ParParsing Preferences and Linguistic Strategies

Rodolfo Delmonte
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 implemented in our parser four parsing strategies that obey LFG grammaticality conditions and follow the hypothesis that knowledge of language is used in a "modular" fashion. The parsing strategies are the following: Minimal Attachment (MA), Functional Preference (FP), Semantic Evaluation (SE), Referential Individuation (RI). From the way in which we experimented with them in our implementation it appears that they are strongly interwoven. In particular, MA is dependent upon FP to satisfy argument/function interpretation principles; with semantically biased sentences, MA, FP and SE apply in hierarchical order to license a phrase as argument or adjunct. RI is required and activated every time a singular definite NP has to be computed and is dependent upon the presence of a discourse model. The parser shows garden path effects and concurrently produces a processing breakdown which is linguistically motivated. Our parser is a DCG (Pereira & Warren, 1980) is implemented in Prolog and obeys a topdown depth-first deterministic parsing policy.

1 Introduction

In order for a parser to achieve psychological reality it should satisfy three different types of requirements: psycholinguistic plausibility, computational efficiency in implementation, coverage of grammatical principles and constraints. Principles underlying the parser architecture should not conform exclusively to one or the other area, disregarding issues which might explain the behaviour of the human processor. In accordance with this criterion, we assume that the implementation should closely mimick phenomena such as garden path effects, or an increase in computational time in presence of semantically vs syntactically biased ambiguous structures. We also assume that a failure should ensue from strong garden path effects and that this should be justified at a psycholinguistic interpretation level.

Since we base most of our grammatical principles on LFG we assume that lexical information is the most important knowledge source in the processing of natural language. However, we also assume that all semantic information should be made to bear on the processing and this is only partially coincident with lexical information as stored in lexical forms. In particular, subcategorization, semantic roles and all other semantic compatibility evaluating mechanisms should be active while parsing each word of the input string. In addition, the discourse model and external knowledge of the world should be tapped when needed to disambiguate ambiguous antecedents.

Differently from what is asserted by global or full paths approaches (see Schubert, 1985; Bear & Hobbs, 1988; Hobbs, Stickel, Appelt, Martin, 1993), we believe that decisions on structural ambiguity should be reached as soon as possible rather than deferred to a later level of representation. In particular, Schubert assumes "...a full paths approach in which not only complete phrases but also all incomplete phrases are fully integrated into (overlaid) parse trees dominating all of the text seen so far. Thus features and partial logical translations can be propagated and checked for consistency as early as possible, and alternatives chosen or discarded on the basis of all of the available information (ibid., 249)." And further on in the same paper, he proposes a system of numerical 'potentials' as a way of implementing preference trade-offs. "These potentials (or levels of activation) are assigned to nodes as a function of their syntactic/semantic/pragmatic structure and the preferred structures are those which lead to a globally high potential. The total potential of a node consists of a) a negative rule potential, b) a positive semantic potential, c) positive expectation potentials contributed by all daughters following the head (where these decay with distance from the head lexeme), and d) transmitted potentials passed on from the daughters to the mother (ibid., 249)." Whereas Hobbs et al. theorize in favour of a delayed ambiguity resolution in LF, they suggest heuristic strategies which can account for most of them, "...a very good heuristic is obtained by using the following three principles: 1) favor right association; 2) override right association if a. the PP is temporal and the second nearest attachment site is a verb or event nominalization, or b. if the preposition typically signals an argument of the second nearest attchment site (verb or relational noun) and not of the nearest attachment site (ibid., 241)".

Other important approaches are represented by Hindle et al., 1993, who attempt to solve the problem of attachment ambiguity in statistical terms. The important contribution they made, which was not possible in the '80s, is constituted by the data on attachment typologies derived from syntactically annotated corpora. We will report similar data on Italian, derived from our corpus of Italian, currently being annotated as a syntactic treebank, a portion of which will be used as sample with comparable length to the English one: the authors worked on a test sample made up of 1000 sentences in which their parser identified ambiguous attachment conditions. In order to simulate automatic disambiguating procedures we shall use data derived from our shallow parser, which has an output comparable to the Fidditch parser quoted by Hindle et al. and shown in the article. The final output as recorded in the Treebank is the result of the concurrent work done by manual annotators, automatic validation procedures and my final supervision of the resulting constituent structures, visualized by a tree-viewer.

1.1 The parser

The parser we work with is organized as shown in Fig.1 and can deal with a certain number of linguistic phenomena at sentence level, while leaving other problems to be solved at discourse level.

The parser we present was conceived in the middle ’80s and started as a transfer module for a Machine Translation Expert system in a very restricted linguistic domain. Then it became a general parser for Italian and English, to be used with LFG students. German was added later on, in the beginning of ’90s. Since the people working at it were interested in the semantics as much as in the syntax, it was soon enriched with a Quantifier Raising algorithm and an Anaphoric Binding Module. In 1994 the Discourse Model and the Inferential Processes algorithms were developed. Finally in 1996 work on a Situational Semantics interface and on the Discourse Structure was carried out. These experiments were finally enriched – two years ago – with a number of Parsing Strategies procedures like setting up a Lookahead mechanism, a Well-Formed Substring Table and a number of other semantically and/or lexically based triggering lookup procedures.

We worked from the very beginning within LFG framework, which already from the start allowed us to think in terms of a much richer representation, closer to semantics, than just a context-free syntactic constituency. In particular, all levels of control mechanisms which allow coindexing at different levels of parsing gave us a powerful insight into the way in which the parser should be organized.

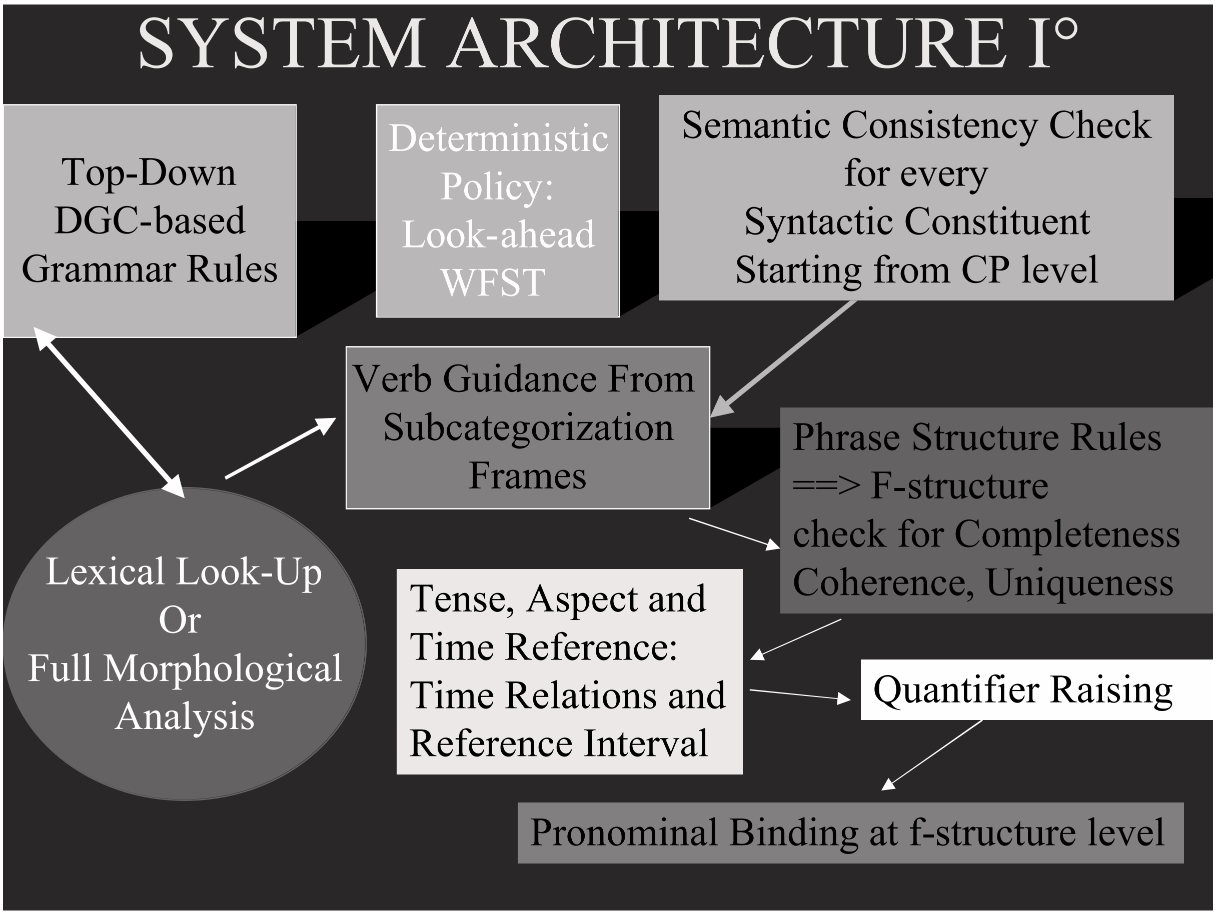
SYSTEM ARCHITECTURE I°
Top-Down DGC-based Grammar Rules
Deterministic Policy: Look-ahead WFST
Semantic Consistency Check for every Syntactic Constituent Starting from CP level
Verb Guidance From Subcategorization Frames
Phrase Structure Rules ==> F-structure check for Completeness Coherence, Uniqueness
Lexical Look-Up Or Full Morphological Analysis
Tense, Aspect and Time Reference: Time Relations and Reference Interval
Quantifier Raising
Pronominal Binding at f-structure level

***Figure 1:** Getaruns parser*

Yet the grammar formalism implemented in our system differs from the one suggested by the theory, in the sense that we do not use a specific Feature-Based Unification algorithm but a DCG-based parsing scheme. In order to follow LFG theory more closely, unification should have been implemented: but DCG gives us full control of a declarative rule-based system, where information is clearly spelled out and passed on and out to higher/lower levels of computation. The grammar is implemented in Prolog using XGs (extraposition grammars) introduced by Pereira (1981;1983). Prolog provides naturally for backtracking when allowed, i.e. no cut is present to prevent it. Furthermore, the instantiation of variables is a simple way for implementing the mechanism for feature percolation and/or for the creation of chains by means of index inheritance between a controller and a controllee, and in more complex cases, for instance in case of constituent ellipsis or deletion.

Apart from that, the grammar implemented is a surface grammar of the languages chosen. Also functional control mechanisms – both structural and lexical – have been implemented as close as possible to the original formulation, i.e. by binding an empty operator in the subject position of a propositional like open complement/predicative function, whose predicate is constituted by the lexical head.

Of course there are a number of marked differences in the treatment of specific issues, concerning Romance languages, which were not sufficiently documented in the linguistic literature at the time. In particular,

- we introduced an empty subject pronominal - little pro - for tensed propositions, which had different referential properties from big PRO; this had an adverse effect on the way in which c-structure should be organized. We soon realized that it was much more efficient and effective to have a single declarative utterance-clause level where the subject constituent could be either morphologically expressed or Morphologically Unexpressed. In turn MUS or little pros could be computed as variables in case the subject was realized in postverbal position. At the time LFG posited the existence of a rule for sentence structure which could be rewritten as VP in case there was no subject, MUS, or in case the subject was expressed in postverbal position, an approach that we did not implement;

- we also use functional constituents like CP and IP: CP typically contains Aux-to-Comp and other preposed constituents, adjuncts and others; IP contains negation, clitics, and tensed verbal forms, simple and complex, and expands VPs as complements and postverbal adjuncts;

- each constituent is semantically checked for consistency before continuing parsing; we also check for Uniqueness automatically by variable instantiation. But sometimes, in particular for subject-verb agreement we have to suspend this process to check for the presence of a postverbal NP constituent which might be the subject in place of the one already parsed in preverbal position!!;

- syntactic constituency is replicated by functional constituency: subject and object are computed as constituents of the annotated c-structure, which rewrite NP - the same for ncomp - this is essential to assign the appropriate annotated grammatical function; this does not apply to VP, a typical LFG functional non-substantial constituent;

- our lexical forms diverge from the ones used in the theoretical framework: we introduced aspectual categories, semantic categories and selectional restrictions in the main lexical entry itself;

- we also have semantic roles already specified in the lexical form and visible at the level of syntactic-semantic parsing;

- rather than generating a c-structure representation to be mapped onto the f-structure via an annotated c-structure intermediate level, we already generated a fully annotated c-structure representation which was then checked for Grammatical Principles Consistency at the level of number/type of arguments and of Adequacy for adjuncts, with a second pass on the output of the parser, on the basis of lexical form of each predicate and semantic consistency crossed checks for adjuncts.

All parser rules from lexicon to c-structure to f-structure amount to 1900 rules, thus subdivided:

1. Calls to lexical entries - morphology and lexical forms = 150 rules
2. Syntactic and semantic rules in the parser proper = 550 rules
3. Parsing strageties and other tools = 185 rules

All syntactic/semantic rules = 850 rules

4. Semantic Rules for F-Structure

Lexical Rules for Consistency and Control: - semantic rules 439

F-structure building, F-command: - semantic rules 170

Quantifier Raising and Anaphoric Control: - semantic rules 441

All semantic f-structure building rules: 1050

2 Theories for Parsing Strategies

Parsing theories are of two kinds: the first garden-path theory (hence GPT) is syntactically biased, and the second incremental-interactive theory (hence IIT) is semantically biased. There is a crucial difference between the two theories: whereas GPT claims that a single analysis of a syntactic structure ambiguity is initially constructed and passed to the semantic module, IIT claims that the syntactic module offers all grammatical alternatives to the semantic one, in parallel, to be evaluated. There is evidence that is favorable to both sides on this issue (see G.Altman, 1989).

The basic claims of GPT are that the sentence processing mechanism (the parser) uses a portion of its grammatical knowledge, isolated from world knowledge and other information, in initially identifying the relationships among the phrases of a sentence. IIT permits a far more intimate and elaborate interaction between the syntax and the semantic/referential modules. Altmann (1989) offers a functional argument against a system in which choices are initially made by a syntactic processor, and later corrected by appeal to meaning and context. He says that if referential or discourse information is available, only a strange processor would make decisions without appealing to it. It is also our opinion that all lower level constraints should work concurrently with higher level ones: in our parser all strategies are nested one inside another, where MA occupies the most deeply nested level.

The list of examples here below includes sentences used in the literature to support one or the other of the two parsing theories, GPT and IIT (see Altman (ed), 1989).

1. Mary put the book on the table.
2. Mary put the book on the table in her bag.
3. Mary saw the cop with the binoculars.
4. Mary saw the cop with the revolver.
5. The thief stole the painting in the museum.
6. The thief stole the painting in the night.
7. John saw Mary in the kitchen.
8. John saw Mary from the bathroom.

Altmann and Steedman (1989) note that several of the ambiguities present in these sentences and resolved by Minimal Attachment involve contrasts between NP modification and other structures. However, examples 2, 3 and 5 require linguistic knowledge.

In example 1 we assume that the pp should be computed as an argument of the main predicate "put", thus following a MA strategy when parsing the np "the book". However, the same strategy would lead to a complete failure in example 2, where the pp should be taken as np modifier. If we look at subcategorization requirements of the verb, example 1 constitutes a clear case of Verb Guidance and of semantic role satisfaction requirements: the main predicate requires an argument which is a locative, so at every decision point in which a pp might be taken, argument requirements should be accessed and a MA strategy imposed locally by FP.

Example 3 contains an instrumental adjunct: when the head preposition “with“ is met, the parser will not close the np “the cop“ and will continue building an internal pp modifier since preposition “with“ heads a compatible np modifier, a comitative. In our dictionary the verb “see“ has one single lexical entry but a list containing two different lexical forms. The first form in the list has a higher number of arguments,

I. see <SUB/perceiv, OBJ/theme, PCOMP/locat>

where the Pcomp predicates a location of the Object. In the second form the Pcomp is absent. In order for a Location pp to be accepted, the head preposition should be adequate, and “with“ does not count as such. In examples 3 and 4, the first decision must be taken when computing np structure. In fact, a pp headed by preposition “with“ is a semantically compatible np modifier – a comitative – and the analysis should be allowed to continue until the pp is fully analysed. In other words SE should verify pp attachment consistency inside the NP constituent. However, this may only happen in case MA is deactivated by FP, after matching with lexical information has failed.

In the following examples (5, 6), argument structure plays no role whatsoever: instrumentals, comitatives, locatives with predicate “steal“ are all cases of sentential adjuncts, and only SE can apply. As a matter of fact, “in the museum“ might be freely attached lower at NP level as well as higher at Sentence level. This is due to the fact that head preposition “in“ constitutes a viable local NP modifier and there are no argument requirements from the main verb predicate. However, “in the night“ is not a possible NP modifier and example 6 is a clear case of minimal attachment sentence. On the contrary, in example 5 PP attachment is ambiguous between a np internal modifier and VP level attachment.

In example 7 we understand that the location at which Mary was when the seeing event took place is the kitchen: we also understand that John might have been in the same location or in a different one, already provided by the previous context, and this can be achieved by FP which activates MA and makes the locative PP available at VP level.

In example 8, on the contrary, we understand that the location from the which the seeing event took place is the bathroom and that John was certainly there; however we are given no information whatsoever about Mary’s location. This case is treated as the previous one, except that the PP is computed as sentence adjunct rather than as VP complement.

As for Referential Identification, the following examples are disambiguated at the level of pronominal binding:

9. The doctor called in the son of the pretty nurse who hurt herself.

10. The doctor called in the son of the pretty nurse who hurt himself.

Pronominal binding is a level of computation that takes place after f-structure has been completely checked and built in LFG –the same applies in GB framework, where S-structure gives way to L-structure and this is where binding takes place. In order to take pronominal binding into account we should be able to backtrack from one level of representation f-structure, to a lower level c-structure and to attach the predicative adjunct constituted by the relative clause at a higher level in case the reflexive pronoun is masculine. A very time-consuming and risky process.

We propose RI, instead, a strategy completely cast in IIT, which addresses a specific semantic level of representation: the Discourse Model, or History List of all entities appeared in the text under analysis with their properties. RI should be activated whenever a Definite NP is completely parsed and adjunct modifiers have to be attached locally, in other words while still in the process of c-structure building.

In examples 9 and 10, we assume that RI should be activated to ascertain whether in the DM there is an entity which has the property of being hurt, associated to the individual “son“ or to the individual “nurse“. In order to produce this kind of interpretation process, we also assume that relative clauses represent presupposed or known facts in case they are factual and these facts should be present in the DM. No syntactic and semantic constraints may be invoked in favour of one or the other interpretation: attachment of the relative adjunct is simply predictable from the DM. We query in the history list to recover the semantic identifier associated to the entity “book“. In turn this Id is used to recover all the properties associated with it and then to match them with the properties described in the relative clause which have to be organized in a semantically adequate representation. If the intersection is null the relative clause is attached locally. However the presence of a reflexive pronoun is the trigger for the local search of an adequate antecedent which in this case would push the system to apply MA and deposit the predicative adjunct in the WFST to be processed higher up.

2.1 Sorting out Language Dependent Differences

In our perspective we would like to take a very pragmatic and experimental stand on the problem of ambiguity. In the first place we want to look only at structural ambiguity and build up a comprehensive taxonomy from a syntactic point of view; secondly, we want to spot language dependent ambiguities in order to define the scope of our research appropriately.

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

3 Implementing Parsing Strategies

Among contemporary syntactic parsing theories, the garden-path theory of sentence comprehension proposed by Frazier (1987a, b), Clifton & Ferreira (1989) among others, is the one that most closely represents our point of view. It works on the basis of a serial syntactic analyser, which is top-down, depth-first –i.e. it works on a single analysis hypothesis, as opposed to other theories which take all possible syntactic analysis in parallel and feed them to the semantic processor.

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 structures 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.

We implemented two simple enough mechanisms in order to cope with the problem of nondeterminism and backtracking. At bootstrapping we have a preparsing phase where we do lexical lookup and we look for morphological information: at this level of analysis of all input tokenized words, we create a stack of pairs input wordform - set of preterminal categories, where preterminal categories are a proper subset of all lexical categories which are actually contained in our lexicon. The idea is simply to prevent attempting the construction of a major constituent unless the first entry symbol is well qualified. When consuming any input wordform, we remove the corresponding pair on top of stack.

In order to cope with the problem of recoverability of already built parses we built a more subtle mechanism that relies on Kay's basic ideas when conceiving his Chart (see Kay, 1980; Stock, 1989). Differently from Kay, however, we are only interested in a highly restricted topdown depthfirst parser which is optimized so as to incorporate all linguistically motivated predictable moves.

An already parsed PP is deposited in a table lookup accessible from higher levels of analysis and consumed if needed. To implement this mechanism in our DCG parser, we assert the content of the PP structure in a PP table lookup storage which is then accessed whenever there is an attempt on the part of the parser to build up a PP. In order to match the input string with the content of the store phrase, we implemented a WellFormed Substring Table (WFST) as suggested by Woods (1973). Now consider the way in which a WFST copes with the problem of parsing ambiguous structure in its chart. It builds up a table of well-formed substrings or terms which are partial constituents indexed by a locus, a number corresponding to their starting position in the sentence and a length, which corresponds to the number of terminal symbols represented in a term. For our purposes, two terms are equivalent in case they have the same locus and the same length.

In this way, the parser would consume each word in the input string against the stored term, rather than against a newly built constituent. In fact, this would fit and suit completely the requirement of the parsing process which rather than looking for lexical information associated to each word in the input string, only needs to consume the input words against a preparsed well-formed syntactic constituent.

To give a simple example, suppose we have taken the PP "in the night" within the NP headed by the noun "painting". At this point, the lookahead stack would be set to the position in the input string that follows the last word "night". As a side-effect of failure in semantic compatibility evaluation within the NP, the PP "in the night" would be deposited in the backtrack storage. The input string would be restored to the word "in", and analysis would be restarted at the VP level. In case no PP rule is met, the parser would continue with the input string trying to terminate its process successfully. However, as soon as a PP constituent is tried, the storage is accessed first, and in case of non emptiness its content recovered. No structure building would take place, and semantic compatibility would take place later on at sentence level. The parser would only execute the following actions:

- match the first input word with the (preposition) head of the stored term
- accept new input words as long as the length of the stored term allows it by matching its length with the one computed on the basis of the input words.

3.1 Principles of Sound Parsing

• **Principle One:** Do not perform any unnecessary action that may overload the parsing process: follow the *Strategy of Minimal Attachment*;

• **Principle Two:** Consume input string in accordance with look-ahead suggestions and analyse incoming material obeying the *Strategy Argument Preference*;

• **Principle Three:** Before constructing a new constituent, check the storage of WellFormed Substring Table (WFST). Store constituents as soon as they are parsed on a stack organized as a WFST;

• **Principle Four:** Interpret each main constituent satisfying closer ties first - predicate-argument relations - and looser ties next - open/closed adjuncts as soon as possible, according to the Strategy of Functional Preference;

• **Principle Five:** Erase short-memory stack as soon as possible, i.e. whenever main constituents receive Full Interpretation.

•***Strategy Functional Preference***: whenever possible try to satisfy requirements posed by predicate-argument structure of the main governing predicate as embodied in the above Principles; then perform semantic compatibility checks for adjunct acceptability.

•***Strategy Minimal Attachment:*** whenever Functional Preference allows it apply a Minimal Attachment Strategy.

The results derived from the application of Principle Four are obviously strictly linked to the grammatical theory we adopt, but they are also the most natural ones: it appears very reasonable to assume that arguments must be interpreted before adjuncts and that in order to interpret major constituents as arguments of some predicate we need to have a completed clause level structure. In turn adjuncts need to be interpreted in relation both to clause level properties like negation, tense, aspect, mood, possible subordinators, and to arguments of the governing predicate in case they are to be interpreted as open adjuncts.

As a straightforward consequence, owing to Principle Five we have that reanalysis of a clause results in a Garden Path (GP) simply because nothing is available to recover a failure that encompasses clause level reconstruction: we take that GP obliges the human processor to dummify all naturally available parsing mechanisms, like for instance look-ahead, and to proceed by a process of trial-and-error to reconstruct the previously built structure in order not to fall into the same mistake.

4 Experimental Results

Here below we give parsing times on a SparcStation 5 for ex.1: `Mary put the book on the table.` With no parsing strategy activated, the PP is taken as adjunct in the NP constituent and the oblique argument is left indefinite.

Time: 1.3 sec.; with only Minimal Attachment activated, Time decreases: 0,87 sec.; with Functional Preference and Minimal Attachment activated, Time increases slightly: 1.05 sec. With Semantic Evaluation and no other strategy activated there is a big Time increase: 2.8 sec. The reason of this increase lies in the fact that SE for arguments of a given predicate is only activated at the end of the parsing process with Functional Unification: at this level the WFST is still filled with accepted constituents which will then be erased. Finally, for the strategy of

Referential Identification we have ex.9 which requires no strategy activation; however in ex.10 the relative clause should be attached adequately on a higher level. Time depends on the size of the DM: With a small one, time increases only a small fraction required to perform a query in the history list to recover the semantic identifier associated to the entity "book". The same procedure would then apply for the higher NP "the son". Overall time increases again and rises to 3.7 sec.

5 Treebank Derived Structural Relations

As noted above in the Introduction, an important contribution to the analysis of PP attachment ambiguity resolution procedures is constituted by the data made available in syntactic Treebanks. Work still underway on our Venice Italian Corpus of 1 million occurrences revealed a distribution of syntactic-semantic relations which is very similar to the one reported by Hindle et al. in their recent paper and shown in Table 1 below.

Argument noun	378	39.5%	
Argument verb	104	11.8%	
Light verb	19	2.1%	
Small clause	13	1.5%	
Idiom	19	2.1%	57%
Adjunct noun	**91**	**10.3%**	
Adjunct verb	**101**	**11.5%**	
Locative indeterminacy	**42**	**4.8%**	
Systematic indeterminacy	**35**	**4%**	
Other	**78**	**8.8%**	**39.4%**
	TOTAL	**880**	**100%**

Table 1: *Shallow Parsing & Statistical Approaches*
(Data from D.Hindle & M.Roth, "Structural Ambiguity and Lexical Relations")

As the data reported above clearly show, most of the prepositional phrases are constituted by arguments of Noun, rather than of Verb. As the remaining data, adjuncts are represented approximately by the same amount of cases, 11% of the sample text.

Venice Italian Corpus
1 million tokens

- **All prepositions - 54 different types or wordforms:**
170,000 occurrences

Argument-like prepositions

DI/of and its amalgams	**78,077 --> 46%**	
A/to and its amalgams	**29,191 --> 17.2%**	
DA/by-from and its amalgams	**13,354 --> 7.9%**	**71.1 %**

Adjunct-like prepositions

IN and its amalgams -	***21,408 --> 12.6%***	
PER and its amagalms -	***12,140 --> 7.1%***	
CON and its amalgams -	***5,958 --> 3.5%***	***23.2%***

Table 2: *Shallow Parsing & Statistical Approaches*

We started looking at our data by collecting all information on prepositions as a whole and then we looked into our Treebank and looked for their relations as encoded in the syntactic constituent structure. Here below we report data related to prepositions for the whole corpus: Notice that in Italian as in English, the preposition *of / di* would be used mainly as a Noun argument/modifier PP.

In contrast to English, however, nominal premodifiers do not exist in Italian, and the corresponding Italian Noun-Noun modification or argument relation without preposition would be postnominal. Such cases are not very frequent and constitute less than 1% of Noun-Noun head relations. We then selected 2000 sentences and looked at all prepositional phrases in order to highlight their syntactic and semantic properties, and we found out the following: *(see page 72)*

PPs not headed by DA or DI	3977		51%		
Argument of verb		944		23.7%	
Argument of Noun		1300		32.7%	
Adjunct of Noun or Verb	1733		43.6%		
PPs headed by DA			*504*		*6.5%*
Argument of Verb		*164*		*32.5%*	
Argument of Noun		*114*		*22.6%*	
Adjunct of Noun or Verb	*226*		*44.9%*		
PPs headed by DI			*3314*		*42.5%*
Argument of Verb		*72*		*2.17%*	
Argument of Noun		*2733*		*82.5%*	
Adjunct of Noun or Verb	*509*		*15.4%*		
TOTAL				**7795**	**100%**
Arguments of Verb			1180		15%
Arguments of Noun			4147		53%
Ambiguous PPs			2468		32%

***Table 3:** Quantitative Syntactic and Semantic Distribution of PPs in VIC*

- The number of prepositional phrases in Italian texts is four times bigger as the one reported for English Texts, and this might be due to the poor use of nominal modifiers which in Italian can only be post-modifiers, attested from an analysis of the sample text;
- PPs Arguments of Nouns are 53% in Italian and 39% in English, i.e. 14% more in Italian;
- PPs Arguments of Verbs are 15% in Italian and 17% in English – if we sum all argument types and idioms
- together -, i.e. 2% more in English;
- Adjuncts of Nouns and Verb are 31% in English and 32% in Italian.

Thus, the only real major difference between the two languages can be traced back to the behavior of PP noun arguments, which in turn can be traced back to a language specific difference: the existence of prenominal modifiers in English and not in Italian – not yet substituted by the use of postnominal modification.

References

Altman G.T.M. (ed.) (1989), Language and Cognitive Processes 4, 3/4, Special Issue – Parsing and Interpretation.

Bianchi D., R.Delmonte, E.Pianta (1992), GETA_RUN – A General Text Analyzer with Reference Understanding, in Proc. 3rd Conference on Applied Natural Language Processing, Systems Demonstrations, Trento, ACL 1992, 9–10.

Clifton C., & F. Ferreira (1989), Ambiguity in Context, in G.Altman (ed), Language and Cognitive Processes, op.cit., 77–104.

Delmonte R. (1984), La “syntactic closure“ nella Teoria della Performance, Quaderni Patavini di Linguistica, 4, Padova, 101–131.

Delmonte R. (1992), Linguistic and Inferential Processing in Text Analysis by Computer, Padova, Unipress.

Frazier L. (1987a), Sentence processing, in M.Coltheart (ed), Attention and Performance XII, Hillsdale, N.J., Lawrence Elbaum.

Frazier L. (1987b), Theories of sentence processing, in J.Garfield (ed), Modularity in knowledge representation and natural language understanding, Cambridge, Mass., MIT Press, 291–308.

Garrod S., Sanford T. (1988), Thematic Subjecthood and Cognitive Constraints on Discourse Structure, Journal of Pragmatics 12, 599–534.

D.Hindle & M.Roth (1993), Structural Ambiguity and Lexical Relations, Computational Linguistics 19, 1, 103–120.

Kay Martin (1980), Algorithm Schemata and Data Structures in Syntactic Processing, CSL-80–12, Xerox Corporation, Palo Alto Research Center.

Kennedy A., et al. (1989), Parsing Complements, in G.Altman (ed), Language and Cognitive Processes, op.cit., 51–76.

Pereira F. and Warren D. (1980), Definite Clause Grammars for Language Analysis. A Survey of the Formalism and a Comparison with Augmented Transition Networks. Artificial Intelligence, 13, 231–278.

Pritchett B.L. (1992), Grammatical Competence and Parsing Performance, The University of Chicago Press, Chicago.

Schubert L.K. (1984), On Parsing Preferences, Proc. of COLING, 247–250.

Steedman M.J. & Altmann G.T.M. (1989), Ambiguity in Context: a Reply, in Altman (ed), op.cit., 105–122.

Stock O. (1989), Parsing with flexibility, dynamic strategies and idioms in mind, in Computational Linguistics, 15, 1, 1–18.

Whitelock P., M.M.Woods, H.L.Somers, T.Johnson, P.Bennett (eds.) (1987), Linguistic Theory and Computer Applications, Academic Press, London.

Woods W.A. (1973), An Experimental Parsing System for Transition Network Grammars, in Rustin (ed) (1973), Natural Language Processing, Algorithmic Press, New York, pp.111–154.