



MORPHO-SYNTACTIC TOOLS FOR SPEECH PROCESSING

D.Cericola, M.Danieli, M.J.Mollo, D.Voltolini

Ing. C. Olivetti & C. S.p.A.
D.O.R. Speech & Language Laboratory
C.so Svizzera 185, 10149 Torino Italy

ABSTRACT

We describe a morpho-syntactic analyzer for the Italian language. This system integrates a lexical data-base handling about 2,000,000 forms and a probabilistic syntactic parser. It is intended to provide linguistic information for the tasks of speech recognition and text-to-speech synthesis. A description of the two components, experimental results and performances of the whole system are given.

0. INTRODUCTION

A morpho-syntactic analyzer for the Italian language has been developed to cope with most of the problems of linguistic processing in both speech recognition and text-to-speech synthesis. It is possible to think of the morpho-syntactic analyzer as a black box that can analyze different kinds of input and provide various information as its output. *Fig. 1* describes

this scheme. The input to the whole system are words, either in a sequence, like a text, or representing the lattice of different hypotheses corresponding to a word sequence. Each word can be written in its graphemic form or in its CPA [1] phonetic transcription. The first step of the analysis provides the full information lattice: it integrates the word graphemic / phonetic information expanding upon the lexical ambiguity and adding to each hypothesis all the information available in the lexicon. The final result is a lattice in which every word is replaced by 1 or more quadruplets

{graphemic form, phonetic transcription, grammatical tag, lemma}.

The second step, the syntactic analysis, is always performed on the full information lattice provided in this way. At the end of the analysis, all the possible syntactic interpretations are ready for further processing and moreover the system provides three different outputs:

- 1) the most likely sequence of quadruplets in the lattice;
- 2) the most likely syntactic interpretation for the chosen sequence [2];
- 3) a copy of the chosen text with prosodic marks inserted in it.

Applications of the morpho-syntactic analyzer are:

- 1) linguistic filtering of word hypothesis lattice provided by a recognizer (Speech, OCR,...) to increase the recognition quality;
- 2) text pre-processing for a text-to-speech system in order to provide the text with stresses and prosodic marks correlated to the syntactic structure of the sentence;
- 3) automatic text grammatical labelling to provide measures used by statistical models of the language.

Section 1 will describe the morphologic lexicon, section 2 is devoted to illustrating the syntactic parser and its knowledge base; section

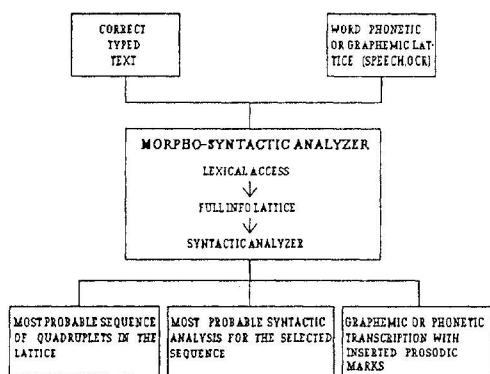


Fig. 1. Input/Output block diagram of the morpho-syntactic analyzer

3 will give some evaluations and measures of the performances of the whole system.

1. MORPHOLOGIC LEXICON

This work aimed at building a general and complete Italian lexicon. It can analyze and/or generate a big set of Italian words, providing their graphemic and phonetic representations, as well as their syntactic category.

With respect to the straightforward solution of storing the information appearing in a conventional dictionary, the system solves the most important problem of handling the inflections and this is made without an explosion of the dictionary size.

The goal is to manage a data base which can correctly provide information about Italian words, without any restriction of semantic domain. This means that the lexicon should know all the words. The information that the lexicon has to provide is determined by the applications it was designed for. Grammatical categorization and phonetic transcription are both necessary to speech processing.

Since morphology plays an important role in Italian, the lexicon model is morphologic. Morphology is divided into inflectional, derivational and alterative. The first deals with word suffixes, while the others with prefixes and infixes. Roughly speaking, suffixes bear grammatical information of tense, gender, number and person concerning verbs, nouns, pronouns and adjectives (BUON-O m.s., BUON-A f.s., BUON-I m.p., BUON-E f.p. [good]; AMA-NO 3th pl. person pres. indicative [they love]).

Other affixes (namely prefixes and infixes) bear information about semantic features of the word and about its main grammatical class. For example:

MAN-O n. [hand],
MAN-IN-A n. [little hand],
MAN-EGG-IARE v. [to handle],
MAN-EGG-EVOL-E adj. [easily-handled]

Their behaviours are difficult to account for. They can either modify a root or not, depending only on semantic relations between the root and the affix itself. If the modification occurs, they could modify both syntactic aspects and the inflectional morphology of the root (CORRERE [to run], intransitive, aux. to be; PER-CORRERE [to run along], transitive aux. to have)(5). Due to these considerations, our morphologic lexicon is mainly based on inflectional morphol-

ogy, since derivational morphology is present only in the management of superlatives and adverbs derived from adjectives. The other types of derived words are considered independent lemmata.

The lexicon is economic and productive. Moreover it is completely general, since it can manage all the regular and irregular inflections and can correctly analyze the clitic pronoun attachment to verbs both for graphemic and phonetic representation of words (2). From the grammatical point of view, we considered 9 main categories (more or less the classical parts of speech). Each root is associated to a main category and a set of syntactic features. In this way, we have split main categories into more specific ones that allow a subtler linguistic analysis and increase the disambiguation power of the syntactic parser. There are 8 different features for verbs, that produce 202 classes, 4 features for adjectives with 15 classes, and so on. The inflectional model optimizes the management of the grammatical knowledge. The system stores the grammatical tags corresponding to 500 suffixes only and this allows to handle a 20 thousand times bigger data-base.

At present, the system contains about 53,000 roots (28,000 noun, 11,000 adjectives, 8,000 verbs, 4,500 adverbs, 1,500 other), corresponding to a full labelled lexicon of about 1.7 millions inflected words. Moreover, the system can both recognize and generate cardinal and ordinal numbers between 0 and 999,999. Proper names obviously are not included in the lexicon, therefore the system deals with them by a set of heuristic rules.

Lexical access can start either from the word graphemic representation or from its phonetic transcription. It has been entirely designed making use of a relational data-base paradigm and implemented both on PC and on a mini computer.

2. SYNTACTIC PARSING

When syntactic parsers have been used for real natural language applications, the problem of their robustness has arisen (1). Real sentences are often much more complicated than those found in linguistic texts (7). "Corruption" of the input may depend both on lexical, syntactical and semantical ambiguity and on degradation occurring, for example, in the output of an acoustic front-end (3).

The parser we describe has been implemented with the aim of achieving robustness and

flexibility. When it fails to find a complete sentence parse, it can yield two or more subtrees providing anyhow useful information to the following stages.

The algorithm is a chart parser (8): we list here the features that characterize our implementation. The parsing strategy is completely bottom-up; a table of reachability from leaf to constituent is used to avoid developing impossible hypotheses; the insertion of complete edges in the chart is ruled by a packing mechanism (6). Since all the edges have an associated score, depending on acoustic evidence (if it is present), and probabilities of bigram categories and syntactic trees, a beam-search pruning mechanism

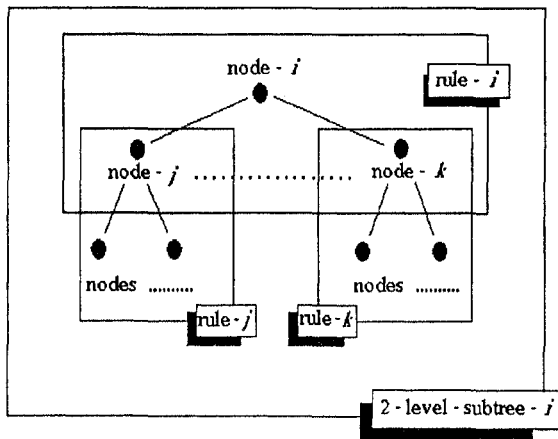


Fig. 2. The objects involved in the computation of syntactic probabilities

is used to process only a predetermined number of best-scored edges.

At the end of the analysis a decision procedure searches for the best scored sequence of quadruplets in the lattice. First the minimum number n of sub-trees needed to analyze the lattice is computed. Then the search take into account all the sequences that are covered by the grammar with n or $n + 1$ subtrees, choosing the one that has the best score, considering in the score a probabilistic weight depending on the number of sub-trees of the particular sequence.

The grammar is a probabilistic augmented context-free grammar for Italian; each rule has a typical rewrite rule part and a complex feature structure part of the form

```

(<phrase-cat>
 (<pattern-1>,...,<pattern-n>)
 <list-of-conditions>
 (<instr-1>,...,<instr-m>)
 <pros-mark>)

```

Both <phrase-cat> and <pattern- i >'s are phrasal categories (e.g. S, NP, Nbar...). These constitute the context-free part of the rule. The augmentations are the rest.

In the list of conditions there are the names of functions embodying the syntactic constraints which must be satisfied to apply the rule. They are checks such as controls about number agreement, presence of an embedded determiner, and so on.

The <instr- i > list contains the instructions for assembling the new constituent that rule the feature inheritance.

<Pros-mark> is the declaration of an optional prosodic mark that the parser has to insert in the new constituent (4).

The grammar provides a probability associated to each syntactic tree. We assume that the tree probability can be approximated by the product of the conditional probabilities of its nodes. The probability associated to a node is

$$P(2\text{-level-subtree} \mid \text{rule})$$

where "2-level-subtree" and "rule" are explained in Fig. 2.

A list of idioms is available. It contains particular locutions, for instance compound nouns, whose rules are particularly idiosyncratic in Italian: the algorithm treats them like leaves of the syntactic analysis.

The grammar has 486 rules. The conditions that express syntactic and lexical constraints are 127. The list of idioms contains 378 expressions. The probabilities for the rules have been computed over a corpus of about 1,300 sentences. We have measured the coverage of the grammar on these sentences.

We say that a sequence of words is "covered by the grammar in n sub-trees" when it is parsed with i sub-trees, where $i \leq n$. We call "coverage-index i " the ratio between the dimension of the set of sequences covered by the grammar in i sub-trees and the total number of analyzed sentences. The grammar has coverage-index 1 of about 76.6%, while coverage-index 2 is about 88.3%. Due to the particular decision procedure of the parser, the coverage-index 2 is the relevant one to understand how large is the portion of natural language the parser is able to cope with.

3. PERFORMANCES

The system has been tested for determining:

- 1) lexical coverage
- 2) performance in isolated word recognition
- 3) performance in inserting prosodic marks in a error free typed text.

We tested the coverage degree of the lexicon on a corpus of 96,045 words provided by EEC, new for the analyzer, and the 96.72% of words were in the lexicon (including proper names, but not acronyms like "EEC").

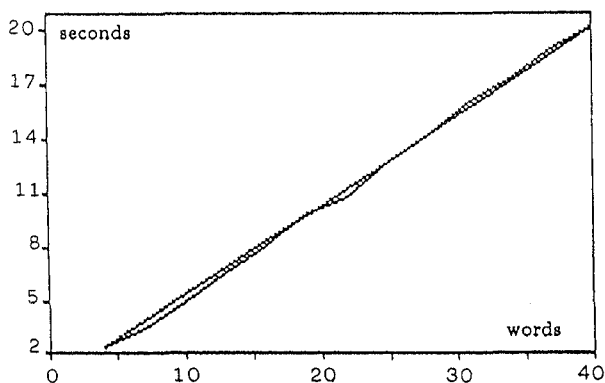


Fig. 3. Effect of sentence length on the parser processing time for Tomita's test; a straight line is also reported for comparison

For the recognition post-processing task, we considered real lattices provided by an acoustic front-end, in which the uttered word was always among the candidates. There were 3,446 words uttered by 4 different speakers; the average number of candidates in the full information lattice was 4.39 and the 17.61% of the times the correct word was not in first position, while after the linguistic analysis we got 4.21% [3].

We measured the labelling capability of the system on a corpus of more than 6,994 words manually tagged. The 0.39% of words were not tagged by the analyzer and the 3.79% were labelled with an incorrect tag.

On a text, new for both the lexicon and the parser, of 2000 words, the system put 90 correct marks, 91.84% on the total of expected marks, while it did not put 8 of them (8.16%) and inserted 9 spurious marks: it means that it mistook the 9.09% of the marks it put.

Efficiency was measured referring to our implementation of the lexical access and the syntactic parsing, made on a SUN WORKSTATION 3/260 [4] respectively in C-language and LISP. The lexical access analyzed the 96,045 words of the coverage test in 945 seconds with an average

access time of 9.8 milliseconds.

To control the growth of parsing time with respect to the length of the sentence, we repeated the test made by Tomita [5]. Fig. 3 represents the actual growth function compared with the linear growth.

REMARKS

[1] CPA is the Computer Phonetic Alphabet, developed for all European languages under the 860 ESPRIT project.

[2] These likelihoods are related to the scoring mechanism of the syntactic analyzer

[3] This figure includes "semantic" errors, i.e. those errors in which the recognized word has the same grammatical tag of the correct word. These errors cannot be recovered by a pure grammatical approach.

[4] SUN WORKSTATION is a registered trademark of Sun Microsystem, Inc.

[5] For details, see chapter 6 in (6), referring to "sample set II": paragraphs 6.1.3, 6.2.1.

REFERENCES

- (1) Carbonell, J. & Hayes, P., *Recovery Strategy for Parsing Extragrammatical Language*, *American Journal of Computational Linguistics*, 1983, pp. 123-146
- (2) Delogu, C., *The Morphological Lexicon of a Speech Recognition System for Italian*, *Rivista di Linguistica*, vol 1, n. 1, 1989, pp. 95-114
- (3) Lytinen, S.L., *Integrating Syntax and Semantics*, *Proceedings of Theoretical and Methodological Issues in Machine Translation for Natural Languages*, Hamilton, 1985
- (4) Quazza, S. et al., *Syntactic Pre-processing for High Quality Text-to-Speech*, in this issue.
- (5) Russo, M., *A Generative Grammar Approach for the Morphologic and Morphosyntactic Analysis of Italian*, *Proceedings of the 3-rd European Chapter of the ACL*, Copenhagen, Denmark, April 1987, pp. 32-37.
- (6) Tomita, M., *Efficient Parsing for Natural Language. A Fast Algorithm for Practical Systems*, Kluwer Academic Publishers, Boston, Mass. 1986
- (7) Tomita, M., "Linguistic" sentences and "real" sentences, *Proceedings of Coling 88*, Budapest, August 1988, pp. 453
- (8) Winograd, T., *Language as a Cognitive Process*, Addison-Wesley, Reading, Mass. 1983