HOW TO INTEGRATE LINGUISTIC INFORMATION IN *FILES* AND GENERATE FEEDBACK FOR GRAMMAR ERRORS

Rodolfo Delmonte, Luminita Chiran, Ciprian Bacalu
Dipartimento di Scienze del Linguaggio
Ca' Garzoni-Moro, San Marco 3417
Università "Ca Foscari"
30124 - VENEZIA
Tel. 39-41-2578464/52/19 - Fax. 39-41-5287683
E-mail: delmont@unive.it - website: byron.cgm.unive.it

Abstract

We present three applications which share some of their linguistic processor. The first application "***FILES***" – Fully Integrated Linguistic Environment for Syntactic and Functional Annotation - is a fully integrated linguistic environment for syntactic and functional annotation of corpora currently being used for the Italian Treebank. The second application is a shallow parser – the same used in ***FILES*** – which has been endowed with a feedback module in order to inform students about their grammatical mistakes, if any, in German. Finally an LFG-based multilingual parser simulating parsing strategies with ambiguous sentences. We shall present the three applications in that sequence.

1. *FILES*

FILES has been used to annotate a number of corpora of Italian within the National Project currently still work in progress. Input to ***FILES*** is the output of our linguistic modules for the automatic analysis of Italian, a tokenizer, a morphological analyser, a tagger equipped with a statistic and syntactic disambiguator and finally a shallow parser. All these separate modules contribute part of the input for the system which is then used by human annotators to operate at syntactic level on constituent structure, or at function level on head-features functional representation. We don't have here space to describe the linguistic processors – but see [8, 9, 10, 11, 12]. As to tag disambiguation, this is carried out in a semi-automatic manner by the human annotator, on the basis of the automatic redundant morphological tagger. The disambiguator takes each token and in case of ambiguity it alerts the annotator to decide which is the tag to choose: the annotator is presented with the best candidate computed on the basis both of syntactic and statistical information. Low level representations are integrated in a relational database and shown in the FILES environment which is an intelligent browser allowing the annotation to operate changes and create XML output automatically for each file. Here below is a snapshot of the six relational databases where all previously analysed linguistic material has been inputted. It contains tokens, lemmata, POS tagging, empty categories, sentences containing each token, tokens regarded as heads as separated from tokens regarded as features and verb subacategorization list.

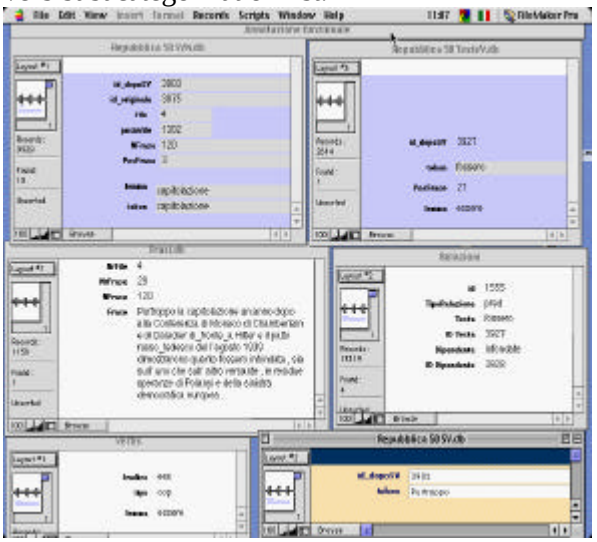

Fig.1 Relational databases to be used as input for the Syntactic and Functional Annotation

An interesting part of the browser is the availability of subcategorization frames for verbs: these are expressed in a compact format which are intended to help the annotator in the most difficult task, i.e. that of deciding whether a given constituent head must be interpreted as either an argument or an adjunct; and in case it is an argument, whether it should be interpreted as predicative or "open" in LFG terms, or else as non-predicative or "close". The list of subcategorization frames contains 17,000 entries. Of course the annotator can add new entries either as new lexical items or simply as new subcategorizations frames, which are encoded in the current list. Notable features of the browser are the subdivision into two separate columns of

verbal heads from non verbal ones, whereas the actual sentence highlights all heads verbal and non verbal in bold. On the righthand side there is a scrollable list of relations and the possibility to move from one sentence to another at will. Finally the XML button to translate the contents of each or any number of sentences into xml format.

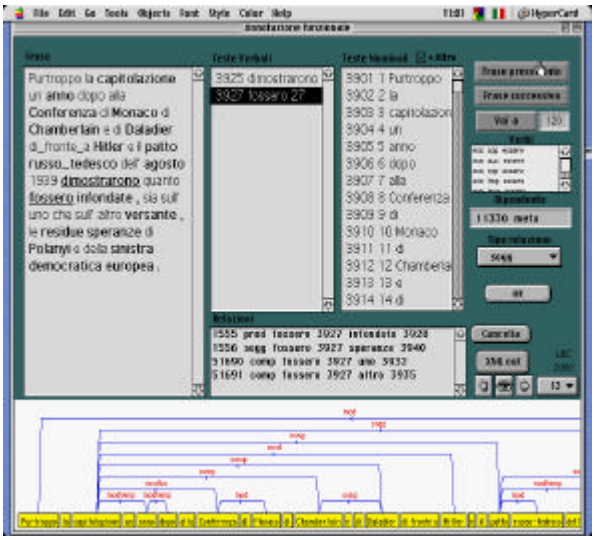

Fig.2 Browser for Functional Annotation with Structural representation

2. *GRAMMCHECK*

The second application is a Grammar Checker for Italian students of German and English. The one for students of English is based on GETARUNS and uses a highly sophisticated grammar which is however a completely separated system from the one presented here and requires a lot more space for its presentation – see [13, 14]. It is available under Internet and will be shown as such.

The one for students of German on the contrary, is based on the shallow parser of Italian used to produce the syntactic constituency for the National Treebank. The output of the parser is a bracketing of the input tagged word sequence which is then passed to the higher functional processor. This is an LFG-based c-structure to f-structure mapping algorithm which has three tasks: the first one is to compute features from heads; the second one is to compute agreement. The third task is to impose LFG's grammaticality principles: those of Coherence and Consistency, i.e. number and type of arguments are constrained by the lexical form of the governing predicate.

The parser is an RTN which has been endowed with a grammar and a lexicon of German of about 8K entries. The grammar is written in the usual arc/transition nodes formalism, well-known in ATNs. However, the aim of the RTN is that of producing a structured output both for wellformed and illformed grammatical sentences of German. To this end, we allowed the grammar to keep part of the rules of Italian at the appropriate structural level, though. Grammar checking is not accomplished at the constituent structure building level, but at the f-structure level.

2.1 THE SHALLOW PARSER

The task of the Shallow Parser is that of creating syntactic structures which are eligible for Grammatical Function assignment. This task is made simpler given the fact that the disambiguator will associate a net/constituency label to each disambiguated tag. Parsing can then be defined as a Bottom-Up collection of constituents which contain either the same label, or which may be contained in/be member of the same net/higher constituent. No attachment is performed in order to avoid being committed to structural decisions which might then reveal themselves to be wrong. We prefer to perform some readjustment operations after structures have been built rather than introducing errors from the start. Readjustment operations are in line with LFG theoretical framework which assumes that f-structures may be recursively constituted by subsidiary f-structures, i.e. by complements or adjuncts of a governing predicate. So the basic task of the shallow parser is that of building shallow structures for each safely recognizable constituent and then pass this information to the following modules.

2.2 Syntactic Readjustment Rules

Syntactic structure is derived from shallow structures by a restricted and simple set of rewriting operations which are of two categories: deletions, and restructuring. Here are some examples of both:

a. Deletion

Delete structural labels internally with the same constituent label that appears at the beginning as in Noun Phrases, whenever a determiner is taken in front of the head noun;

b. Restructuring

As explained above, we want to follow a policy of noncommittal as to attachment of constituents: nonetheless, there are a number of restructuring operations which can be safely executed in order to simplify the output without running the risk of

taking decisions which shall have later to be modified.
Restructuring is executed taking advantage of agreement information which in languages like Italian or German, i.e. in morphologically rich languages, can be fruitfully used to that aim. In particular, predicative constituents may belong to different levels of attachment from the adjacent one. More Restructuring is done at sentence level, in case the current sentence is a coordinate or subordinate sentence.

3 FROM C-STRUCTURE TO F-STRUCTURE

Before working at the Functional level we collected 2500 grammatical mistakes taken from real student final tests. We decided to keep trace of the following typical grammatical mistakes:
- Lack of Agreement NP internally;
- Wrong position of Argument Clitic;
- Wrong Subject-Verb Agreement;
- Wrong position of finite Verbs both in Main, Subordinate and Dependent clauses;
- Wrong case assignment.

Example 1. Heute ***willst*** ich ***mich*** eine bunte Krawatte umbinden.

```
cp-[
savv-[avv-[heute]],
vsec-[vsupp-[willst],
      fvsec-[sogg2-[sn-[pers-[ich]]],
                  ogg-[sn-[clitdat-[mich]]],
            ogg1-[snsempl-[art-[eine],ag-[bunte],
                  n-[krawatte]]],
            ibar2-[vit-[umbinden]]]
                  ], punto-[.]]
```

The parser will issue two error messages:
The first one is relative to Case assignment, “mich” is in the accusative while dative is required. The second one is relative to Subject-Verb agreement, “willst” is second person singular while the subject “ich” is first person singular.
As to the use of f-structure for grammar checking the implementation we made in GETARUN – a complete system for text understanding, is a case where parsing strategies are used.
This is a web-based multilingual parser which is based mainly on LFG theory and partly on Chomskian theories, incorporating a number of Parsing Strategies which allow the student to parse ambiguous sentences using the appropriate strategy in order to obtain an adequate grammatical output. The underlying idea was that of stimulating the students to ascertain and test by themselves linguistic hypotheses with a given linguistically motivated system architecture. The parser builds c-structure and f-structure and computer anaphoric binding at sentence level; it also has provision for quantifier raising and temporal local interpretation. Predicates are provided for all lexical categories, noun, verb, adjective and adverb and their description is a lexical form in the sense of LFG. It is composed both of functional and semantic specifications for each argument of the predicate: semantic selection is operated by means both of thematic role and inherent semantic features or selectional restrictions. Moreover, in order to select adjuncts appropriately at each level of constituency, semantic classes are added to more traditional syntactic ones like transitive, unaccusative, reflexive and so on. Semantic classes are of two kinds: the first class is related to extensionality vs intensionality, and is used to build discourse relations mainly; the second class is meant to capture aspectual restrictions which decide the appropriateness and adequacy of adjuncts, so that inappropriate ones are attached at a higher level.

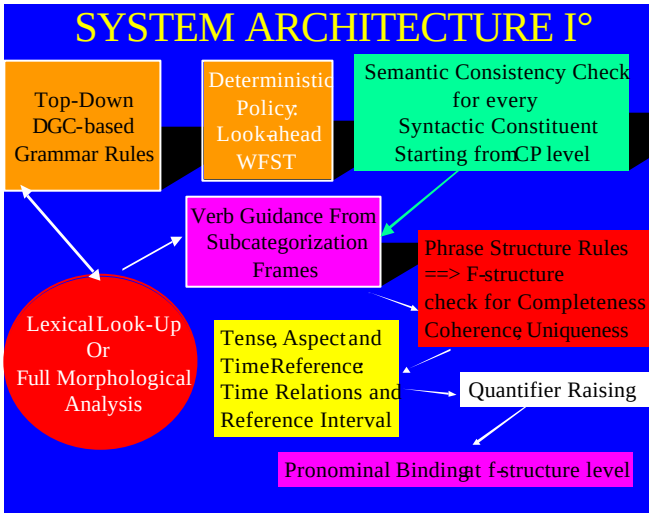

TABLE 1. GETARUNS PARSER

3.1 Parsing Strategies

Ambiguities dealt with by the parser go from different binding solution of a pronoun contained in a subordinate clause by two possible antecedents, chosen according to semantic and pragmatic strategies based on semantic roles and meaning associated to the subordinating conjunction, as in the following examples:
i.The authorities refused permission to the demonstrators because they feared violence

ii.The authorities refused permission to the demonstrators because they supported the revolution
iii.The cop shot the thief because he was escaping
iv.Mario criticized Luigi because he is hypercritical
v.Mario criticized Luigi because he ruined his party
vi.Mario called Luigi because he needed the file
vii.The thieves stole the paintings in the museum
viii.The thieves stole the painting in the night

The underlying mechanisms for ambiguity resolution takes one analysis as default in case it is grammatical and the other/s plausible interpretations are obtained by activating one of the available strategies which are linguistically and psychologically grounded.

From our perspective, it would seem that parsing strategies should be differentiated according to whether there are argument requirements or simply semantic compatibily evaluation for adjuncts. As soon as the main predicate or head is parsed, it makes available all lexical information in order to predict if possible the complement structure, or to guide the following analysis accordingly. As an additional remark, note that not all possible syntactic structure can lead to ambiguous interpretations: in other words, we need to consider only cases which are factually relevant also from the point of view of language dependent ambiguities. To cope with this problem, we built up a comprehensive taxonomy from a syntactic point of view which takes into account language dependent ambiguities

A. Omissibility of Complementator
- NP vs S complement
- S complement vs relative clause

B. Different levels of attachment for Adjuncts
- VP vs NP attachment of pp
- Low vs high attachment of relative clause

C. Alternation of Lexical Forms
- NP complement vs main clause subject

D. Ambiguity at the level of lexical category
- Main clause vs reduced relative clause
- NP vs S conjunction

E. Ambiguities due to language specific structural proprieties
- Preposition stranding
- Double Object
- Prenominal Modifiers
- Demonstrative-Complementizer Ambiguity
- Personal vs Possessive Pronoun

Here below is a snapshot of the output of the parser for the sentence: “The doctor called in the son of the pretty nurse who hurt herself/himself’. The c-structure is followed by the f-structure representation where binding has taken place and relative clause attachment is consequently realized with the higher or lower NP head according to the different agreement requirements imposed by the two reflexive pronouns herself/himself either with “the nurse” or with “the son”.

From a theoretical point of view this phenomenon is dubbed Short Binding, and is dealt with at the same level of Grammaticality Principles, rather than as a case of Anaphoric Binding. In this way a failure is imposed to the parser by agreement constraints between the reflexive pronoun and its binder.

References

[1] P. Tapanainen and Voutilainen A.(1994), Tagging accurately - don't guess if you know, ***Proc. of ANLP '94***, pp.47-52, Stuttgart, Germany.

[2] Brants T. & C.Samuelsson(1995), Tagging the Teleman Corpus, in ***Proc.10th Nordic Conference of Computational Linguistics***, Helsinki, 1-12.

[3] Lecomte J.(1998), Le Categoriseur Brill14-JL5 / WinBrill-0.3, INaLF/CNRS,

[4] Chanod J.P., P.Tapanainen (1995), Tagging French - comparing a statistical and a constraint-based method". ***Proc. EACL'95***, pp.149-156.

[5] Brill E. (1992), A Simple Rule-Based Part of Speech Tagger, in ***Proc. 3rd Conf. ANLP***, Trento, 152-155.

[6] Cutting D., Kupiec J., Pedersen J., Sibun P., (1992), A practical part-of-speech tagger, in ***Proc. 3rd Conf. ANLP***, Trento.

[7] Voutilainen A. and P. Tapanainen,(1993), Ambiguity resolution in a reductionistic parser, ***in Sixth Conference of the European Chapter of the ACL***, pp. 394-403. Utrecht.

[8] Delmonte R., E.Pianta(1996), "IMMORTALE - Analizzatore Morfologico, Tagger e Lemmatizzatore per l'Italiano", in ***Atti V Convegno AI*IA***, Napoli, 19-22.

[9] Delmonte R. G.A.Mian, G.Tisato(1986), A Grammatical Component for a Text-to-Speech System, ***Proceedings of the ICASSP'86***, IEEE, Tokyo, 2407-2410.

[10] Delmonte R., R.Dolci(1989), Parsing Italian with a Context-Free Recognizer, ***Annali di Ca' Foscari XXVIII***, 1-2,123-161.
[11] Delmonte R., E.Pianta(1999), Tag Disambiguation in Italian, in ***Proc.Treebanks Workshop ATALA***, pp.41-49.
[12] Delmonte R.(1999), From Shallow Parsing to Functional Structure, in ***Atti del Workshop AI*IA -*** IRST Trento,pp.8-19.
[13] Delmonte R.(2000), Parsing with GETARUN, ***Proc.TALN2000, 7° confèrence annuel sur le TALN***,Lausanne, pp.133-146.
[14] Delmonte R.(2000), Generating and Parsing Clitics with GETARUN, *Proc. CLIN'99*, Utrech, pp.13-27.

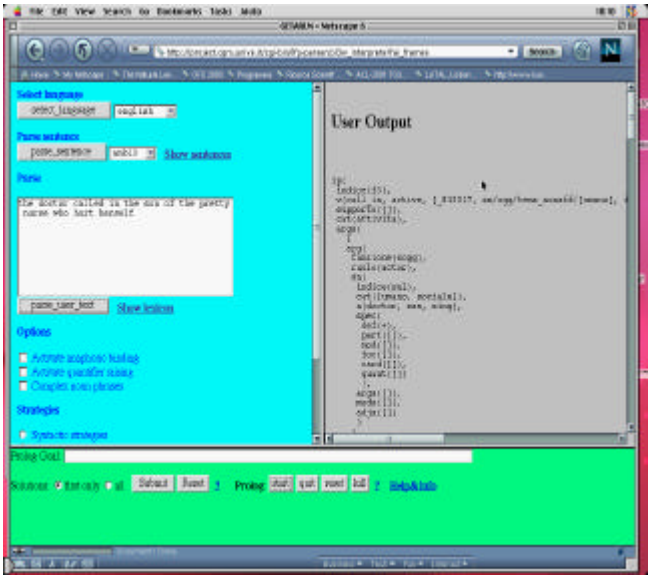

Fig. 3 GETARUN parsing from user window