Method and Means of Development of Context-Free Grammars for Natural Language Parsers

A. GELBUKH,[1,2] G. SIDOROV,[1] S. N. GALICIA-HARO,[1] and I. A. BOLSHAKOV [1]

[1] Natural Language and Text Processing Laboratory,
Center for Computing Research, National Polytechnic Institute,
Av. Juan Dios Batiz s/n, Zacatenco 07738, Mexico City
MEXICO

[2] Department of Engineering and Computer Science,
Chung-Ang University,
221 Huksuk-Dong, DongJak-Ku, 156-756, Seoul,
KOREA

Abstract: - We discuss the requirements for the system that performs the analysis of natural language at the syntactic level. We also present the environment that allows development of context-free grammars for natural language parsers. The environment was tested for Spanish language, resulting on the development of a Spanish morphological analyzer. The environment gives the user the possibilities to develop and debug grammars of new languages. It has an option of ordering different parsing variants according to their probabilities on the basis of a specialized dictionary of government patterns.

Key-words: Context-free grammars, natural language parsing, language engineering, knowledge engineering.

1 Introduction

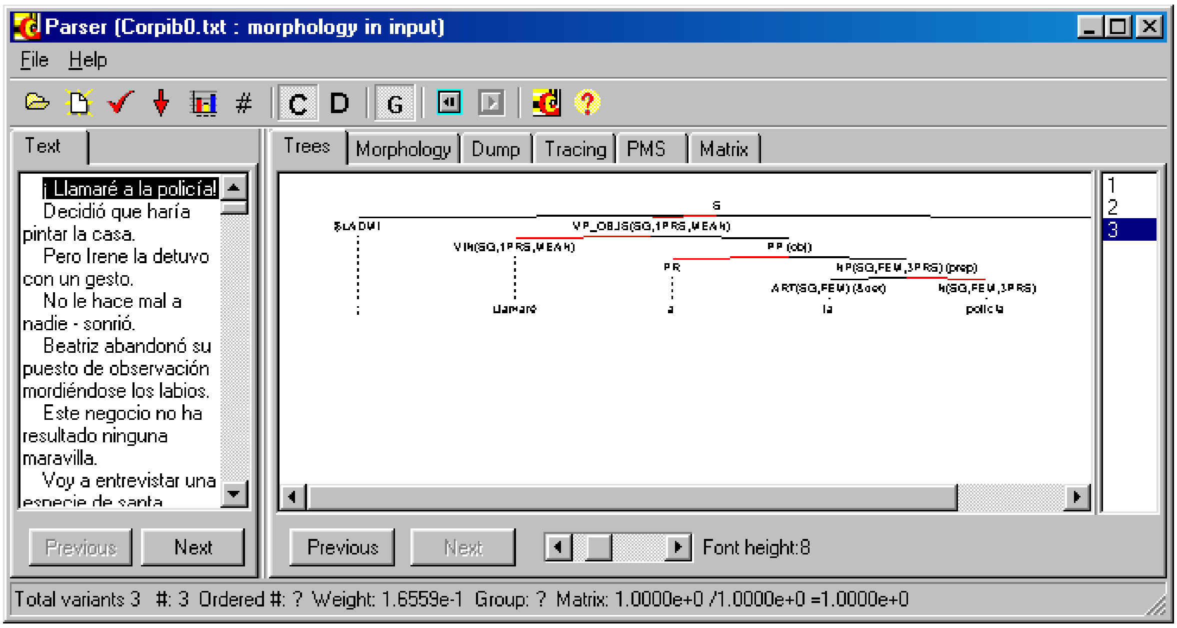
A natural language processing system usually relies on two possible sources of information: statistical information and/or large lexical resources (dictionaries). As any software system, it requires developing and implementation of algorithms. Linguistic algorithms, apart from programs, include formal grammars that can be applied by programs called parsers, for syntactic analysis of texts. Formal grammars differ from traditional grammars from textbooks in that they have strict format and can be automatically interpreted by specialized computer programs.

Large lexical resources are dictionaries that contain different kinds of information. For example, morphological dictionary contains the information about grammatical classes of words or stems, as well as other relevant information for correct declension or conjugation [Gelbukh and Sidorov, 2003]. A dictionary of subcategorization frames (government patterns) [Gelbukh *et al.*, 1998] presents information about syntagmatic relations between words that play different syntactic roles, such as direct object, indirect object, etc., usually expressed through prepositions or grammatical cases. A special type of dictionaries is thesaurus, which lists for the words their relations with other words such as homonyms, hyponyms, etc.; see, for example, [Gelbukh *et al.*, 1999].

Statistical information is obtained by analyzing another type of large textual resources, namely, corpora. A corpus is a large collection of texts with certain characteristics, such as certain genre or a balanced mixture of genres, etc. Its size can vary from several megabytes to several gigabytes. Corpora augmented by additional linguistic information, for example, morphological tags, syntactic relations, etc., usually are more useful for linguistic research.

In this paper we discuss an approach to building machine-readable grammars for syntactic analysis of natural language texts. It is based on lexical resources, though statistical information was used for its construction— for example, for construction of the dictionary of subcategorization frames.

In the rest of the paper, we describe the environment that presents a means for development of context-free grammars for natural language processing. The environment also uses some specialized dictionaries, such as morphological dictionary and dictionary of subcategorization frames.



¡ Llamaré a la policía!
Decidió que haría pintar la casa.
Pero Irene la detuvo con un gesto.
No le hace mal a nadie - sonrió.
Beatriz abandonó su puesto de observación mordiéndose los labios.
Este negocio no ha resultado ninguna maravilla.
Voy a entrevistar una especie de santa

Total variants 3 #: 3 Ordered #: ? Weight: 1.6559e-1 Group: ? Matrix: 1.0000e+0 /1.0000e+0 =1.0000e+0

Figure 1. Representation of syntactic tree in graphic form.

The paper is organized as follows. Section 2 presents the environment for developing natural language grammars. Section 3 describes the use of this environment. Finally, Section 4 draws some conclusions.

2 Description of the Environment

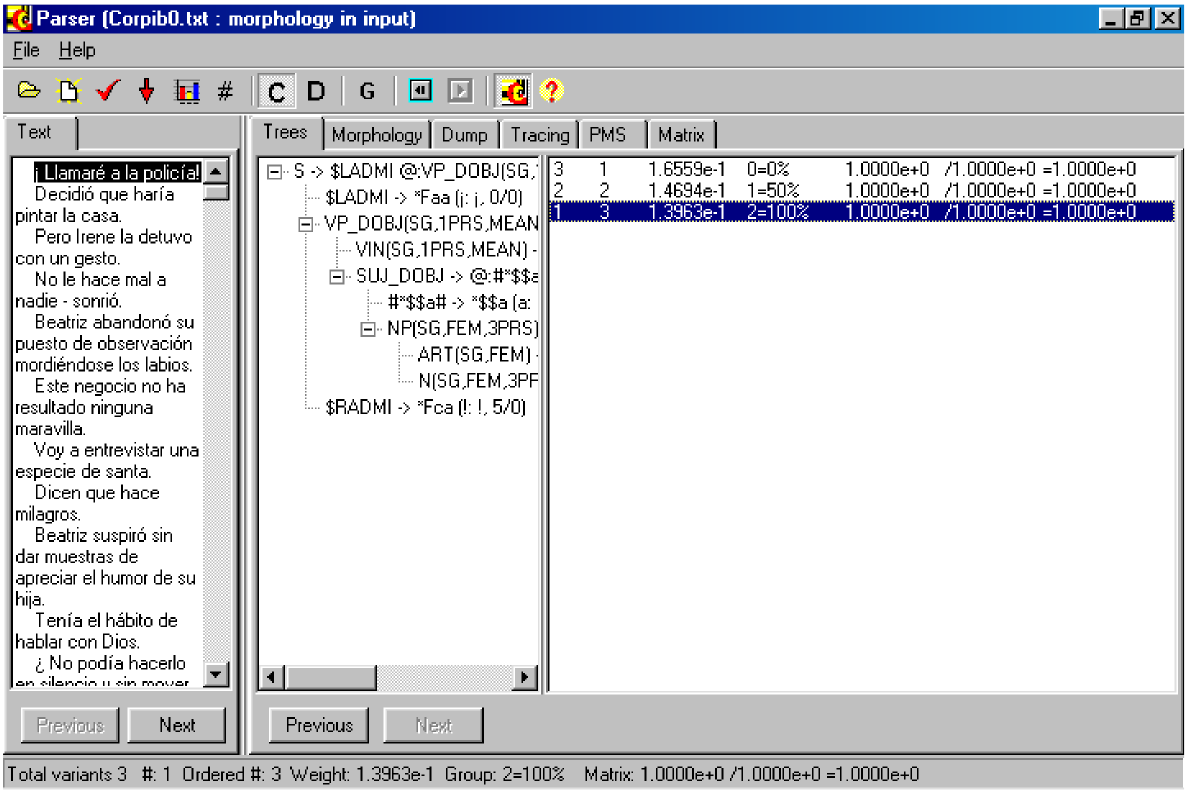
Our environment loads a context-free grammar and then applies it to parsing of sentences written in natural language. During the parsing it presents information on the steps it executes and the rules it uses, so that the developer of the grammar can easily see the effects of changes to the grammar and the work of individual rules and groups of rules.

In the environment there are possibilities to write a new text or load for the analysis already existing texts. The text can have different formats:

- Texts in free format. In this case the environment applies morphological analysis to the words of the text. In our case the morphological analyzer is for Spanish language, however, by changing the module of morphological analysis the system can process other languages.
- The sentences can have the form of <*word, lemma, grammar information*>, one tuple per line. This is the format of one of existing Spanish corpora, namely, LEXESP. The format of grammar information was borrowed from the same corpus.
- Another possibility to avoid morphological analysis is to add all necessary words in their different grammar forms to the CF grammar. Note that the words in languages other than English can have very many grammar forms, so it is not very convenient solution. Say, in Spanish verbs have 65 forms, not counting clitic forms.

The main result of the application of a grammar is the syntactic tree built for each sentence, such as the one shown in Figure 1. The left part of the figure shows the individual sentences of the text under analysis. Moving the position of the active sentence in the left-hand panel, the user can see the syntactic tree (more precisely, the variants of the syntactic tree) built by the program for the current sentence. The representation of the tree can be textual (as a list of nodes and their relationships) or graphical (as a tree drawn by lines on the screen). Obviously, graphical form is much more intuitive.

The environment allows different modes of representation of syntactic information: in form of dependencies or in form of constituents [Mel'cuk, 1988]. The difference between these formalisms is that constituents reflect the nested structure of the grammatical parts of the sentence, while dependencies focus on the relations between the head word and its dependents [Steele, 1990]. Representation using dependencies is presented in textual form in Figure 2. The conversion from the



¡ Llamaré a la policía!
Decidió que haría pintar la casa.
Pero Irene la detuvo con un gesto.
No le hace mal a nadie - sonrió.
Beatriz abandonó su puesto de observación mordiéndose los labios.
Este negocio no ha resultado ninguna maravilla.
Voy a entrevistar una especie de santa.
Dicen que hace milagros.
Beatriz suspiró sin dar muestras de apreciar el humor de su hija.
Tenía el hábito de hablar con Dios.
¿ No podía hacerlo en silencio u sin mover

```
⊟ S -> $LADMI @:VP_DOBJ(SG,
   $LADMI -> *Faa (¡: ¡, 0/0)
   ⊟ VP_DOBJ(SG,1PRS,MEAN
      VIN(SG,1PRS,MEAN) -
      ⊟ SUJ_DOBJ -> @:#*$$a
         #*$$a# -> *$$a (a:
         ⊟ NP(SG,FEM,3PRS)
            ART(SG,FEM) -
            N(SG,FEM,3PF
   $RADMI -> *Fca (!: !, 5/0)
```

```
3   1   1.6559e-1   0=0%      1.0000e+0  /1.0000e+0 =1.0000e+0
2   2   1.4694e-1   1=50%     1.0000e+0  /1.0000e+0 =1.0000e+0
1   3   1.3963e-1   2=100%    1.0000e+0  /1.0000e+0 =1.0000e+0
```

Total variants 3 #: 1 Ordered #: 3 Weight: 1.3963e-1 Group: 2=100% Matrix: 1.0000e+0 /1.0000e+0 =1.0000e+0

Figure 2. Weights of variants.

constituency output of the grammar to the dependency tree can be performed thanks to inclusion of head markers in the grammar. With this, inclusion of a constituent into a larger one can be treated as a dependency link between the correspondent heads.

Usually there are many syntactic variants that can correspond to the given sentence. For example, the sentence *I see a cat with a telescope* can be interpreted as

[*I see* [*a cat*] [*with a telescope*]]

(meaning ‘I use a telescope to see a cat’) or

[*I see* [*a cat* [*with a telescope*]]]

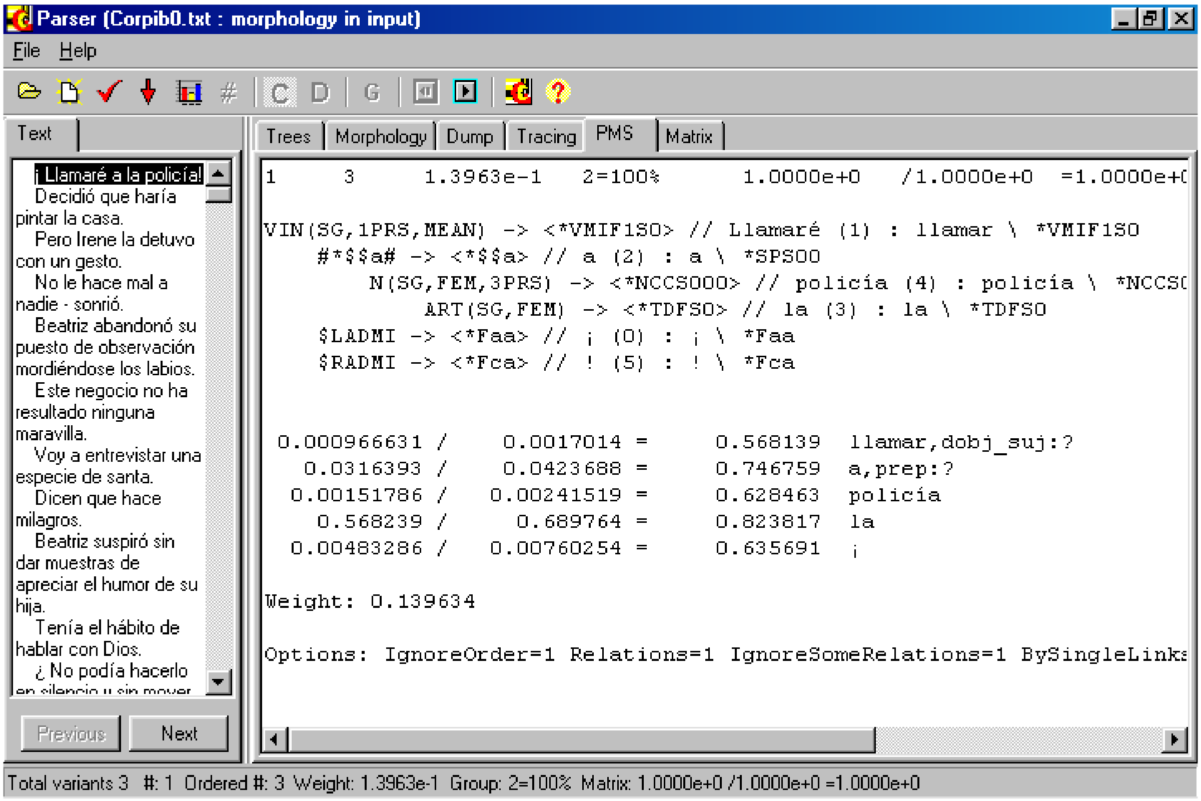
(meaning ‘I see a cat that has a telescope’). Our syntactic analyzer assigns likelihood weights to different possible variants. The variants are ordered according to these weights, so that the most probable variant is shown first. However, the user can select other variants to see the corresponding alternative trees. The menu of the variants is shown in the right-hand side of Figure 1.

The details of calculation of the likelihood weights are shown in Figure 2. They are calculated on the basis of the information about subcategorization frames present in the dictionary. An example of this information is presented in Figure 3. In this figure, one can see that different possible combinations of syntactic roles have different probabilities (learnt automatically from a large text corpus), and the resulting value is a combination of these values. More details about the method of assigning the likelihood weights through the learnt probabilities can be found in [Gelbukh, 1999].

The system provides different views that show different aspects of the analysis. Very important is the possibility to trace each step of the application of grammar. The correspondent tracing is shown in Figure 4. This view allows analyzing which rules were applied and in what order. The result of the current connected branch of the parsing tree for each rule can be seen at the right side of the *Tracing* view.

In addition to the syntactic structure, tracing, and information about subcategirzation, which explains the weights assigned to variants of parsing, the system presents various views that allow debugging of the grammar, such as *Morphology* and *Dump*. We do not describe here these additional views.

The system relies on external software modules for compilation of the grammar prior to its use. The language engineer writes a context-free grammar as a plain text file in a special format. Then it is automatically converted into a binary grammar file used by the system to process the input texts. From



```
¡ Llamaré a la policía!
   Decidió que haría
pintar la casa.
   Pero Irene la detuvo
con un gesto.
   No le hace mal a
nadie - sonrió.
   Beatriz abandonó su
puesto de observación
mordiéndose los labios.
   Este negocio no ha
resultado ninguna
maravilla.
   Voy a entrevistar una
especie de santa.
   Dicen que hace
milagros.
   Beatriz suspiró sin
dar muestras de
apreciar el humor de su
hija.
   Tenía el hábito de
hablar con Dios.
   ¿ No podía hacerlo
en silencio y sin mover
```

```
1      3      1.3963e-1    2=100%       1.0000e+0    /1.0000e+0    =1.0000e+0

VIN(SG,1PRS,MEAN) -> <*VMIF1SO> // Llamaré (1) : llamar \ *VMIF1SO
    #*$$a# -> <*$$a> // a (2) : a \ *SPS00
        N(SG,FEM,3PRS) -> <*NCCS000> // policía (4) : policía \ *NCCS0
            ART(SG,FEM) -> <*TDFSO> // la (3) : la \ *TDFSO
    $LADMI -> <*Faa> // ¡ (0) : ¡ \ *Faa
    $RADMI -> <*Fca> // ! (5) : ! \ *Fca


 0.000966631 /    0.0017014 =     0.568139  llamar,dobj_suj:?
   0.0316393 /    0.0423688 =     0.746759  a,prep:?
  0.00151786 /   0.00241519 =     0.628463  policía
    0.568239 /     0.689764 =     0.823817  la
  0.00483286 /   0.00760254 =     0.635691  ¡

Weight: 0.139634

Options: IgnoreOrder=1 Relations=1 IgnoreSomeRelations=1 BySingleLinks
```

```
Total variants 3  #: 1  Ordered #: 3  Weight: 1.3963e-1  Group: 2=100%  Matrix: 1.0000e+0 /1.0000e+0 =1.0000e+0
```

Figure 3. Subcaterization information.

the point of view of the user, this conversion is done through execution of the compilation program with certain parameters. During the execution of the program unification procedures are applied to model various aspects of grammatical concordance. Thus it is not necessary for the linguist to write all the possible rules manually, as shown in the following example.

Consider an example of Spanish grammar:

NP(nmb,gnd)
→ *[det:DETER(nmb,gnd)] @:NOM(nmb,gnd)*
→ *@:PPR [prep:PP]*
→ *[det:DETER(nmb,gnd)] @:'cual'*
→ *mod:'todo' @:NP(nmb,gnd)*

AP(nmb,gnd)
→ *@:ADJ(nmb,gnd) comp:ADJ(nmb,gnd)*
→ *@:ADJ(nmb,gnd) [',' comp:AP(nmb,gnd)]*
→ *AP(nmb,gnd) @:CONJ c_conj:ADJ(nmb,gnd)*
→ *@:ADJ(nmb,gnd) prep:PP*

The left-hand part of the rules is written before the first arrow "→". The variants of the right-hand part of a rule are given after the arrows. This notation is a shorthand for several rules with the same left-hand part.

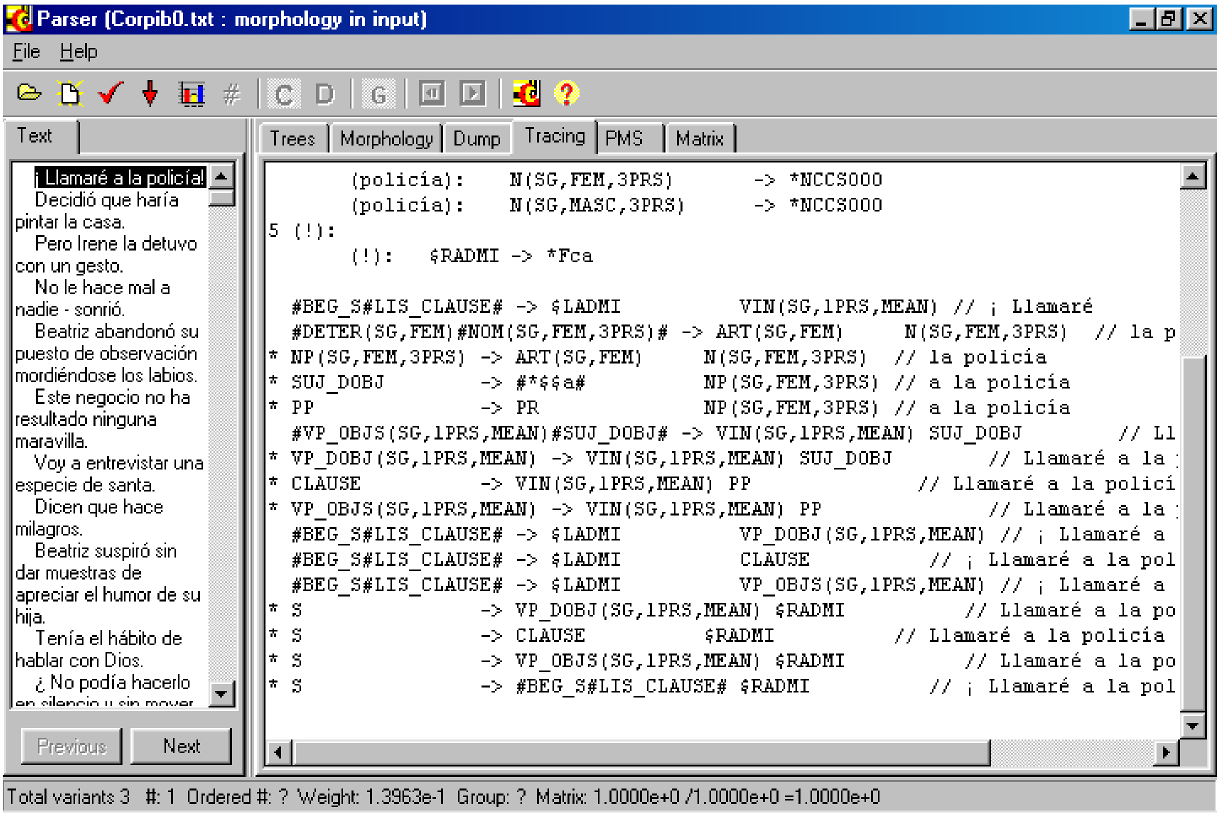
The parts of the rules in square brackets are optional; thus a rule with a construction in square brackets is a shortcut for two rules, one with the construction in question and another one without this construction.

The symbol @ marks the head of the syntactic relation. This information is not needed for the analysis of the constituency structure, but is used for conversion of the constituency tree into dependency tree. Namely, the dependency links are drawn from the head of the enclosing constituent to the heads of the immediately nested constituents.

The use of the variables *nmb* and *gnd* allows for definition of the agreement between noun and its modifier using unification techniques. Namely, a rule with variables is a shorthand for several rules, one for each combination of the specific values of the variables. For example, the rule with two varaibles, say, number and gender, stands for four individual rules with singular masculine, singular feminine, plural masculine, and plural feminine.

Some special words with grammatical function in Spanish grammar can be indicated. For example, in case of nominals, these are *todo* 'all', *cual* 'which', etc.

A real-world Spanish grammar that we developed contains only 150 lines of manually written rules. So little number of rules is possible due to unification mechanism described above. After compilation, these rules are automatically converted in ca. 10,000 standard context-free rules in the Chomsky normal form, without any



¡ Llamaré a la policía!
Decidió que haría pintar la casa.
Pero Irene la detuvo con un gesto.
No le hace mal a nadie - sonrió.
Beatriz abandonó su puesto de observación mordiéndose los labios.
Este negocio no ha resultado ninguna maravilla.
Voy a entrevistar una especie de santa.
Dicen que hace milagros.
Beatriz suspiró sin dar muestras de apreciar el humor de su hija.
Tenía el hábito de hablar con Dios.
¿ No podía hacerlo en silencio y sin mover

```
       (policía):    N(SG,FEM,3PRS)       -> *NCCS000
       (policía):    N(SG,MASC,3PRS)      -> *NCCS000
5 (!):
       (!):   $RADMI -> *Fca

  #BEG_S#LIS_CLAUSE# -> $LADMI          VIN(SG,1PRS,MEAN) // ¡ Llamaré
  #DETER(SG,FEM)#NOM(SG,FEM,3PRS)# -> ART(SG,FEM)     N(SG,FEM,3PRS)  // la p
* NP(SG,FEM,3PRS) -> ART(SG,FEM)     N(SG,FEM,3PRS)  // la policía
* SUJ_DOBJ        -> #*$$a#          NP(SG,FEM,3PRS) // a la policía
* PP              -> PR              NP(SG,FEM,3PRS) // a la policía
  #VP_OBJS(SG,1PRS,MEAN)#SUJ_DOBJ# -> VIN(SG,1PRS,MEAN) SUJ_DOBJ        // Ll
* VP_DOBJ(SG,1PRS,MEAN) -> VIN(SG,1PRS,MEAN) SUJ_DOBJ        // Llamaré a la :
* CLAUSE          -> VIN(SG,1PRS,MEAN) PP              // Llamaré a la polici
* VP_OBJS(SG,1PRS,MEAN) -> VIN(SG,1PRS,MEAN) PP                // Llamaré a la :
  #BEG_S#LIS_CLAUSE# -> $LADMI          VP_DOBJ(SG,1PRS,MEAN) // ¡ Llamaré a
  #BEG_S#LIS_CLAUSE# -> $LADMI          CLAUSE               // ¡ Llamaré a la pol
  #BEG_S#LIS_CLAUSE# -> $LADMI          VP_OBJS(SG,1PRS,MEAN) // ¡ Llamaré a
* S               -> VP_DOBJ(SG,1PRS,MEAN) $RADMI          // Llamaré a la po
* S               -> CLAUSE          $RADMI          // Llamaré a la policía
* S               -> VP_OBJS(SG,1PRS,MEAN) $RADMI          // Llamaré a la po
* S               -> #BEG_S#LIS_CLAUSE# $RADMI         // ¡ Llamaré a la pol
```

Total variants 3 #: 1 Ordered #: ? Weight: 1.3963e-1 Group: ? Matrix: 1.0000e+0 /1.0000e+0 =1.0000e+0

Figure 4. The trace record of the analysis.

shorthands and any special notation. However, the information on the heads of the constituents and the names of the grammatical relations is preserved in these rules. An example of the obtained rules can be given as follows:

N(PL,FEM) → @:N(PL,FEM) comp:N(PL,FEM)
N(PL,MASC) → @:N(PL,MASC) comp:N(PL,MASC)

As one can see, instead of variables *nmb* and *gnd*, their real values MASC, FEM, SG, and PL are used (standing for masculine, feminine, singular, plural).

3 Method of Development of the CF Grammars

The process of manual development of a large grammar using the presented system can be characterized as follows. The linguistic knowledge engineer prepares a CF grammar taking in account the unification features that will be applied to expand the manually written grammar to the complied internal form. The grammar is complied and then loaded into the environment.

A small corpus of several test sentences in the given language should be prepared. It is desirable that these sentences be representative of different syntactic phenomena of the given language.

Then the main procedure of development and debugging of the grammar is being carried out. According to the information presented by the system under the *Tracing* and *PMS* (subcategorization frames with assigned probabilities) Tabs, the grammar developer can find out the reasons for some sentences to not be parsed or to be parsed incorrectly. The corresponding changes of grammar are made and its recompilation is performed. This process is repeated iteratively until satisfactory results are obtained.

4 Modular Structure of the Weighting Process

The weighting process described above currently uses only the information about subcategorization properties of words. However, many different sources of evidence can—and should—be taken into account when determining the likelihood of a variant of the syntactic tree to be the intended one [Galicia-Haro et al., 2001]. These can be of statistical nature with different features selected for generalization, of semantic, pragmatic, or some another nature.

The system includes the possibility to plug in new modules that provide sources of evidence for syntactic disambiguation, and control the way the scores assigned to the syntactic variants by the individual sources are combined ("vote" for the final weighting). The study of the effects of different sources of evidence and different ways of their combination on the overall performance of syntactic disambiguation is one of the main applications of the system.

The additional modules being currently under development are based on:

– Collocations (word combinations): some specific words tend to be used together (e.g., *read a book*) and some not (e.g., *read a house*); this can be learnt automatically from corpora.
– Selectional preferences: some words occur together with words of certain semantic classes of the WordNet hierarchy, (e.g., *eat* FOOD). Similarly, some classes of words tend to occur together (e.g., MOVE MATERIAL-OBJECT).

5 Conclusions and Future Work

We have presented a method and the means (software environment) that allow for development of the context-free grammars for a given natural language language. The method consists in development of a CF grammar and its further debugging using the described environment. For the time being, a grammar for Spanish language has been built using the described procedure.

The future work consists in developing the possibilities of lexicalization of the grammar and development of a tool that would permit writing and debugging of a grammar at the same time in an interactive manner.

In addition, we plan to add new modules contributing to the disambiguation process. Finally, word sense disambiguation and anaphora resolution facilities are planned to be incorporated into the program, which is to improve the performance of syntactic disambiguation modules.

Acknowledgements

The work was done under partial support of Mexican Government (CONACyT, SNI, IPN, PIFI) and Korean Government (KIPA Professorship for visiting academic positions in Korea). The first author is currently on Sabbatical leave at the Chung-Ang University.

References:

[1] Galicia-Haro, S. N., A. Gelbukh, and I. A. Bolshakov. Three Mechanisms of Parser Driving for Structure Disambiguation. In: Computational Linguistics and Intelligent Text Processing (CICLing-2001), Lecture Notes in Computer Science, N 2004, Springer-Verlag, 2001, pp. 190–192.

[2] Gelbukh, A., I. Bolshakov, S. Galicia Haro. Automatic Learning of a Syntactical Government Patterns Dictionary from Web-Retrieved Texts. *Int. Conf. on Automatic Learning and Discovery*, Pittsburgh, USA, June 11–13, pp. 261–267, 1998.

[3] Gelbukh, A., G. Sidorov, A. Guzman-Arenas. Use of a weighted topic hierarchy for document classification. *Václav Matoušek et al. (Eds.). Text, Speech and Dialogue. 2nd International Workshop TSD-99*, Plzen, Czech Republic, September 13–17, 1999. Lecture Notes in Artificial Intelligence, N 1692, Springer-Verlag, pp. 130–135.

[4] Gelbukh, A. Syntactic disambiguation with weighted extended subcategorization frames. In: *Proc. PACLING-99, Pacific Association for Computational Linguistics*, University of Waterloo, Waterloo, Ontario, Canada, August 25-28, 1999, pp. 244–249.

[5] Gelbukh, A. and G. Sidorov. Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In: *Computational Linguistics and Intelligent Text Processing. Proc. CICLing-2003, 4th International Conference on Intelligent Text Processing and Computational Linguistics*, February 15–22, 2003, Mexico City. Lecture Notes in Computer Science, N 2588, Springer-Verlag, pp. 215–220.

[6] Mel'cuk, I. A. *Dependency Syntax: Theory and Practice*. State University of New York Press. 1988.

[7] Steele, J. *Meaning – Text Theory. Linguistics, Lexicography, and Implications*. James Steele, editor. University of Ottawa press. 1990.