-PORTUGUESE ENCONVERTER-DECONVERTER PROTOTYPE

RONALDO T. MARTINS
RICARDO HASEGAWA
LUCIA H.M. RINO
OSVALDO N. OLIVEIRA JR.
M. GRAÇAS V. NUNES

Nº 63

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Out./1997

Universidade de São Paulo - USP
Universidade Federal de São Carlos - UFSCar
Universidade Estadual Paulista - UNESP

Specification of the UNL-Portuguese enconverter-deconverter prototype

Ronaldo T.Martins Lucia H.M.Rino Osvaldo N.Oliveira Jr.
Ricardo Hasegawa M.Graças V.Nunes

NILC-TR-97-1

October, 1997

UNL-Brazil Project

Computacional Linguistics Interinstitucional Nucleus Report Series
NILC - ICMSC-USP, Caixa Postal 668, 13560-970 São Carlos, SP, Brazil

Specification of the UNL-Portuguese enconverter-deconverter prototype

Ronaldo T. Martins Lucia H.M.Rino[*] Osvaldo N.Oliveira Jr.[#]
Ricardo Hasegawa M. Graças V. Nunes[%]

NILC-São Carlos
UNL-Brazil Project

Abstract

This technical report contains a brief description of a prototype system for converting Brazilian Portuguese into the Universal Networking Language (UNL) and deconverting UNL representations into Brazilian Portuguese. This system was developed by the brazilian team of the UNL Project, specifically the Computational Linguistics Interinstitutional Group of São Carlos (NILC).

Resumo

Este relatório técnico descreve um sistema protótipo para codificação do Português do Brasil em Universal Networking Language (UNL) e decodificação de UNL para Português do Brasil. Este sistema foi desenvolvido pela equipe brasileira participante do Projeto UNL, mais especificamente, pela equipe do Núcleo Interinstitucional de Linguística Computacional de São Carlos (NILC).

1. Introduction

This technical report contains a brief description of a prototype system for converting Brazilian Portuguese into the Universal Networking Language (UNL) and deconverting UNL representations into Brazilian Portuguese. Details of the UNL approach will only be given in the context of the prototype development. Details of the UNL formalism can be found in [1] or in the internet sites www.unl.ias.unu.edu, www.unl.cits.br, www.nilc.icmsc.sc.usp.br/projects/unl-br/unl_br. The philosophy

[*] Departamento de Computação - UFSCar
[#] Instituto de Física de São Carlos - USP
[%] Departamento de Computação - ICMSC-USP

adopted here is similar to the prototyping strategy employed in software engineering in that results from assessing a prototype will be used to guide the project in later stages. We have therefore chosen in this initial stage to develop the enconverting-deconverting system using standard procedures used in natural language processing. This approach is to be followed by a detailed analysis of its limitations and strengths and its applicability to the Portuguese language.

Experience from machine translation has shown that at least three independent modules for language processing are required, as follows: the lexical, the syntactic and the semantic modules. In the system designed, these three modules will be used by both, the "enconverter" (Portuguese → UNL) and the "deconverter" (UNL → Portuguese) modules, with no *a priori* specification for the internal architecture of the modules. The general architecture for the system is depicted in Fig. 1. Each of its modules is presented in the following sections.

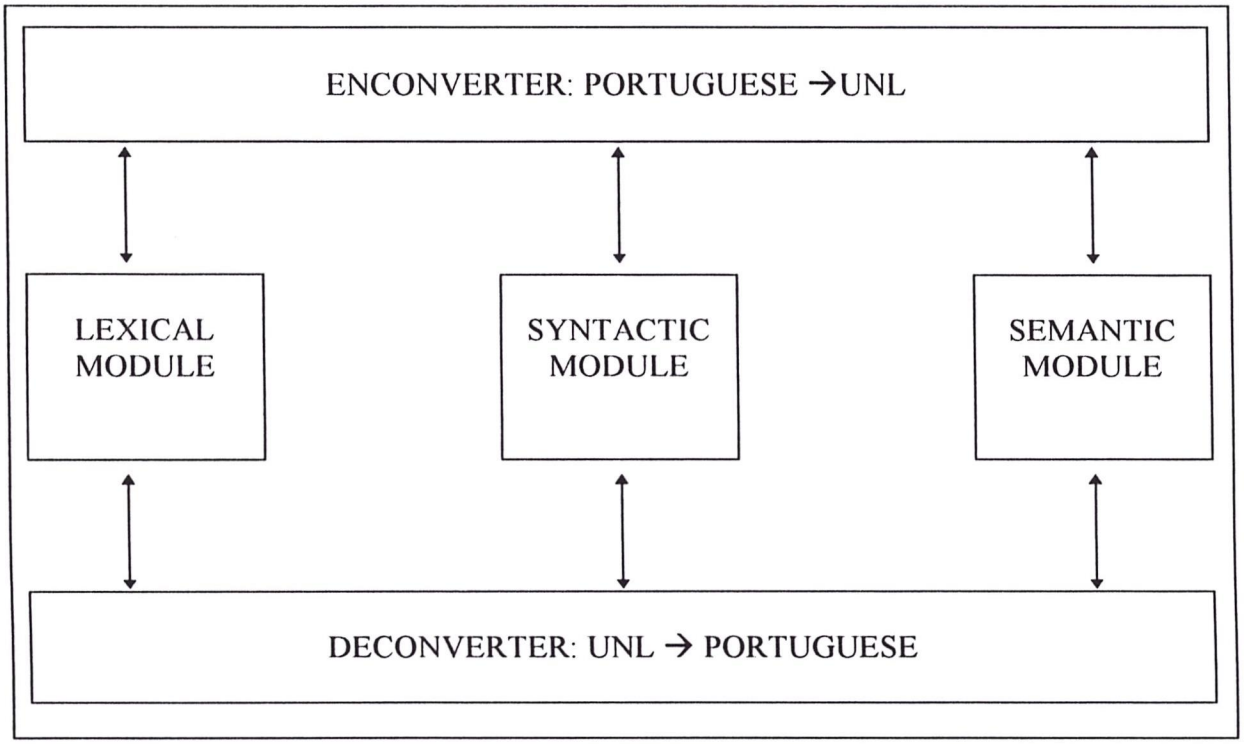

Fig. 1 - General Architecture of the Prototype

2. The Lexical Module

This module encompasses five submodules, as illustrated in Fig. 2: i) the lexicon, ii) a table for converting Portuguese into attribute labels (Portuguese → AL Table), iii) a lexical analyzer, iv) a lexical generator and v) a lexical disambiguating system. The two first submodules represent the database on which the enconverter and deconverter are

to operate. The lexical analyzer performs morphological analysis and converts words from Portuguese into universal words (UWs) and attribute labels (ALs), as needed. The lexical generator carries out the corresponding converse task, by converting UWs and ALs into words in Portuguese. Both, the lexical analyzer and generator rely on the disambiguating system for choosing the appropriate gramatical classes and gramatical and semantic attributes of the Portuguese words. They also map UWs and ALs onto Portuguese expressions, as adequate.

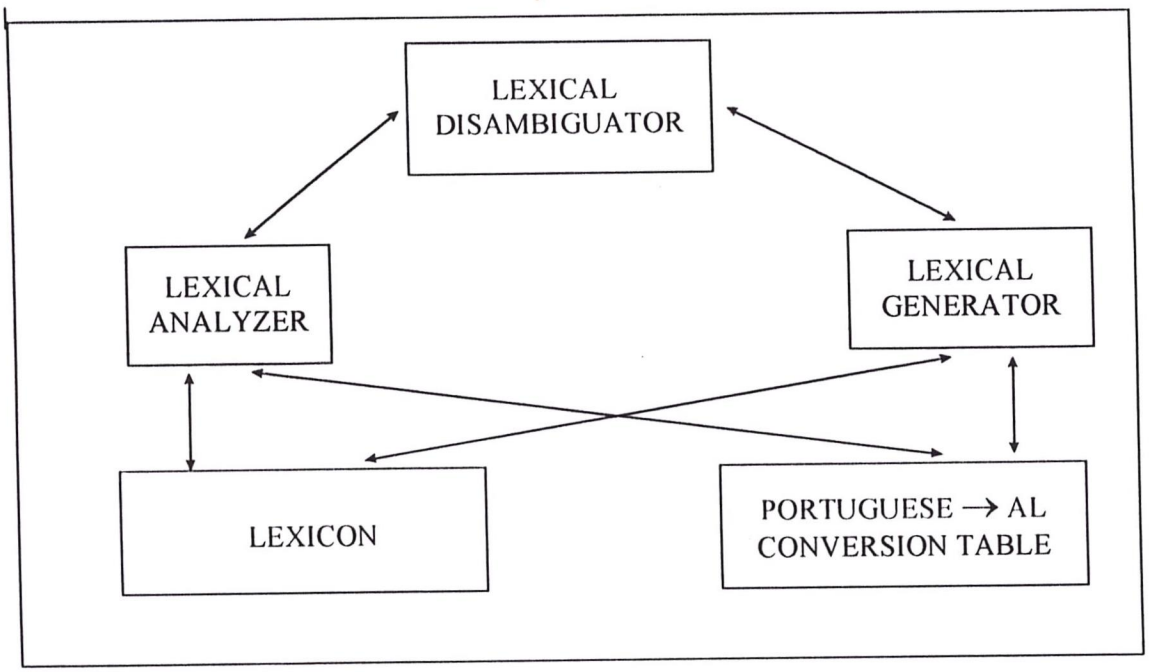

Fig. 2 - Five submodules of the lexical analyzer

The internal representation of the lexicon is practically the same as that of the dictionary of UWs. Access to the lexicon is made through chosen patterned words, which are non-inflected Portuguese words. For instance, in its patterned form, a noun, or an adjective, is represented according to its number and gender, i.e., *singular, masculine*. For a verb, the infinitive representation is adopted. The lexicon is so indexed to allow for access of any UW that corresponds to a Portuguese word (which is either inflected or not in this case) or vice-versa, i.e., to retrieve a Portuguese word and its semantic and gramatical attributes from a given UW. In addition, collocations that are highly frequent in Portuguese texts are also represented as single units in the lexicon (e.g., *por outro lado* , which means *On the other hand* in English, is represented by the UW on_the_other_ hand). Pieces of information related to lexical items can be: the Portuguese words themselves, their corresponding UWs, gramatical roles (noun, adjective, verb, etc.), morphological categories (number, gender, tense), syntactic roles (subject, predicate, etc.), and semantic attributes.

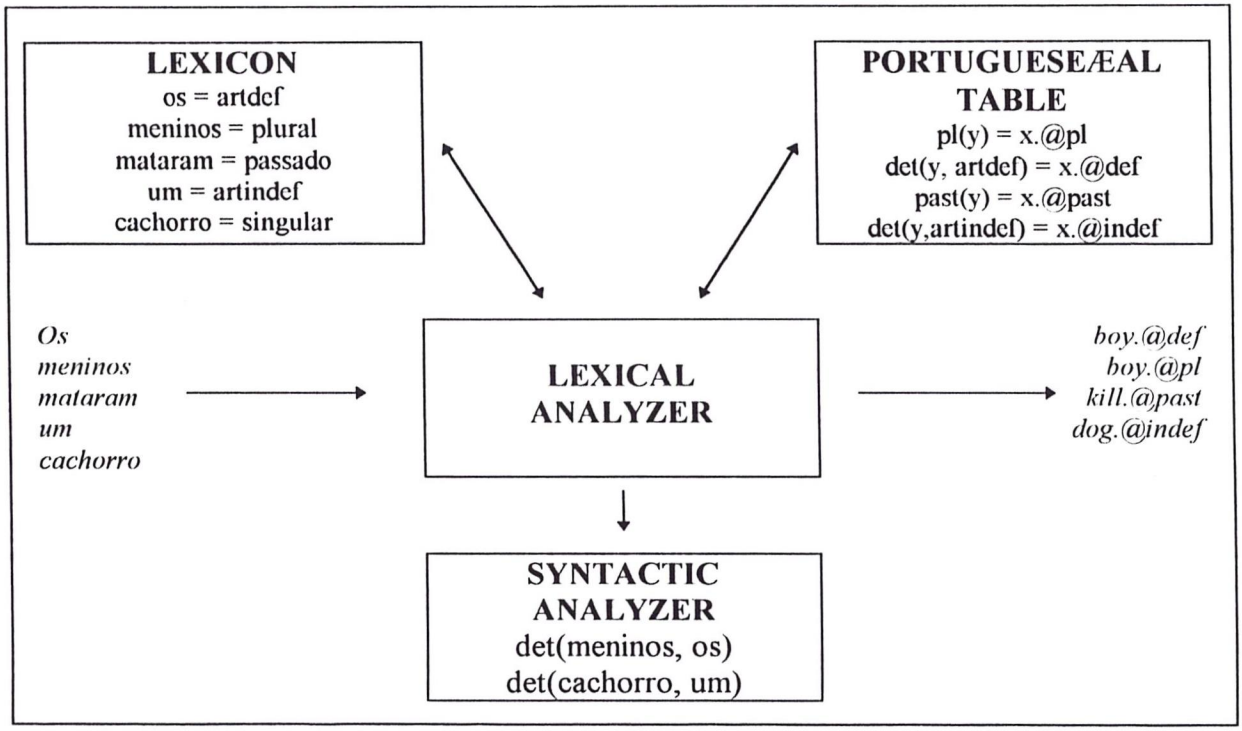

Fig. 3 - Example of the lexical analyzer operation

The Portuguese→AL Table holds the correspondence between ALs and Portuguese representations (gramatical and syntactic roles) which is designed to allow the automatic processing of the morphological categories for both the enconverter (by means of the lexical analyzer) and the deconverter (by means of the lexical generator). The lexical analyzer is required for encoding nouns and verbs from Portuguese into UNL. Two procedures are necessary for this: i) a morphological analysis operating directly on the lexicon and on the Portuguese→AL Table, for obtaining the attribute labels (ALs) corresponding to Portuguese words (such as number - for a noun - and tense - for a verb); ii) a syntactic analysis, for obtaining ALs that are not apparent in the morphological representation of the words, like the head of a noun phrase. This task is performed by getting information from both, the syntactic analyzer and the Portuguese→AL Table. The behavior of the lexical analyzer upon receiving a Portuguese sentence is exemplified in Fig. 3.

The lexical generator operates in precisely the opposite direction to the lexical analyzer, by converting lexical attributes represented in UNL into Portuguese words. Two steps are also followed: i) a morphological synthesis through which inflected Portuguese words are obtained from UWs; ii) the fulfillment of syntactic roles by means

of ALs that bring information about additional categories represented in the UWs. The underlying processes of the lexical generator are illustrated in Fig. 4.

A lexical disambiguating system is needed in the prototype because Portuguese words, or even UWs, can have more than one classification, as illustrated below:

a) Grammatical category:

Portuguese: *canto* (noun), *canto* (verb)

UNL: *cause* (noun), *cause* (verb)

b) Morphology:

Portuguese: *lápis* (sing), *lápis* (plural)

Portuguese: *dentista* (masc), *dentista* (fem)

UNL: presumably none.

c) Syntax:

Portuguese: *ser* (auxiliary), *ser* (intransitive)

UNL: presumably none.

d) Semantics:

Portuguese: *manga* (sleeve), *manga* (mango)

UNL: book (publication), book (from accounting)

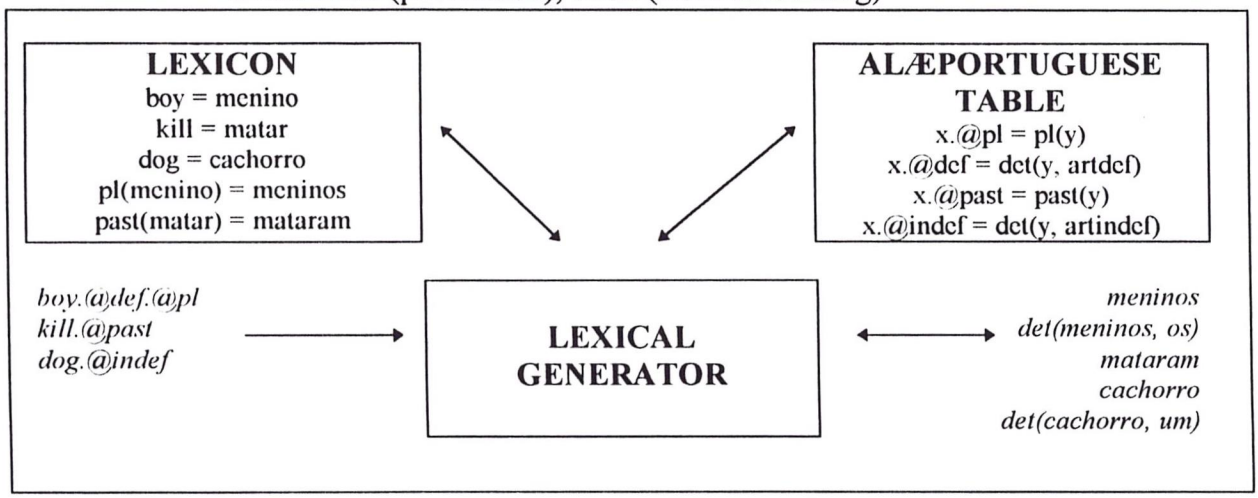

Fig. 4 - Example of the lexical generator operation

In developing the disambiguating system we are taking advantage of the previous experience of the group in developing ReGra [2], a grammar checker for Brazilian Portuguese. However, extensions of ReGra have been investigated, specially concerning the treatment of semantics, since semantic attributes were not available in its lexicon. The disambiguator is to be accessed when either the lexical analyzer, or the lexical generator, and the syntactic analyzer access the lexicon to look for a classification of a word and find more than one option.

3. The Syntactic Module

The syntactic module, depicted in Fig. 5, also includes five submodules: a table of syntactic production rules, a parser (for automatic syntactic analysis), a generator of syntactic structures, a disambiguator of syntactic roles, and a syntactic regularizer. The table of production rules contains, in principle, all the syntactic rules to generate correct constructions in Brazilian Portuguese. The starting point here was a table based on a context-free grammar developed for ReGra. Agreement relations have not been employed to constrain syntactic variations, since these were precisely the relations to be checked by ReGra. Currently, the table is being expanded and modified by introducing such agreement features.

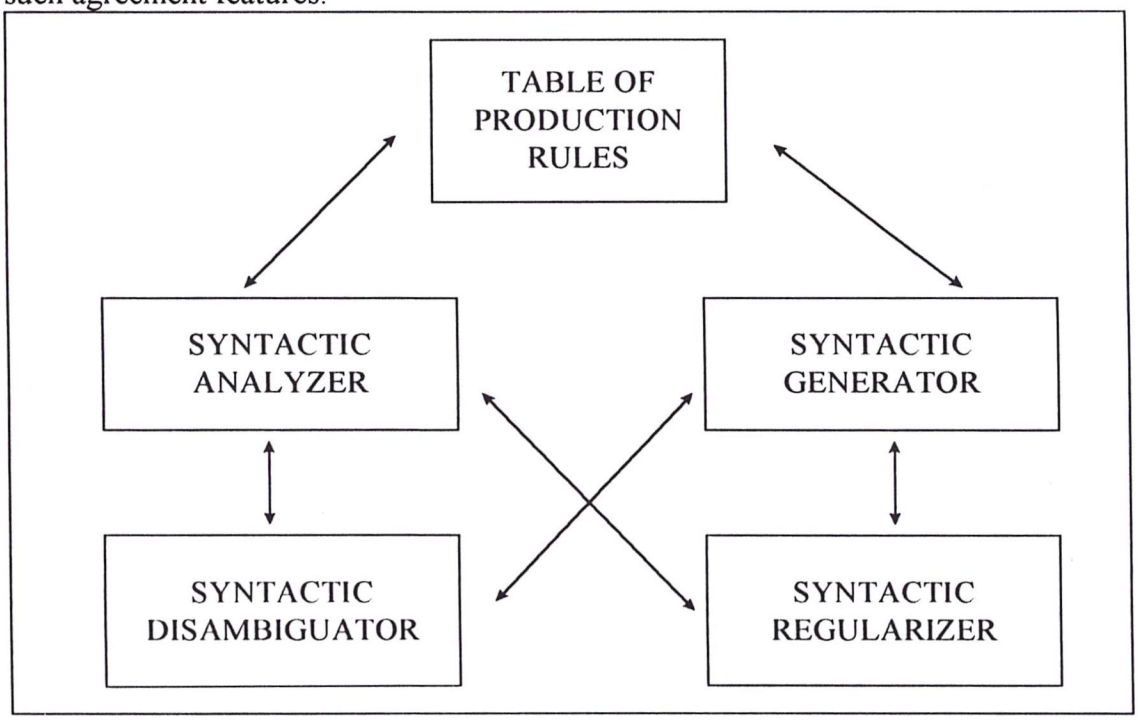

Fig. 5 - The syntactic module

The syntactic analyzer is also based on the one available for ReGra. This is a top-down left-to-right parser that includes some lookahead and backtracking strategies up to the last decision-making point. These strategies are used when a match to a syntactic construction is not successful. The analyzer identifies the syntactic role of each lexical item of an input sentence, including all its dependence relations that are essential for the semantic analyzer. Fig. 6 illustrates the functioning of the syntactic analyzer. It may be interesting - in later stages of this project - to introduce strategies for bottom-up parsing to cope with cases of mismatch and syntactic ambiguity.

Furthermore, use could be made of a parsing model which analyzes individual nuclei, particularly useful for generation.

Obviously, ambiguities appear in establishing the syntactic roles for some words. Therefore, a fourth submodule, the syntactic disambiguator, is required. Two basic functions of this module are: a) to select one among several possible matches of a given sentence in Portuguese; b) to select the most adequate syntactic role of each semantic attribute in the UNL representation, when non-biunivocal correspondences between syntax and semantics occur. The disambiguator is activated by both syntactic and semantic analyzers.

The syntactic generator was designed to convert UNL semantic relations into Brazilian Portuguese syntactic roles. Agreement relations are then checked by the syntactic regularizer, aiming at producing a grammatically correct output. This system is also responsible for filling in the appropriate gaps of information when binding is done and for taking into account ellipses and anaphora resolution. In other words, the syntactic generator is responsible for retrieving the deep structure of a UNL input representation.

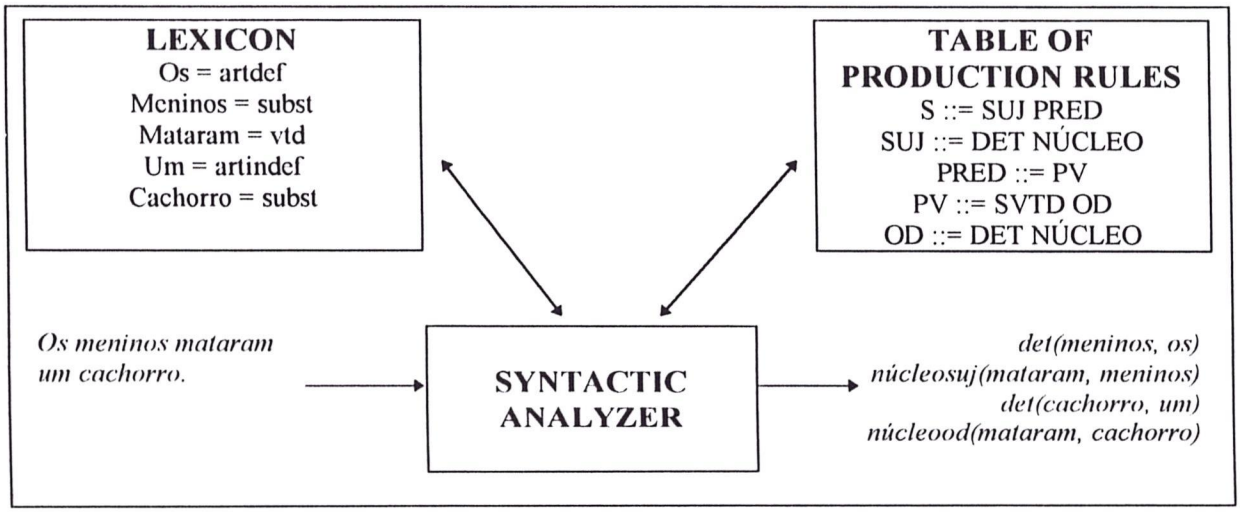

Fig. 6 - The syntactic analyzer

4. The Semantic Module

The semantic module comprises three submodules (see Fig. 7), as follows: a table for converting Portuguese into relation labels (Portuguese→RL Table), a semantic analyzer, and a semantic disambiguator. The Portuguese→RL Table stores semantic relations among syntactic variations corresponding to conceptual entities in a sentence written in Brazilian Portuguese. It maps syntax into semantics, i.e., semantic attributes are retrieved based on the syntactic description of Portuguese lexical items.

In addition, such mapping allows for syntactic attributes to be obtained from semantic relations indicated by the UNL. The semantic analyzer is responsible for transducing syntactic information into semantic attributes, by operating on the output of the syntactic analyzer to provide the necessary semantic relations. For example, the analyzer can retrieve relationships between words connected by binding relations, such as *agent* (agt) and *object* (obj).

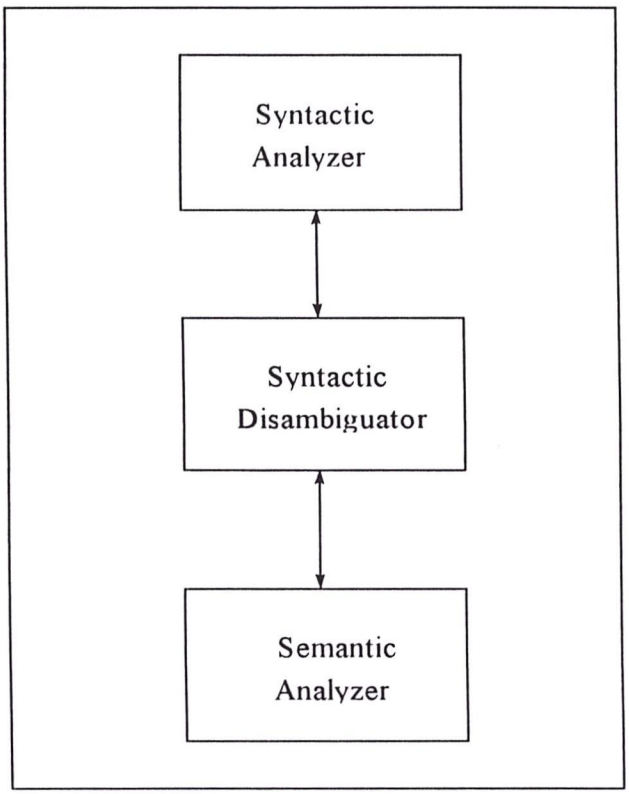

Fig. 7 - The semantic module

The semantic disambiguator is responsible for resolving structural ambiguities that have not been solved by either the lexical or the syntactic disambiguators.

5. Brief comments on the prototype implementation

The prototype is being developed in Visual C++ under Windows-95 using the object-oriented approach. For the sake of simplicity, only the main steps of the prototype are shown here.

5.1. The Enconverter

The enconverting process comprises the following steps:

1. Read the sentence in Portuguese. Sentence here is a piece of text between two boundary markers. As such, a unit to be anayzed can be a clause, instead of a whole sentence. Boundary markers can be conjunctions and even punctuation marks (semi-colon, colon).
2. Activate the syntactic analyzer to retrieve the syntactic role of each lexical item of the input. A message of failure is returned if there is no match to a production rule.
3. Activate the lexical analyzer to retrieve the attributes of the words.
4. Activate the semantic analyzer to retrieve the semantic role of each lexical item. Those lexical items that are not recognized (currently, the only reason is for not being defined in the lexicon) are returned with no semantic attribute.
5. Format the information provided by the former processes according to the UNL specification.

Example:

1. ***A menina comprou um livro.***

2. suj(comprou, menina)
   obj(comprou,livro)
   det(menina,a)
   det(livro,um)

3. menina.@def
   comprou.@past
   livro.@indef

4. agt(comprou, menina)
   obj(comprou, livro)

5. agt(buy.@past,girl.@def)
   obj(buy.@past,book.@indef)

5.2. The Deconverter

The enconverting process comprises the following steps:

1. Activate the lexical generator for retrieving lexical attributes. The syntactic roles corresponding to ALs are filled. Portuguese words corresponding to UWs are inflected according to the AL specification.
2. Activate the semantic generator for retrieving intrasentential relationships.
3. Activate the syntactic generator for building up the sentence. Semantic relations are retrieved from the UNL representation and syntactic roles of each UNL item are filled in. The sentence so represented is then realized in Portuguese.

Example:

1. comprar(passado)

2. suj(comprou, menina)
   obj(comprou,livro)
   det(menina,a)
   det(livro,um)

3. ***A menina comprou um livro.***

6. Conclusions

The prototype has been explored to deal with a variety of sentences from a UNL corpus for which the corresponding UNL representation is available. This prototype was implemented in a few weeks despite the complexity of the whole project. It was aimed at investigating the potentialities and limitations of the UNL formalism. For a more robust deconverter and enconverter system, other formalisms should be considered. In fact, a deconverter system for UNL-Portuguese has been currently implemented by using DeCo, the deconverter tool supplied by the IAS/UNU [3].

Acknowledgments

We are very gratefull to all other members of NILC who collaborated to the implementation of this prototype, specially to Fabiano M.C.Vieira.

References

[1] Universal Networking Language: An Electronic Language for Communication, Understanding and Collaboration. Booklet from UNL Center, IAS/UNU, Tokyo, 1996.

[2] M.G.V. Nunes, C.M. Ghiraldelo, G. Montilha, M.A.S. Turine, M.C.F. de Oliveira, R. Hasegawa, R.T. Martins and O.N. Oliveira Jr., *Desenvolvimento de um sistema de revisão gramatical automática para o português do Brasil*, II Encontro para o Processamento Computacional de Português Escrito e Falado, Curitiba, Pr, Brasil, Outubro de 1996 (in Portuguese).

[3] UNL (1997). *DeConverter Specification.* Version 1.0 (Tech. Rep. UNL-TR1997-010). UNU/IAS/UNL Center. Tokyo, Japan.