Revista de
Informática
Teórica e Aplicada

Natural Language Syntax Learning and Syntactic Guided Spelling Repair

José Gabriel Lopes
Centro de Inteligência Artificial, UNINOVA, Portugal
Rosa Maria Viccari
II – Universidade Federal do Rio Grande do Sul

Sumário

A priori, é impossível definir uma gramática que descreva todas as frases válidas de uma língua e um dicionário que contenha todas as palavras a que um utilizador possa recorrer durante uma interação com uma Interface de Língua Natural (ILN) para um domínio de aplicação restrito. Como consequência é importante dotar uma ILN com capacidades que lhe permitam realizar diferentes tipos de aprendizagem. Neste artigo trataremos apenas as capacidades de correção de erros ortográficos e de aprendizagem de novas palavras e respectivas categorias morfológicas, novos conceitos gramaticais e novas regras sintáticas. Os processos de aprendizagem e de correção dependem do conhecimento da ILN sobre o uso da língua, das propostas de correção feitas pela interface e do grau de concordância e cooperatividade do seu utilizador. O trabalho referenciado neste artigo difere de outras abordagens existentes essencialmente porque explora a impossibilidade do conhecimento gramatical de uma interface não ser completo, propondo por isso uma metodologia de adaptação permanente.

Abstract

It is impossible to define a priori a set of rules to describe all syntactically valid sentences and a dictionary having all the words a user may write for consulting a Natural Language Interface (NLI) for a given application area. As a consequence it is desirable a NLI to have natural language learning capabilities. In this paper we focus our attention on the NLI's ability for correcting spelling, learning new words and their morphologic categories, learning new grammatical concepts and new syntactic rules. Learning and repairing depend on the NLI's knowledge about the used language, on the NLI's hypothesis proposal and on the user's agreement and cooperation. Our work differs from comparable approaches because we contend that the grammatical knowledge of a NLI can never be complete and, for this reason, it must be prone to change. Our work explores this claim.

Key Words: Artificial Intelligence, Machine Learning, Cognitive Models, Natural Language Interfaces.

1. Introduction

In recent years, Natural Language Interfaces (NLI) became more "understanding", "flexible", "robust", "forgiving" and "transportable". But they still work best with a well defined (closed) application, they still don't allow for much use of shorthand, they don't take into account past experience and memory organization, they require experienced builders for augmenting interface's knowledge coverage, they can't cope with changing themes of conversation. They keep going dinosauric, unable to adapt to new situations.

In order to overcome some of these deficiencies we started a series of experiments (whose results can be consulted in [CLV89], [Tri88], [Vic90] and [RL88]) for enabling automatic natural language learning during natural language interactions with human users. More specifically, in this paper, we focus our attention on **automatic syntactic guided spelling repair, learning new words, new grammatical concepts and new syntactic rules** ([Tri88] and [Vic90]). We are concerned with this subject because it is **impossible to describe** a grammar (set of rules) capable of defining **all syntactically valid sentences of a natural language** or a dictionary having **all its words**. For simplification, in this paper, we put aside learning of semantic and pragmatic aspects (some of these questions are treated in [RL88]). Our experience is concerned with written communication (Portuguese and English) and it is portable to other natural languages. Horn Clauses are used for representing knowledge. Learning requires updating of user models and transformation of interface knowledge through abstraction and generalization when the NLI compares its own knowledge with the knowledge represented in each of its user's models (for an extended discussion about this subject consult [CLV89]).

In the experiments we did, learning new words, new syntactic rules and new syntactic categories is based on hypotheses generation. Testing is accomplished through dialogues with users. Hypothesis generation depends crucially on the knowledge the NLI has at a given instant. Different sequences of written interactions will lead probably to different grammars. The spelling repair is based on string comparison techniques (statistical strategies) that do not directly exploit linguistic knowledge.

2. Gramatical Learning and Spelling Repair

2.1 Syntactic Description of a natural language

Let us assume that a natural language grammar is written according to the following considerations:

- let a be a syntactic category that may be rewritten as the concatenation of two other categories b and c. The rule of a context-free grammar that describes this fact may be written as

$$a \rightarrow b,\ c. \tag{2.1}$$

  In this case we say a is a *nonterminal category.*

- Let A, B, C be meta-variables denoting relations between the morpho-syntactic categories a, b, c and the word strings having these categories. The rule of a logic grammar describing fact 2.1 will be written as

$$A \rightarrow B,\ C. \tag{2.2}$$

  where A denotes a nonterminal category. We will say that A denotes a minimal hypercategory, resulting from the concatenation of the categories denoted by B and C. B and C denote maximal hypocategories of A.

- If x is a word categorized as an a, this fact can be described by the rule

$$A \rightarrow [x]. \tag{2.3}$$

  where A is a meta-variable denoting a relation among x and a. A may still convey additional information. In this case we say a is a *terminal category.*

- *empty* is a distinguished terminal category

$$empty \rightarrow [\ ] \tag{2.4}$$

- *unknowncat(I)*, with I denoting a positive integer, will denote an unknown terminal category which can not be identified with *unknowncat(J)*, for every value of J less than the value denoted by I.

- For these grammar rules, it is possible to insert additional control that is embraced by brackets { and }. So, previous rules could be rewritten as

$$A \rightarrow \{D_1\},\ B,\ \{D_2\},\ C,\ \{D_3\}. \tag{2.5}$$

$$A \rightarrow [x],\ \{D_4\}\ . \tag{2.6}$$

  where D_1, D_2, D_3 and D_4 function as control descriptions.

Any usual scheme for writing logic grammars (DCG [PW80], XG [Per83], Gapping Grammars [DA84], Restricted Logic Grammars [Sta87], and others) allows the description of natural language subsets, using a methodology analogous to the one presented before.

Writing grammar rules as we did for 2.2 and 2.3 may be justified because the meta-interpretation process is made easier — one's attention can be focussed alternatively either on B or on C, whenever it is necessary to prove that a string of words belongs to category denoted by A, using rule 2.2.

If you intend to use another kind of approach, such as one based on categorial grammars [HKM87], (at the lexical level, the category of each word defines the syntactic context where it may appear, the categories of its pre and post modifiers), you may adapt easily the reasoning method we develop in this paper.

2.2 Causes for Parsing Failure

We can identify three kinds of causes for parsing failure:

1. There is an **unknown word**. This may occur either because it is **misspelled** (it is an improperly unknown word) or because there is no procedure for finding out a similar word known by the system, whose category is coherent with the system's grammar (we say it is a **properly unknown word**).

2. There is an **unknown grammar rule**. This may occur because the sentence written by the user is syntatically correct but has a syntactic structure unknown by the system.

3. The sentence is syntactically incorrect.

At a given instant, during an analysis process of a sentence (string of words), let rule 2.2 be applied in order to categorize a sub-string of that string. If it fails, after the successful application of a rule for rewriting B, one may put the blame for the failure on the application of existing rules for rewriting C. Another hypothesis is concerned with reviewing the application of rules for rewriting the category denoted by B. A sub-string of words may have been successfully categorized as a B and still a parsing failure may have occurred (remember gardenpath sentences).

Depending on whether C denotes a nonterminal category,

$$C \rightarrow E,\ F. \tag{2.7}$$

or a terminal category,

$$C \rightarrow [w]., \tag{2.8}$$

the NLI must put forward different kinds of hypothesis and check for its validity.

When C denotes a terminal category and the application of a rule similar to 2.8 fails, because the first elements of **U** (the string of words still unparsed) can not be categorized as a C, the system can put forward three hypothesis:

1. First word of **U** is misspelled and, due to this, it is unknown.

2. First word of **U** is properly unknown.

3. Neither of these cases is confirmed by a cooperative user. A hypothesis must be set up either for changing the rewriting rules of every minimal hypercategory of the category denoted by C or for putting the blame for the failure on the application of existing rules for rewriting category denoted by B.

   When the application of a rule for identification of a nonterminal category, similar to 2.7 fails, because the first elements of **U** (the string of words still unparsed) can not be categorized as an E, we can put forward two additional hypothesis:

4. First words of **U** are known but its category does not match the category denoted or required by E. Depending on whether E denotes a non terminal or a terminal category, hypothesis put forward must be one of the 5 cases considered in this section.

5. First word of **U** and its category are both unknown.

Let us now look carefully into each of these cases.

2.3 How to solve the Failure Problem

In this paper we are going to consider an **interactive mode** of learning — once an hypothesis has been put forward, the NLI asks the user for confirmation. Depending on the answer (**yes**, **no** or **don't know**), the course of action proceeds coherently. We chose to work on this mode. An **exclusively deductive mode** of learning (which we have locally experimented) requires heavy machinery for selecting a set of minimally conservative solutions — those solutions for which the number of grammar rules changed and added is a least lower bound [RL88]. In this paper we only consider **menu driven clarification dialogues**.

Syntactic Guided Misspelled Words Correction

Consider the case 1, where the first word of a string of words still unparsed is misspelled, let $[w_1, w_2, \cdots, w_i, \cdots, w_l]$ be a string of words; let w_i be an unknown word (either because it is misspelled, or because it is properly unknown); let $[w_k, \cdots, w_{i-1}]$ $(k < i-1)$ be categorized as a B, due to application of rule 2.2. At this point of the parsing, it is expected that $[w_i, \cdots, w_j]$ $(i < j)$ will be categorized as a C (rule 2.2).

> **If the parsing process executed so far is correct, and C denotes a terminal category, then it is expected w_i to have category denoted by C.**

As w_i is not known as a word having category denoted by C, the NLI assumes first that it is misspelled and then must behave accordingly in order to find out the correct spelling for w_i [1].

If a set of words similar to w_i is detected, a clarification dialogue starts and the user will choose the menu's items that suit his/her perspective.

According to my knowledge, word w_i is misspelled.
Isn't it? (Yes / No/Don't Know)

Different conclusions may be infered as a consequence of the user's choice. If the user confirms the NLI's view, a sub-dialogue will follow, where one or more hypothesis for correcting the misspelled word are shown. When the user denies the NLI's expectation, a properly unknown word may have been found.

Learning Properly Unknown Words

Consider case 2, when the user denies the NLI's expectation and tells the system that word w_i is not misspelled. An analogous case occurs when it is not possible to find out a word similar to w_i. Now, a clarification dialogue follows in order to ascertain if w_i belongs to category denoted by C infered during the analysis process. If the NLI's hypothesis is confirmed the problem is locally solved. To the user's model is added this new syntactic fact.

Assuming a menu driven dialogue mode, when the user doesn't accept the NLI's suggestion for categorizing w_i, a clarification sub-dialogue will be necessary. The system will ask the user for choosing one of the known categories for w_i.

Choose one of the following categories for w_i

noun
adjective
verb
determiner
another category
don't know

After the choice of the user, the behaviour of the system depends on the rule currently being used for parsing and will be discussed in the next section.

Learning Unknown Syntactic Rules and Categories

We are now in the state where:

- if w_i was misspelled it has already been corrected;

[1] A lot of work done for correcting misspelled words (see [Ber87], for a logic programming approach and for an exhaustive reference list. See [Tri88] and [Vic90] , for the algorithm used in the experiments we did).

- the hypothesis of w_i having category denoted by C has been denied;
- the NLI knows that category of w_i is one of the open categories (noun, adjective, verb, determiner or other) because the user has chosen it. Let us represent this category as Cat [2].

Let Cat denote de category for w_i. We can identify two cases:
In a first case, where C denotes a terminal category,

- either we admit the existence of a hypercategory C_1 such that

$$C_1 \rightarrow Cat,\ C. \tag{2.9}$$

This hypothesis requires w_{i+1} to have category denoted by C and is easily checked. However, the grammar must evolve in order to include rule 2.9 and the transformation of every rule for minimal hypercategories of C. In the generic case we are following, rule 2.2 should be replaced by

$$A \rightarrow B;\ C_1. \tag{2.10}$$

We also must add the grammar rule 2.11 for Cat in order to take into account the fact that it may denote the empty category.

$$Cat \rightarrow empty \tag{2.11}$$

- or we take into account the hypothesis of C denoting the empty category. In this case a new rule should be added

$$C \rightarrow empty. \tag{2.12}$$

and the parsing should continue using the rule that required the application of 2.2.

- or we must backtrack and review the categorization of $[w_k \dots w_{i-1}]$ $(k < i - 1)$ as a B.

In a second case, where C denotes a nonterminal category, according to the application of rule 2.4, the parsing procedure focus its attention on the application of the rules for rewriting E and, independently of E denoting a terminal or a nonterminal category, we are again in a case similar to the one we explained for C.

[2] Another situation can occur if we allow the user to tell that he/she does not know the category of the word. This case has not been fully worked.

Rewriting of Grammar Rules

The procedure for rebuilding the user grammar can be decomposed into the following steps:

- Determine conflict category (it was C in previous explanation).
- Generate and test hypothesis for

  1. misspelled words — if this hypothesis succeeds, the misspelling is repaired and the grammar remains untouched;
  2. properly unknown words — if this hypothesis succeeds a new entry for that word is created — a new rule is added to the grammar

     $C \rightarrow [w_i]$.
  3. rewriting conflict category and rebuilding the grammar.

For rewriting conflict category and rebuilding the grammar it is necessary:

- to rename conflict category (C_1).
- For each grammar rule, where the conflict category appears in its body (the right hand side of the focussed rule), replace the name of the conflict category by its new name, forget the old grammar rule and add the new corresponding one.
- Add a new rule to the old grammar for rewriting the renamed conflict category (equivalent to case where rule 2.9 was added).
- Add new rules to the old grammar for categorizing the new word, including the case where it was decided to rewrite it as the empty category.

Depending on the model for grammar evolution adopted (see [CLV89]) it may be necessary to keep an ordered memory of all rule substitutions, and of all added rules that occurred during an interaction with the user. This memory can be used for reconstructing the grammar with which a specific user was working in his/her last interaction with the NLI.

3. Examples

This is a very simple context-free grammar for Portuguese.

$$s \rightarrow np,\ vp. \quad (3.1)$$

$$np \rightarrow det,\ np\_nucleus. \quad (3.2)$$

$$np \rightarrow pronoun. \quad (3.3)$$

$$np\_nucleus \rightarrow proper\_noun. \quad (3.4)$$

$$np\_nucleus \rightarrow noun,\ adj. \quad (3.5)$$

The rule for English corresponding to Portuguese rule 3.5 would reverse the order: *np_nucleus → adj, noun.* In some contexts, this is also the case for Portuguese. (See example in section 3.3.)

vp → vp_nucleus, pp. (3.6)

vp_nucleus → be, be_comp. (3.7)

vp_nucleus → vi. (3.8)

vp_nucleus → vt, np. (3.9)

be_comp → adj. (3.10)

be_comp → np. (3.11)

pp → prep, np. (3.12)

pp → empty. (3.13)

adj → empty. (3.14)

det → empty. (3.15)

The examples shown in the next sections were tested in an environment where an intelligent tutor for Prolog teaching interacted with its pupils, one at a time.

3.1 Repairing Misspellings

When a student chooses a menu entry for tutor consultation, a window is displayed showing the topics about which the tutor can talk about. Prolog is one of these topics. During the use of the tutor for Prolog teaching, it was observed that children wanted to develop their own data bases. They chose to work on **flowers**, kinds of **sands** and **fishes** [3].

Tópicos	Topics
flores	**flowers**
prolog	prolog
areias	sands
peixes	fishes

⇒ Quais são as **fllres** bonitas?	⇒ What are the beautiful **fllwers**?
A palavra *fllres* está mal escrita.	Word *fllwers* is misspelled.
Não é verdade? (**Sim** / Não)	Isn't it? (**Yes** / No)
Pretendia escrever *flores*.	You intended to write *flowers* .
Não é verdade? (**Sim** / Não)	Didn't you? (**Yes** /No)

[3]Pupils choices in a menu are bold. Tutor's enhancements are italic. Prompt ⇒ marks the sentences the student writes.

By the grammar rules we presented before, word **fllres** (or **fllwers**, for the English version) could be a noun. In the morpho-syntactic dictionary there is an entry for a similar[4] noun **flores** (**flowers**) distinguished by a character, an l instead of an o, in the same position.

3.2 Learning New Words

Consider a situation where the student uses a word that the tutor does not recognize because, in its dictionary, there are no similar words having the same category, as infered from the syntactic sentence context and the tutor's knowledge. Consider the example

⇒ Quais são as plantas crucíferas?	⇒ Which are the cruciferous plants?

where **crucíferas** is not known by the tutor while the other sentence words and their syntactic categories agree with known grammar rules — it is possible to parse the sentence if the tutor issues a hypothesis about the unknown word

Não conheço a palavra *crucíferas*. Mas parece-me que é um adjectivo como:	I don't recognize word *cruciferous*. But it seems to me that it is an adjective such as:
grande	big
miúda	small
bonitas	beautiful
Não é verdade? (**Sim** / Não / Não sei)	Isn't it? (**Yes** / No / Don't know)

3.3 Learning New Syntactic Structures

Let's consider the following example.

⇒ Que são aquelas bonitas flores?	⇒What are those beautiful flowers?
Na sua frase, *bonitas* é um substantivo.	In your sentence, *beautiful* is a noun.
Não é verdade? (Sim / **não** / Não sei)	In your sentence, *beautiful* is a noun.
Então qual é a categoria sintáctica da palavra *bonitas*?	Then what's the syntactic category of word *beautiful*?

[4]The concept of *similar words* is well defined in [Ber87].

escolha uma categoria	**choose one categorie**
determinante	determiner
substantivo	noun
pronome	pronoun
adjectivo	**adjective**
verbo	verb
outra	another

At this moment, when rule 3.5 is being used and *noun* is a terminal category, a new kind of noun description, *noun1*, must be added [5]

noun1 → adj, noun. **(3.16)**

and, because rule 3.5 is the only one where old noun category appears at its right side, it must be substituted by

np_nucleus → noun1, adj. **(3.17)**

Remember that the tutor does not know **adverbs**.

⇒As rosas são flores *muito* bonitas.	⇒Roses are *very* beautiful flowers.
Não conheço a palavra *muito.*	I don't know word *very.*
Qual é a sua categoria sintáctica?	Which is its syntactic category?
determinante	determiner
substantivo	noun
pronome	pronoun
adjectivo	adjective
verbo	verb
outra	**another**

This conversation gives rise to the conclusions: a new adjective type must be rewritten as an *unknowncat(1)* followed by an old adjective type; a word belonging to *unknowncat(1)* may or may not be present. As a consequence one must complete the pupil's model with rules

adj1 → unknowncat(1) , adj. **(3.18)**

unknowncat(1) → [ ]. **(3.19)**

unknowncat(1) → [muito]. **(3.20)**

substitute rules 3.17, 3.10 and 3.16 by the new rules

be_comp → adj1. **(3.21)**

np_nucleus → noun1, adj1. **(3.22)**

noun1 → adj1, noun. **(3.23)**

The name of the *unknowncat(1)* may be obtained later.

---

[5] There is no need to add rule adj → [ ] because it is already known by the tutor as 3.14.

4. Related Work

There is a considerable amount of systems with capabilities for repairing misspellings and to acquire knowledge about new words to be used in new application areas. Concerning transportable interfaces for querying data bases we may mention the systems EUFID [TB83], ASK [TT85], TEAM [GAMP87], TELI [BS88], IRUS [BB83], TQA [Dam85], HAPPY [RL88] and others. For Automatic Translation we recall system XTRA [Guo87]. For teaching see Prolog-Tutor [Vic90].

For each of these systems (except the Prolog-Tutor) their grammars are unchangeable. In the experiment we describe, we start with the thesis that grammatical knowledge can never be complete and, for this reason, an information system grammar for a natural language must be prone to change. Evolution may start at the level of a system's core grammar and will be induced by modifications carried out by the user's grammar model (see [CLV89]).

The work by [BP87] on natural language syntax learning has a different intention. It alerts for the limits of formal inductive inference. But does not count on the possibility of carrying out clarification dialogues for limiting hypothesis search space.

5. Concluding Remarks

Whenever a NLI is called by an user, the user's model is interpreted and modifies, for some specific aspects, the tutor's knowledge about the world. During an interaction a tutor must learn with a user and transform the user's model. This means that a tutor should not take for granted everything a student tells it. Only when there is a considerable amount of information, related with changes of tutor's knowledge, induced by different users, only then should it be possible the generalization of the NLI's own grammar knowledge (see [CLV89]).

It should be noticed that learning of new rules of a natural language grammar may depend on the order of sentence presentation to the NLI. The procedure for generalization of a NLI's grammar must take this into account.

We adopted the strategy that rules are normally modified by prefixing. However, if this is the only strategy, it will lead to learning incorrect rules. This would be the case with the following example.

$\Rightarrow$Eu gosto muito de flores. $\Rightarrow$I like flowers very much.

where the *unknowncat (1)* muito would be prefixed to a preposition, giving rise to the rule

prep1 $\rightarrow$ *unknowncat (1), prep.* **(5.1)**

and requiring the substitution of rule 3.12 by *pp → prep1, np.* **(5.2)**

Using the same kind of reasoning, there is no apparent reason why shouldn't an adverb (*unknowncat (1)*) prefix the governing category of *prep*, giving rise to a new rule [6]

pp1 → unknowncat (1), pp **(5.3)**

and to the modification of rule 3.6

vp → vp_nucleus, pp1. **(5.4)**

However, for the example above this wouldn't lead to correct learning. Correct learning also requires the use of a *suffixing strategy*, and exploration of all the possibilities for suffix and prefix binding [7]. For checking all these possibilities there must be generation of alternatives to the initial sentence. Word meaning representation and pragmatic use of words are aspects that must also be taken into account (This was the main concern of [RL88]).

The method we adopted is limited. It is based only on meta-interpretation of syntactic rules. However, for the purpose of hypothesis generation, the strategies (suffixing and prefixing) seem rather fertile.

In the first experiment with pupils, the Prolog-Tutor could learn new words and correct misspelled ones. However the underlying program could not cope with syntactic rule learning. A second phase, for experimenting the ideas we have organized in this paper, started in January at two different places, at LNEC and at UNINOVA. This phase finished by the end of July, 1988. The Prolog-Tutor is now learning new syntactic categories and rules. Information about morphemic features is also included. Current implementation also learns how to change grammar rules in order to prevent inconvenient parses due to incomplete information. For example, rule 3.1 has to be changed in order to include agreement information.

Acknowledgements

We wish to thank Luís Moniz Pereira, Helder Coelho, Ernesto Costa, Pedro Barahona, Miguel Calejo and Gabriel David, who read and commented earlier versions of this paper. The work reported was supported by JNICT, under contract No.87439, INIC, under project SAPIA, FCT/UNL, Gabinete de Filosofia do Conhecimento and CAPES.

[6] In Portuguese, there is evidence that rule 5.1 is meaningful for situations such as **muito a norte** (far north?), **muito de noite** (well inside the night?), etc.

[7] A suffixing strategy would right-bind the *adverb* to one of the rightmost daughters of *vp_nucleus*, i.e. *vi*, because verb 'gostar' (like) is intransitive.

References

[BB83] M. Bates and M. Bobrow. A transportable natural language interface for information retrieval. In *6th International ACM SIGIR Conference*, Washington D.C., 1983.

[Ber87] H.L. Berghel. A logic framework for the correction of spelling errors in electronic documents. *Information Processing and Management*, 23(5):477–494, 1987.

[BP87] R. C. Berwick and S. Pilato. Learning syntax by automatic induction. *Machine Learning*, 2:9–38, 1987.

[BS88] B. W. Ballard and D. E. Stumberger. Semantic acquisition in teli: a transportable user-customized natural language processor. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 20–29, Association for Computational Linguistics, New-York, June 1988.

[CLV89] Helder Coelho, Gabriel Lopes, and Rosa Viccari. Modelos para a concepção de gramáticas evolutivas. In *Actas do 6º Simposium Brasileiro de Inteligência Artificial*, Rio de Janeiro, Brasil, Novembro 1989.

[DA84] V. Dahl and H Abramson. *On Gapping Grammars.* Technical Report 84-2, Department of Computer Science, The University of British Columbia, 1984.

[Dam85] F. Damerau. Problems and some solutions in customization of natural language database front ends. *ACM Trans. on Office Information Systems*, 3(2):165–184, 1985.

[GAMP87] B. J. Grosz, D. E. Appelt, P. A. Martin, and F. C. Pereira. Team: an experiment in the design of transportable natural language interfaces. *Artificial Intelligence*, 32:173–243, 1987.

[Guo87] C.-M. Guo. Interactive vocabulary acquisition in xtra. In *Proceedings of IJCAI'87*, pages 715–717, Milan, August 23-28 1987.

[HKM87] W. Haddock, E. Klein, and G. Morril. *Categorial Grammar, Unification Grammar and Parsing.* Volume 1 of *Papers on Cognitive Science*, Centre for Cognitive Science, University of Edinburgh, 1987.

[Per83] F.C.N. Pereira. *Logic for Natural Language Analysis.* Technical Note 275, SRI International, Stanford, 1983.

[PW80] F.C.N. Pereira and D.H.D. Warren. Definite clause grammars for language analysis. *Artificial Intelligence*, 13:231–271, 1980.

[RL88] P. Rosado and G. Lopes. Interfaces de língua natural com capacidade para aprenderem novos vocábulos, seu significado e para se adaptarem a novos utilizadores: uma experiência. In *Actas do 5º Simpósio Brasileiro de Inteligência Artificial*, Natal, Brasil, Novembro 1988. in Portuguese.

[Sta87] E. P. Stabler. Restricting logic grammars with government-binding theory. *Computational Linguistics*, 13(1-2):1–10, 1987.

[TB83] M. Templeton and J. Burger. Problems in natural language interface to dbms with examples from eufid. In Association of Computational Linguistics, editor, *Proceedings of the 1st Conference on Applied Natural Language Processing*, pages 3–16, 1983.

[Tri88] P. Trigo. *Study of a System that Adapts to the User Linguistic Idiosyncrasies.* Technical Report, Faculdade de Ciências, Universidade de Lisboa, Lisboa, 1988. in Portuguese.

[TT85] B. H. Thompson and F. B. Thompson. Ask is transportable in half a dozen ways. *ACM Trans. on Office Information Systems*, 3(2):17–24, 1985.

[Vic90] R. M. Viccari. *An Intelligent Tutor for Logic Programming: Idealization, Project and Development.* PhD thesis, Faculdade de Engenharia, Universidade de Coimbra, Coimbra, 1990. submitted,PhD Thesis, in Portuguese.

Curriculum vitae (José Gabriel Lopes)

Doutor em Engenharia de Computadores pela Universidade Técnica de Lisboa. Pesquisador da Universidade Nova de Lisboa - Unl/UniNova. Área de interesse: Processamento de Língua Natural.

Endereço para contato:
Universidade Nova de Lisboa
Centro de Inteligência Artificial
Quinta da Torre
UNINOVA, 2825, Monte da Caparica
Portugal
E-mail: gp@fct.unl.pt

Curriculum vitae (Rosa Maria Viccari)

Doutora em Engenharia Eletrotécnica e Computadores pela Universidade de Coimbra. Professora do Instituto de Informática da Universidade Federal do Rio Grande do Sul. Área de interesse: Aprendizagem Simbólica Automática.

Endereço para contato:
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Caixa Postal 1501
90001 - Porto Alegre -RS
Brasil
E-mail: rosa@sbu.ufrgs.anrs.br

Artigo submetido em março 1990. Aceito na forma final em agosto 1990. Formatado para o estilo da Revista por Raul F. Weber a partir do texto fornecido em LaTeX pelos autores via e-mail.