Deep & Shallow Linguistically Based Parsing: Parameterizing Ambiguity in a Hybrid Parser

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: project.cgm.unive.it

Abstract

In this paper we will present an approach to natural language processing which we define as "hybrid", in which symbolic and statistical approaches are reconciled. In fact, we claim that the analysis of natural language sentences (and texts) is at the same time a deterministic and a probabilistic process. In particular, the search space for syntactic analysis is inherently deterministic, in that it is severely limited by the grammar of the specific language, which in turn is constituted by a set of peripheral rules to be applied in concomitance with the more general rules of core grammar. Variations on these rules are only determined by genre, which can thus contribute a new set of peripheral rules, or sometimes just a set of partially overlapping rules to the ones already accepted by a given linguistic community. In the paper we will criticize current statistical approaches for being inherently ill-founded and derived from a false presupposition. People working in the empirical framework have tried to credit the point of view that what happened to the speech research paradigm was also applicable to the NLP paradigm as a whole. In other words, the independence hypothesis which is at the heart of the use of Markov models imported from the speech community into the empirical statistical approaches to NLP does not seem to be well suited to the task at hand simply because the linguistic unit under consideration – the word, the single tag or both – are insufficient to assure enough contextual information, given language model building techniques based on word-tags with tagsets containing only lexical and part-of-speech information. In line with the effort carried out by innovative approaches like the ones proposed within the LinGO ERG proposal, we purport our view that the implementation of sound parsing algorithm must go hand in hand with sound grammar construction. Extragrammaticalities can be better coped with within a solid linguistic framework rather than without it. A number of parsing strategies and graceful recovery procedures are then proposed which follow a strictly parameterized approach to their definition and implementation. Finally, a shallow or partial parser needs to be implemented in order to produce the default baseline output to be used by further computation, like Information Retrieval, Summarization and other similar tools which however need precise linguistic information with a much higher coverage than what is being recently offered by the currently available parsers.

1. Introduction

In this paper we will present an approach to natural language processing which we define as "hybrid", in which symbolic and statistical approaches are reconciled. In fact, we claim that the analysis of natural language sentences (and texts) is at the same time a deterministic and a probabilistic process. In particular, the search space for syntactic analysis is inherently deterministic, in that it is severely limited by the grammar of the specific language, which in turn is constituted by a set of peripheral rules to be applied in concomitance with the more general rules of core grammar. Variations on these rules are only determined by genre, which can thus contribute a new set of peripheral rules, or sometimes just a set of partially overlapping rules to the ones already accepted by a given linguistic community.

Probabilistic processes come into play whenever lexical information is tapped to introduce knowledge into the parsing process. The determination of meaning at propositional level is the product of the meaning of its internal component, the process being compositional in nature. At a semantic level, meaning variations at word and at phrase level come into play to contribute uncertainty to the overall process. Word sense disambiguation procedures coupled with constituent attachment disambiguation should be used to take decisions as to the most probable clause level analysis to assign to a given input sentence.

In the paper we will criticize current statistical approaches for being inherently ill-founded. People working in the empirical framework have tried to credit the point of view that what happened to the speech research paradigm was also applicable to the NLP paradigm as a whole. People like Ken Church have paved the way to the NLP scientific community by presenting the task of NLP as being parallel with what the speech community did, in the Introduction to the Special Issue on Using Large Corpora published by Computational Linguistics in 1993[1]. In this seminal but partially misleading article, the author claimed that what happened to the speech community in the last 20 years or so, that is a slow trasformation of their approach to natural language understanding from a knowledge-based approach to an empirical statistically-based approach, could also be applied to NLP. Seen that both communities were apparently dealing with the same basic materials, linguistic units of some kind, the trasition to the empirical approach was simply to be treated as a truism by people of the NLP community. The only true part of the argument was on the contrary constituted by the need felt by most computational linguists to move away from the analysis of hand-made list of sentences and start tackling 'real texts'.

That is, the need to move away from what theoretical linguists would still regard as a sufficiently representative sample of the language under analysis - usually a list of 3-4000 simplex sentences; in order to start using large corpora. However, it is just the fact that the linguistic units being addressed as their main object were totally different, that the comparison does not hold and is badly misleading. In the section below, we will discuss in detail the reason why in our opinion the empirical statistical approach to NLP it its current experimental and empirical design should not be pursued, and in fact this would be the same reason why the speech community has come to the same conclusions already some time ago with respect to the need to address higher level linguistic units in the speech waveform usually referred to as prosodic units[17]. In particular, both the speech synthesis community and the speech recognition community have implicitly admitted to the fact that the choice of the linguistic units to be addressed, i.e. a segmental unit constitutes nowadays the bottleneck for further improvements in the field (see R.Sproat and J.van Santen, in [18]).

On a strictly intuitive basis, the segmental unit approach is wrong for the simple reason that one would need to model information coming from all linguistic levels into the one single segment - being it a phone, a phoneme, a diphone unit or a triphone unit from the n-gram approach advocated by the speech community and transferred by the new empiricists onto the part-of-speech tagging task. This position is both untenable and implausible. It is untenable for reasons related to tagset size and linguistic coverage, i.e. how big a training corpus should be in order to cope with the well-known problem of data sparseness or sparcity. For instance in the LinGO framework, the tagset being used amounts to over 8,000 different single tags, which makes it very hard even with a database of 10,000 utterance to make up a representative and statistical useful training corpus, as their authors comment in the entry webpage of the project at

http://lingo.stanford.edu

under the title "Why Another (Type of) Treebank?" which we report here below:

> "For the past decade or more, symbolic, linguistically oriented methods like those pursued within the HPSG framework and statistical or machine learning approaches to NLP have typically been perceived as incompatible or even competing paradigms; the former, more traditional approaches are often referred to as 'deep' NLP, in contrast to the comparatively recent branch of language technology focussing on 'shallow' (text) processing methods. Shallow processing techniques have produced useful results in many classes of applications, but they have not met the full range of needs for NLP, particularly where precise interpretation is important, or where the variety of linguistic expression is large relative to the amount of training data available. On the other hand, deep approaches to NLP have only recently been able to achieve broad enough grammatical coverage and sufficient processing efficiency to allow the use of HPSG-type systems in certain types of real-world applications.
>
> Fully-automated, deep grammatical analysis of unrestricted text remains an unresolved challenge. In particular, realistic applications of analytical grammars for natural language parsing or generation require the use of sophisticated statistical techniques for resolving ambiguities. We observe general consensus on the necessity for bridging activities, combining symbolic and stochastic approaches to NLP…
>
> An important recent advance in this area has been the application of log-linear models (Agresti, 1990) to modeling linguistic systems. These models can deal with the many interacting dependencies and the structural complexity found in constraint-based or unification-based theories of syntax. The availability of even a medium-size treebank would allow us to begin exploring the use of these models for probabilistic disambiguation of HPSG grammars."

And further on the webpage includes details of the implementation, which we report here below,

> "The key innovative aspect of the Redwoods approach to treebanking is the anchoring of all linguistic data captured in the treebank to the HPSG framework and a generally-available broad-coverage grammar of English, viz. the LinGO English Resource Grammar. Unlike existing treebanks, there will be no need to define a (new) form of grammatical representation specific to the treebank (and, consequently, less dissemination effort in establishing this representation). Instead, the treebank will record complete syntacto-semantic analyses as defined by the LinGO ERG and provide tools to extract many different types of linguistic information at greatly varying granularity. Depth of Representation and Transformation of Information Internally, the [incr tsdb()] database records analyses in three different formats, viz. (i) as a derivation tree composed of identifiers of lexical items and constructions used to construct the analysis, (ii) as a traditional phrase structure tree labeled with an inventory of some fifty atomic labels (of the type S, NP, VP et al.), and (iii) as an underspecified MRS meaning representation… While (ii) will in many cases be similar to the representation found in the Penn Treebank, (iii) subsumses the functor-argument (or tectogrammatical) structure as it is advocated in the Prague Dependency Treebank or the German TiGer corpus. Most importantly, however, representation (i) provides all the information required to replay the full HPSG analysis…"

Even though the overall tone of the researchers involved in the LinGO consortium is enthusiastic the actual coverage of the PET parser in real texts is as usual limited by grammar and vocabulary coverage. However we find the approach in line with ours even though the underlying technical framework is totally different.

1.2 Shallow and Partial Parsing and Statistical Processing

In their Chapter - Language Analysis and Understanding [12] in the section dedicated to Shallow Parsing [ibid:113-114], they use the term shallow syntax as a generic term for analyses that are less complete than the output from a conventional parser. The output from a shallow analysis is not a phrase-structure tree. A shallow analyzer may identify

some phrasal constituents, such as noun phrases, without indicating their internal structure and their function in the sentence. Another type of shallow analysis identifies the functional role of some of the words, such as the main verb, and its direct arguments. Systems for shallow parsing normally work on top of morphological analysis and disambiguation. The basic purpose is to infer as much syntactic structure as possible from the lemmata, morphological information, and word order configuration at hand. Typically shallow parsing aims at detecting phrases and basic head/modifier relations. A shared concern of many shallow parsers is the application to large text corpora. Frequently partial analyses are allowed if the parser is not potent enough to resolve all problems.

Abney [1] comments on statistical methods applied to the problem of Part-of-Speech Tagging as being a fairly success story. People engaging in this kind of pioneering research effort at the beginning of the '90s showed that it was possible to "carve part-of-speech disambiguation out of the apparently monolithic problem of natural language understanding, and solve it with impressive accuracy" [1:1]. What the people [2,3,4] involved in that approach actually were interested in showing was that even if the exact solution to the NLU problem is far beyond reach, a reasonable approximate solution is quite feasible.

In [1] the author discusses the feasibility of another important aspect of the NLU problem: that of syntactic analysis, by proposing as a solution what he defines "Partial Parsing". This is regarded as a cover term for a range of different techniques for recovering some but not all of the information contained in a traditional syntactic analysis. As he comments "Partial parsing techniques, like tagging techniques, aim for reliability and robustness in the face of the vagaries of natural text, by sacrificing completeness of analysis and accepting a low but non-zero error rate."[1:3]

Further on in the same paper, we are told that a 5% error rate is certainly a remarkable achievment in terms of accuracy, and can be achieved in a very short term indeed - one month work of a computational linguist. However, if we consider the sentence as the relevant unit onto which to gauge the goodness of such an accuracy figure, we come up with a completely different figure: assuming an average of 20-word sentences and 4% per-word error rate we end up with a 56% per-sentence error rate. To get a 4% per-sentence error rate, we require accuracy figures which range beyond 99%, actually 99.98%. This is clearly unfeasible for any statistically or even rule-based tagger presented in the literature.

Partial parsing tries to offer a solution to the problem posed by unrestricted texts to traditional parsers which basically due to the incompleteness of both lexicon and grammar are subject to failures and errors. Errors are also a subproduct of the length of sentences and the inherent ambiguity of grammars. What partial parsers do is recovering the nonrecursive core of constituent structure by factoring out the parse into those pieces of structure that can be reliably recovere with a small amount of syntactic information. This is usually done without using lexical information as would typically do all unfication based parsers. Chunks and simplex clauses can then safely be used for bootstrapping lexical association information which is used to take decisions related to attachment of arguments and adjuncts. The output of any such chunkers can be regarded as a useful intermediate representation to be used for any further computation. In terms of efficiency, as [1:10] reports, the fastest parsers are all deterministic rule-based partial parsers.

We have developed our partial parser as a finite-state cascaded machine that produces a final parser by cycling on the input and passing the output of each parse to the following FSA. The parser was originally a recursive transition network, and has been built expressedly to eliminate recursion from the parsing process. However, even if the partial parser is good at recognizing constituent chunks or to do phrase-spotting, without having to analyze the entire sentence. When it comes to clauses the error rate increases a lot up to statistically valid threshold of 5-6%. As [5] has shown, a partial parser can be put to use in a variety of ways, in particular in extracting subject-verb and verb-object pairs in order to provide a crude model of selectional restrictions.

1.3 Issues Related to the use of Partial and Shallow Approaches

In his Chapter on Sentence Modeling and Parsing, Fernando Pereira [13] defines what in his opinion are the main issues in applying linguistic theory to the development of computational grammars: coverage, predictive power and computational requirements. However this is done in order to promote the use of statistically based approaches to parsing and thus the issues are highlighted as shortcomings.

As far as Coverage is concerned his comment is the following:

> "Linguistic theories are typically developed to explain puzzling aspects of linguistic competence, such as the relationships between active and passive sentences, the constraints on use of anaphoric elements, or the possible scopes of quantifying elements such as determiners and adverbs. However, actual language involves a wide range of other phenomena and constructions, such as idioms, coordination, ellipsis, apposition and extraposition, which may not be germane to the issues addressed by a particular linguistic theory or which may offer unresolved challenges to the theory. Therefore, a practical grammar will have to go far beyond the proposals of any given theory to cover a substantial proportion of observed language. Even then, coverage gaps are relatively frequent and difficult to fill, as they involve laborious design of new grammar rules and representations."

Then he continues by lamenting the lack of Predictive Power of linguistic grammars, which in his opinion

> "… being oriented towards the description of linguistic competence, are not intended to model distributional

> regularities arising from pragmatics, discourse and conventional use that manifest themselves in word and construction choice. Yet those are the regularities that appear to contribute most to the estimation of relative likelihoods of sentences or analyses." [ibid:137]

As far as Computational Requirements are concerned, recent implementations seem to have made progress in the direction towards tractable grammatical formalisms which are reported to constitute polynomial-time and space parsing algorithms: however he then asserts that, "… even polynomial-time algorithms may not be sufficiently fast for practical applications, given effect of grammar size on parsing time." [ibid:138]
Eventually in his "Future Directions" paragraph, Pereira comments on the current challange which we also endorse fully:

> "The issue that dominates current work in parsing and language modeling is to design parsers and evaluation functions with high coverage and precision with respect to naturally occurring linguistic material (for example, news stories, spontaneous speech interactions). Simple high-coverage methods such as n-gram models miss the higher-order regularities required for better prediction and for reliable identification of meaningful relationships, while complex hand-built grammars often lack coverage of the tail of individually rare but collectively frequent sentence structures (cf. Zipf's law). Automated methods for grammar and evaluation function acquisition appear to be the only practical way to create accurate parsers with much better coverage. The challenge is to discover how to use linguistic knowledge to constrain that acquisition process." [ibid:140]

More or less of the same overall tone is the Chapter on Robust Parsing by Ted Briscoe [14], where he comments on the question of disambiguation which will also be discussed by us further on in this paper. Here is his comment:

> "Despite over three decades of research effort, no practical domain-independent parser of unrestricted text has been developed. Such a parser should return the correct or a useful close analysis for 90% or more of input sentences. It would need to solve at least the following three problems, which create severe difficulties for conventional parsers utilizing standard parsing algorithms with a generative grammar:
>
> 1. chunking, that is, appropriate segmentation of text into syntactically parsable units;
> 2. disambiguation, that is, selecting the unique semantically and pragmatically correct analysis from the potentially large number of syntactically legitimate ones returned; and
> 3. undergeneration, or dealing with cases of input outside the systems' lexical or syntactic coverage.
>
> Conventional parsers typically fail to return any useful information when faced with problems of undergeneration or chunking and rely on domain-specific detailed semantic information for disambiguation.
>
> The problem of chunking is best exemplified by text sentences (beginning with a capital letter and ending with a period) which |and this sentence is an example| contain text adjuncts delimited by dashes, brackets or commas which may not always stand in a syntactic relation with surrounding material… an analysis of the 150K word balanced Susanne Corpus… reveals that over 60% of sentences contain internal punctuation marks and of these around 30% contain text-medial adjuncts…
>
> Disambiguation using knowledge-based techniques requires the specification of too much detailed semantic information to yield a robust domain-independent parser. Yet analysis of the Susanne Corpus with a crude parser suggests that over 80% of sentences are structurally ambiguous… (statistically based) systems have yielded results of around 75% accuracy in assigning analyses to (unseen) test sentences from the same source as the unambiguous training material. The barrier to improvement of such results currently lies in the need to use more discriminating models of context, requiring more annotated training material to adequate estimate the parameters of such models. This approach may yield a robust automatic method for disambiguation of acceptable accuracy, but the grammars utilized still suffer from undergeneration, and are labour-intensive to develop. Undergeneration is a significant problem, in one project, a grammar for sentences from computer manuals containing words drawn from a restricted vocabulary of 3000 words which was developed over three years still failed to analyze 4% of unseen examples… This probably represents an upper bound using manual development of generative grammars; most more general grammars have far higher failure rates in this type of test. Early work on undergeneration focussed on knowledge-based manual specification of error rules or rule relaxation strategies… This approach, similar to the canonical parse approach to ambiguity, is labour-intensive and suffers from the difficulty of predicting the types of error or extragrammaticality liable to occur." [ibid:142]

In his Chapter on Statistical Parsing, John A.Carroll [10] gives a rather pessimistic view of current and future possibilities for statistical approaches in NLP. This even though he is among the people working within the HPSG constrain unification framework quoted above in the LinGO project, who seem convinced of the contrary and are actually working with optimistic plans as the presentation of the Redwoods Treeback effort and subsequent parser creation and testing demonstrates. In [10:525] Carroll states what in his opinion, are the major problems that parsing of natural language should address and they are:
a. how to resolve the (lexical, structural, or other) ambiguities that are inherent in real-world natural language text;
b. how to constrain the form of analyses assigned to sentences, while still being able to return "reasonable" analyses for as wide a range of sentences as possible.
He then criticizes NLP approaches wrought within the generative linguistic tradition because in his opinion they have a number of major drawbacks that disqualify them as adequate and successful candidates for the analysis of real texts. These parsing systems - like ours - uses hand-built grammars in conjunction with parsing algorithms which either,
c. return all possible syntactic analyses, which would then be passed on to detailed, domain-dependent semantic and pragmatic processing subsystems for disambiguation;
d. use special purpose, heuristic parsing algorithms that are tuned to the grammar and

e. invoke hand-coded linguistic or domain-specific heuristics to perform disambiguation and
f. invoke grammar relaxation techniques to cope with extragrammatical input
a position which I find totally in line with our approach except for c, being deterministic and as such using all possible semantic knowledge as soon as possible and in any case before any major constituent is being licensed.
However, Carroll [10:526] finds that such an approach cannot be good because it has the following dramatic drawbacks:
g. computing the full set of analyses for a sentence of even moderate length with a wide-coverage grammar is often intractable (we also agree with this point)
h. if this is possible, there is still the problem of how to apply semantic processing and disambiguation efficiently to a representation of a (possibily very) large set of competing syntactic analyses (same as above);
i. although linguistic theories are often used as devices for explaining interersting facts about a language, actual text in nontrivial domains contains a wide range of poorly understood and idiosyncratic phenomena, forcing any grammar to be used in a practical system to go beyond established results and requiring much effort in filling gaps in coverage;
j. hand-coding of heuristics is a labour-intensive task that is prone to mistakes and omissions, and makes system maintenance and enhancement more complicated and expensive;
k. using domain-specific hand-coded knowledge hinders the porting of a system to other domains or sublanguages.
As to i.-k., seen that j.-k. are given as a logical/natural consequence to the widely acceptable and shared assertion contained in i. it needs looking into the quite obvious fact that neither hand-crafted grammars nor statistical ones are exempt from being inherently language-dependent abstract representation of the linguistic structure of a specific language in a specific genre and domain. Statistically built will do that by tuning their grammar to a given training corpus with a certain number of caveats that we will discuss in detail below. Contrary to what is being assumed by Carroll, rule-based symbolic systems can take advantage of the generality of core grammars which as we will discuss furthen on in the paper offer core rules to be applied over an enormous gamut/range of natural languages, something which empirically built systems cannot take advantage of.
So it would seem that a lot of the current debate over the uselessness, inefficiency, inherent inability of rule-based symbolic systems is due to a fundamental choice in the type of parsing strategy and parsing algorithm, which as I understand it reflects Carroll's choice of constraint-based and unification-based formalisms. These formalisms supplanted ATNs and RTNs in the '80s and slowly came to fore of the linguistic audience supported by a number of linguistic theories, one of which, LFG is also at the heart of the system I will be presenting in this paper.
It seems to me that there hasn't been enough nerve in taking decisions which could counter the pervading feeling of the time when people working with mathematically sound and clean constraint-based unification-based parser discovered their inefficiency, and simply denounce that. The effort devoted to the construction of hand-built lexica and rules are useful nonetheless to the theoretical linguistic community and certainly to students. These algorithms are unfit for the parsing of real texts required by systems for Information Retrieval and the more ambitious Natural Language Understanding community.
In addition to this, HMMs based statistical parser also suffer from another drawback, pointed out by [9] in their paper where they take into account parameter estimation vs the use of conditional model structures:
“This paper separates conditional parameter estimation, which consistently raises test set accuracy on statistical NLP tasks, from conditional model structures, such as the conditional Markov model used for maximum-entropy tagging, which tend to lower accuracy. Error analysis on part-of-speech tagging shows that the actual tagging errors made by the conditionally structured model derive not only from label bias, but also from other ways in which the independence assumptions of the conditional model structure are unsuited to linguistic sequences…
The claim is that the independence assumptions embodied by the conditionally structured model were the primary root of the lower accuracy for this model. Label bias and observation bias are both explaining-away phenomena, and are both consequences of these assumptions. Explaining-away effects will be found quite generally in conditionally structured models, and should be carefully considered before such models are adopted.”[ibid:9]
In other words, the independence hypothesis which is at the heart of the use of Markov models imported from the speech community into the empirical statistical approaches to NLP does not seem to be well suited to the task at hand simply because the linguistic unit under consideration – the word, the single tag or both – are insufficient to assure enough contextual information, given language model building techniques based on word-tags with tagsets containing only lexical and part-of-speech information. It is precisely because of the independence hypothesis that people within the LinGO project have tried to map onto the tagset all layers of linguistic information, from morphological up to syntactic, semantic and possibly pragmatic information. However, this move will make it very hard for a corpus to have statistical significance seen that the number of occurrences required to cover most cases useful to serve any n-gram model will be certainly over tenths of million of words.

2. Linguistically-based Parsing and Linguistic Strategies

Shallow parsing is currently considered as the only viable solution to the problem of unlimited vocabulary text understanding, in general. This is usually accomplished by a sequence of cascaded partial syntactic analyzers, or chunkers, which are specialized to take care of particular linguistic structures - say, NPs, APs and PPs. The remaining part of the input text is either left unparsed or is passed on to another processor, and this is repeated in a cascade until clause level is reached.
Ambiguity is one of the main problems faced by large-scale computational grammars. Ambiguities can arise basically from Part Of Speech (POS) tagging associated to any given word of the input sentence. Natural languages, with narrow tagset of say 100 tags, will come up with an average amguity ratio of 1.7/1.8 per word: i.e. each input word can be assigned in average to two different tags. This base level ambiguity has to be multiplied by rule interactions, via alternative subcategorization frames related to governing lexical entries, or simply from linguistically motivated syntactic ambiguities. As opposed to human speakers, computational grammars are not yet able to always (100%) determine the contextually correct or intended syntactic analysis from a set of alternative analyses. Thus, a computational grammar based on unification and constraint based formalisms, and covering a realistic fragment of natural language will, for a given sentence, come up with a large number of possible analyses, most of which are not perceived by humans or are considered inappropriate in the given context. In order to reduce the number of possible analyses, local tag disambiguation is carried out on the basis of statistical and syntactic algorithms. For a sentence like,

(1) John wanted to leave

there should only be one analysis available due to local linguistic and statistically derived restrictions that prevent the word "to" from being interpreted as a preposition after the word "wanted" a verb and not a noun and be assigned to the category of complementizers or verb particles. So, even though on the basis of a bottom up analysis, a word like "leave" could be analysed both as a noun and as a base verb, by means of disambiguation carried out in a topdown fashion, the word "to" will trigger the appropriate interpretation of the word "leave" as base verb and not as noun. This is not always ensured, in particular in case a chart parser with a bottom up policy is chosen and all possible linguistic analaysis are generated in a parallel fashion.
Disambiguation should be carried out on a separate module, and not be conflated with parsing in case one wants to simulate Garden Paths while at the same time avoiding crashes or freezing of the parser to take place. This allows the topdown depth-first parser to backtrack and try the other analysis. However, backtracking should be allowed only whenever real Garden Path are in order. This kind of information is not hidden but can be derived from linguistic information.
In this paper we will discuss our proposal to solve ambiguity by means of linguistically related lexical and structural information which is used efficiently in a number of disambiguation strategies. Since the parser we will present is a multilingual parser, strategies will be also related to UG parameters in order to take advantage of the same Core Grammar and use Peripheral Rules for that task.

2.1 Shallow and Deep Parsing

The shallow parsing approach is very efficient and usually prevents failures. However the tradeoff with deep parsing is a certain percentage of text not being fully parsed due to local failures, especially at clause level. This may also result as a wrong choice of tag disambiguation, which carries on to constituent level. Another important shortcoming is the inherent inability of this approach to ensure a semantically consistent mapping of all resulting constituent structures. This is partly due to the fact that clause level analysis is only approximated and not always fully realized. In addition, all attachments are also approximated in lack of a stable clause level analysis. Eventually, also subcategorization frames cannot be used consistently but only tentatively matched with the available information.
As a counterpart to this situation, shallow parsers can easily be ported to other languages and so satisfy an important requirement, that of reusability. In theoretical linguistic terms, this concept is easily understood as a subdivision of tasks between the parameters and principles components vs the rule component which being universal, relies on X-bar based constituency.
Though X-bar based parsing may be inefficient, one way to improve it would be that of encoding lexical ambiguity, both at word level and at the ensuing structural level. We would like to assume that specialization in language dependent ambiguity resolution is one of the components of the language acquisition process.
The lexicon as the source of syntactic variation is widely accepted in various theoretical frameworks (see Bresnan, in press). We assume that be it shallow or deep, parsing needs to be internally parameterized in oder to account for ambiguities generated both at structural and at semantic level.
The parser we present has been built to simulate the cognitive processes underlying the grammar of a language in use by a speaker, taking into account the psychological nuances related to the wellknown problem of ambiguity, which is a pervading problem in real text/life situation, and it is regarded an

inseparable benchmark of any serious parser of any language to cope with.

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.

In other words, looking at parsing from a performance-based perspective, to justify speakers' psycholinguistic behaviour and its simulation in a running parser, we think it should be organized as a topdown depth-first symbolic rule compiler ordered according to efficiency criteria and using Lookahead and a Well-Formed Substring Table (WFST) not to duplicate effort.

This is just the opposite of a Unification Grammar which uses Chart parsing in a bottom up breadth-first manner which is norm in Constraint-Based formalisms like HPSG or LFG. However, what's more important, the parser should know what kind of amguities could cause unwanted Garden-Paths and Crashes, to refrain from unwanted failures in order to mimick human processing. Constraint unification is in our opinion unable to satisfy the efficiency requirements and prevent unwanted failures: it is insufficient to simply have a list of lexical items with their features, and a grammar with a list of rules which obey to a certain number of principles and constraints. A "sound" parser needs to be told which ambiguous structures are expected in which language.

In general terms, ambiguity is generated by homophonous words in understanding activities and by homographs in reading activities. In both cases Garden Paths or Crashes may only result in a given language in presence of additional conditions which are strictly dependent on the structure of the lexicon and the grammar. But some UG related parameters, like the "OMISSIBILITY OF THE COMPLEMENTIZER" in English may cause the parser to crash or freeze. Generally speaking, all types of ambiguity affecting parsing at a clause level will cause the parser to go into a Garden Path. The typical example quoted in psycholinguistic literature is the reduced relative case, reported here below, determined by the lexical ambiguity of English verbs being at the same time interpretable as Past Participle - Past Tense and shown below in the Reduced Relative Clause well-known example,

(2) The horse raced past the barn fell.

is one such case. The English speaker will attempt treating the verb "raced" as main tensed verb, but on discovery of sentence final verb "fell" which can only be interpreted as tensed past tense the whole sentential level analysis crashes and a Garden Path ensues causing a complete restart of the mental parser.

We assume that from a psycholinguistic point of view, parsing requires setting up a number of disambiguating strategies, basically to tell arguments apart from adjuncts and reduce the effects of backtracking.

The system is based on LFG theoretical framework (see Bresnan, [16]) and has a highly interconnected modular structure. It is a top-down depth-first DCG-based parser written in Prolog which uses a strong deterministic policy by means of a lookahead mechanism with a WFST to help recovery when failure is unavoidable due to strong attachment ambiguity.

It is divided up into a pipeline of sequential but independent modules which realize the subdivision of a parsing scheme as proposed in LFG theory where a c-structure is built before the f-structure can be projected by unification into a DAG. In this sense we try to apply in a given sequence phrase-structure rules as they are ordered in the grammar: whenever a syntactic constituent is successfully built, it is checked for semantic consistency, both internally for head-spec agreement, and externally, in case of a non-substantial head like a preposition dominates the lower NP constituent; other important local semantic consistency checks are performed with modifiers like attributive and predicative adjuncts. In case the governing predicate expects obligatory arguments to be lexically realized they will be searched and checked for uniqueness and coherence as LFG grammaticality principles require.

Whenever a given predicate has expectancies for a given argument to be realized either optionally or obligatorily this information will be passed below to the recursive portion of the parsing: this operation allows us to implement parsing strategies like Minimal Attachment, Functional Preference and other ones (see Delmonte and Dolci 1989; Delmonte and Dolci 1997).

As to multilinguality, the basic tenet of the parser is based on a UG-like perspective, i.e. the fact that all languages share a common core grammar and may vary at the periphery: internal differences are predicted by parameters. The DCG grammar allows the specification of linguistic rules in a highly declarative mode: it works topdown and by making a heavy use of

linguistic knowledge may achieve an almost complete deterministic policy. Parameterized rules are scattered throughout the grammar so that they can be made operative as soon as a given rule is entered by the parser.

In particular, a rule may belong either to a set of languages, e.g. Romance or Germanic, or to a subset thereof, like English or Italian, thus becoming a peripheral rule. Rules are activated at startup and whenever a switch is being operated by the user, by means of logical flags appropriately inserted in the right hand side of the rule. No flags are required for rules belonging to the common core grammar.

Some such rules include the following ones: for languages like Italian and Spanish, a Subject NP may be an empty category, either a referential little pro or an expletive pronoun; Subject NPs may be freely inverted in postverbal position, i.e. preverbal NP is an empty category in these cases. For languages like Italian and French, PP or adverbial adjuncts may intervene between Verb and Object NP; adjectival modifiers may be taken to the right of their head Noun. For languages like English and German, tense and mood may be computed in CP internal position, when taking the auxiliary or the modal verb. English allows an empty Complementizer for finite complement and relative clauses, and negation requires do-support. Italian only allows for a highly genre marked (literary style) untensed auxiliary in Comp position.

Syntactic and semantic information is accessed and used as soon as possible: in particular, both categorial and subcategorization information attached to predicates in the lexicon is extracted as soon as the main predicate is processed, be it adjective, noun or verb, and is used to subsequently restrict the number of possible structures to be built. Adjuncts are computed by semantic compatibility tests on the basis of selectional restrictions of main predicates and adjuncts heads.

Syntactic rules are built using CP-IP functional maximal projections. Thus, we build and process syntactic phenomena like wh- movement before building f-structure representations, where quantifier raising and anaphoric binding for pronominals takes place. In particular, all levels of Control mechanisms which allow coindexing at different levels of parsing give us a powerful insight into the way in which the parser should be organized.

Yet the grammar formalism implemented in our system is not fully compliant with the one suggested by LFG 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. On the other hand, DCGs being based on Prolog language, give 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. In addition, we find that topdown parsing policies are better suited to implement parsing strategies that are essential in order to cope with attachment ambiguities (but see below). We use 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 chosen languages. 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.

Being a DCG, the parser is strictly a top-down, depth-first, one-stage parser with backtracking: differently from most principle-based parsers presented in Berwick et al.(1991), which are two-stage parsers, our parser computes its representations in one pass. This makes it psychologically more realistic. The final output of the parsing process is an f-structure which serves as input to the binding module and logical form: in other words, it constitutes the input to the semantic component to compute logical relations. In turn the binding module may add information as to pronominal elements present in the structure by assigning a controller/binder in case it is available, or else the pronominal expression will be available for discourse level anaphora resolution. As to the most important features of DCGs, we shall quote from Pereira and Warren(1980) conclusions, in a comparison with ATNs:

> "Considered as practical tools for implementing language analysers, DCGs are in a real sense more powerful than ATNs, since, in a DCG, the structure returned from the analysis of a phrase may depend on items which have not yet been encountered in the course of parsing a sentence. ... Also on the practical side, the greater clarity and modularity of DCGs is a vital aid in the actual development of systems of the size and complexity necessary for real natural language analysis. Because the DCG consists of small independent rules with a declarative reading, it is much

> easier to extend the system with new linguistic constructions, or to modify the kind of structures which are built. ... Finally, on the philosophical side, DCGs are significant because they potentially provide a common formalism for theoretical work and for writing efficient natural language systems."(ibid,278).

2.2 Disambiguating constituency with functional mapping

As shown in Fig.1 below, the parser is made up of separate modules:

1. The Grammar, based on DCGs, incorporates Extraposition to process Long Distance Dependencies, which works on annotated c-structures: these constitute the output to the Interpretation Module;
2. The Interpretation Module checks whether f-structures may be associated to the input partially annotated c-structure by computing Functional Uniqueness, Coherence and Completeness. Semantic roles are associated to the input grammatical function labels at this level, after semantic selectional restrictions are checked for membership;
3. The Mapping scheme, to translate trees into graphs, i.e. to map c-strutures onto f-structures. The parser builds annotated c-structure, where the words of the input sentence are assigned syntactic constituency and functional annotations. This is then mapped onto f-structure, i.e. constituent information is dropped and DAGs are built in order to produce f-structure configuration.

Mapping into f-structure is a one-to-many operation: each major constituents may be associated with different functional values: this is why we activate grammatical function calls whenever possible in order to take into account the position of constituents to be built by the parser. This is particularly true for NPs, but can also be applied to other constituents as can be seen from the following discussion on constituent-grammatical function mapping:

a. NP --> SUBJect, both in preverbal and postverbal position - VP internally, VP adjoined and IP adjoined (see Delmonte, 1987) - with any kind of verbal category; OBJect, usually in VP internal position, but also in preverbal position at Spec CP in case of reversed transitive structures; NCOMP predicative function - if not proper noun - occuring with copulative, and ECM verbs like "consider, believe"; closed ADJunct with [temporal] value, as the corresponding English example "this morning", which however in Italian can be freely inserted in sentence structure;

b. AP --> Modifier of an NP head, occurring as attribute in prenominal and as predication in postnominal position; ACOMP predicative function occurring with copulative, and ECM verbs; open XADJunct occurring freely at sentence level. Other examples of open adjuncts are: floating quantifiers, which however may only occur VP internally; doubling emphatic pronoun "lui" which also occurs VP internally and is computed as open adjunct;

c. AdvP --> Open or closed Adjuncts according to its selectional properties, occurring anywhere in the sentence according to their semantic nature;

d. PP --> OBLiques, when selected by a given predicate; PCOMP predicative function, when selected by a given predicate - both these two types of argument usually occur VP internally but may be fronted; open XADJunct or closed ADJunct according to semantic compatibility checks;

e. VP' --> VCOMP infinitivals, when selected by a given predicate; SUBJect propositional clauses; closed ADJuncts with semantic markers like "for"; VP' gerundive and participial, which are always computed respectively as closed ADJuncts the former and as open ADJuncts the latter;

f. S' -->or CP as main clauses, or subordinate clauses, as well as sentential complements and SUBJect propositional clauses;

g. Clitics and Pronominal elements are also computed as Nps or PPs, because they are assigned grammatical functions when not associated to NP dislocation in preverbal position: in that case, the clitic is simply erased and TOPic function is associated with the binder NP.

2.3 Tracing c-structure rules

The parser looks for syntactic constituents adjoined at CP level: in case of failure, it calls for IP level constituents, including the SUBJect which may either be a clause or an NP. This is repeated until it reaches the Verbal Phrase: from that moment onward, the syntactic category associated to the main verb - transitive, unergative, unaccusative, impersonal, atmospheric, raising, psych, copulative - and the lexical form of the predicate, are both used as topdown guidelines for the surface realization of its arguments. Italian is a language which allows for empty or morphologically unexpressed Subjects, so that no restriction may be projected from the lexicon onto c-structure: in case it is empty, a little pro is built in subject position, and features are left as empty variables until the tensed verb is processed.

The grammar is equipped with a lexicon containing a list of fully specified inflected word forms where each entry is followed by its lemma and a list of

morphological features, organized in the form of attribute-value pairs. However, morphological analyzers for Italian and English are also available with big root dictionaries (90,000 for Italian, 25,000 for English) which only provide for syntactic subcategorization, though. The fully specified lexicon has been developed for Italian, English and German and contains approximately 5,000 entries for each language. In addition to that there are all lexical form provided by a fully revised version of COMLEX, and in order to take into account phrasal and adverbial verbal compound forms, we also use lexical entries made available by UPenn and TAG encoding. Their grammatical verbal syntactic codes have then been adapted to our formalism and is used to generate an approximate subcategorization scheme with an approximate aspectual and semantic class associated to it. Semantic inherent features for Out of Vocabulary Words , be they nouns, verbs, adjectives or adverbs, are provided by a fully revised version of WordNet in which we used labels similar to those provided by CoreLex.

Once the word has been recognized, lemmata are recovered by the parser in order to make available the lexical form associated to each predicate. 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.

Grammatical functions are used to build f-structures and the processing of pronominals. They are crucial in defining lexical control: as in Bresnan (1982), all predicative or open functions are assigned a controller, lexically or structurally. Lexical control is directly encoded in each predicate-argument structure, and in case shallow parsing does not make that information available it will be impossible for the parser to bind the empty subject of all predicative open functions built in all predicative structures (or small clauses) to the approriate syntactic controller (or binder).

As said above, we think it highly important to organize c-structure rules for sentence level representation by means of the introduction of functional major constituents at the following basic levels:

```
CP -->  Spec, C'
C'  -->  C, IP
IP  -->  Spec=NP(subject),  I'
I'   -->  Inflected Tensed Verb Form, VP.
```

According to this configuration, adjuncts and constituents like wh- words for questions and topicalized NPs, adjoined at sentence level, will be computed at first in a CP constituent and then passed down to the lower level of analysis. This organization of constituency allows for complementizers, i.e. the head of CP, to be kept separate in C' level so that a nice interaction may be possible, if needed.

Here below we list some of the higher rules of the grammar with one of the interpretation rules for copulative constructions:

Tab. 1 Some higher level rules of the parser

```
utterance          --> assertion_direct
utterance          --> standard_utterance
standard_utterance-->  wh_question
standard_utterance-->  yes_no_question
standard_utterance-->  assert_cp
assert_cp-->       aux_to_comp
                   adjunct_cp
                   i_double_bar
assert_cp-->       object
                   adjunct_cp
                   pro=SI
                   verb_phrase_impersonal
assert_cp-->       object
                   adjunct_cp
                   negat
                   pro=CLI, {Case=acc}
                   verb_phrase_focalized
assert_cp-->       object
                  adjunct_cp
                   i_double_bar
i_double_bar-->           subject
                   negat
                   adjs_preverbal
                  parenthetical
```

```
                    i_one_bar
i_one_bar--> verb_phrase_pass_canonic
i_one_bar--> clitics,
                { germanic_aux,
                  clitics,
                  adjs_post_aux,
                  germanic_vp  ;
                 all_languages_vp }
verb_phrase_copulative-->  adv_phrase
                           check_clitic_object
                           xcomp
                           prepositional_phrases
interpret_copulative:-
          lexical-form& predicate-argument_structure
            interpret_subject
            interpret_xcomp
            assign_control_xcomp
            interpret_adjuncts
```

Notice that i_one_bar rewrites as passive VP and in case of failure as active VP: again this is required by the need to activate the appropriate interpretation rule for transitive verb which in most languages is morphologically determined by the presence of the appropriate auxiliary/ies and the past participle of the main verb.

2.4 Elliptical Structures

In a framework like this, all elliptical structures are left over at the end of grammar traversal, simply because they cannot possibly be computed as any of the grammatically complete sentence level analyses, either as main clauses , as complement clauses or as subordinate or coordinate clauses. Just consider a simple case like the following:

(3) The nights must be cold and the days warm

which has been from a test text for NLUnderstanding distributed by Mitre. In order to compute the vp-ellipsis the rest of the previous computation constituted by the string,

3.1 [and, the, days, warm]

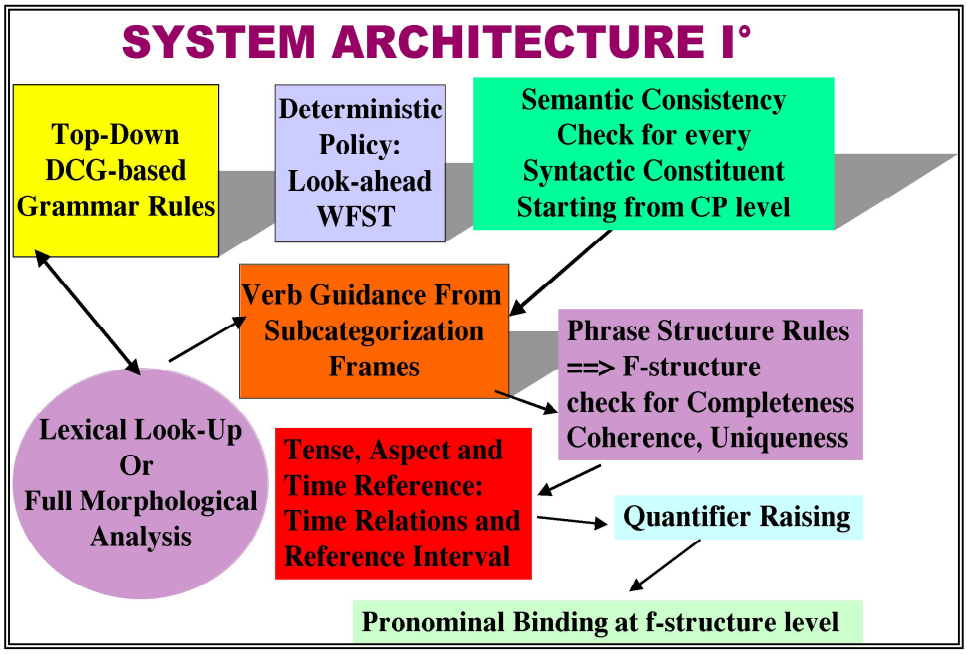
Fig.1 GETARUN Parser Architecture

must be evaluated in relation to the overall input sentence which is available from the lookahead stack. This is done in order to check for some parallel pattern at the level of tag assignment. This is also done in order to certify failures due to some form of agramaticallity present in the input sentence. When the parallelism has been ascertained, the main clause is used in order to duplicate the governing elliptical verbal predicate and the rest of the sentence is parser in its component constituents. This is done by accessing an iterative call which is being used by all complements whenever a transitive verb has been detected or simply whenever there is not enough information to decide on verb subcategorization frame. The resulting list of constituents is then interpreted as in any normal non elliptical sentence by adding all verb related syntactic and semantic information which is lacking in elliptical sentences. The output will be a coordinate sentential structure which has the same verb information as the main preceding clause.

The call to recover from failures with elliptical structures is also used in case of ungrammatical structures with a feedback message being generated on the basis of the words still to be processed. In one case we manage to recover from failure due to ambiguously computed constituents which however do not motivate any preliminary choice either from the tag disambiguation procedure or from parsing strategy. These are cases of adjunct PP or similar constructions which do not depend on lexical information for their interpretation. One example is a case of parasitic gap construction like the following,

(4) This is the kind of food that must be cooked before Mary eats.

In this example, “before Mary” will be computed as a PP which will then be appropriately interpreted as adjunct to the main verb “cook”. So the ending word “eats” will be left over to be filtered by the rule for elliptical constructions. This will be the trigger to recover the wrong analysis.



The shallow parser which runs in parallel with the deep parser, is built as shown in Fig.3 below, with standard components like a statistical/syntactic tagger and a cascaded shallow parser which in a final run turns syntactic constituents into functionally labelled arguments/adjuncts. Subcategorization information is derived from COMLEX as well as from the subcategorized lexicon made available by Upenn. Semantic information is encoded in the 200,000 entries semantic lexicon built on the basis of EuroWordnet with a number of additions coming from computer, economics, and advertising semantic fields. Semantic class encoding has followed Corelex close semantic set labeling with 60 semantic labels.

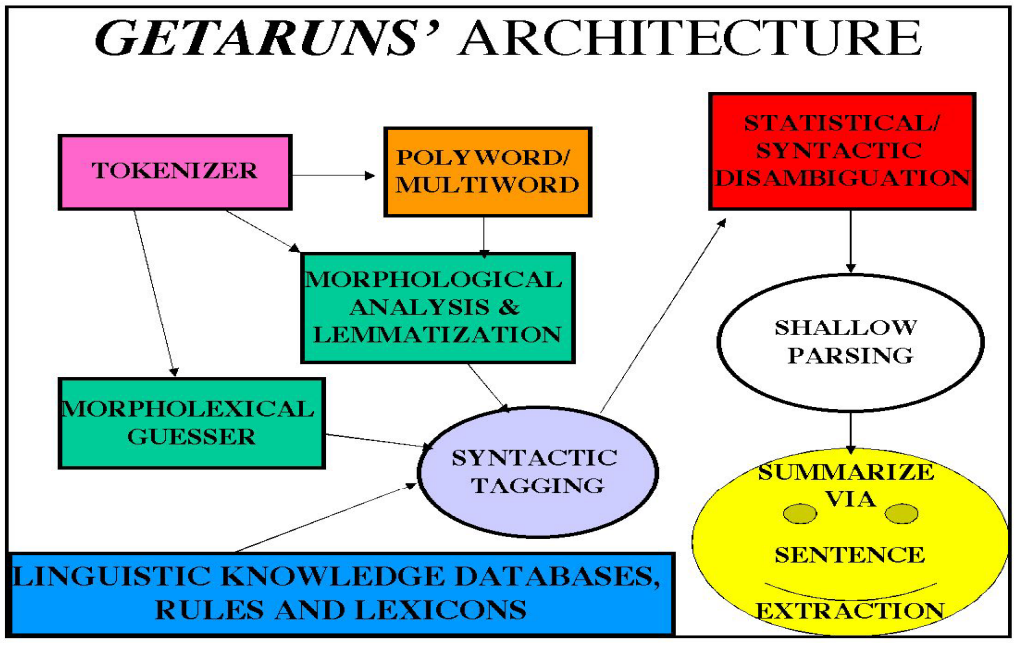


Fig.2 Shallow GETARUNS Architecture

2.4 Parameters and Strategies

Here below is a short list of parameterized ambiguities: some of them are to be solved by parsing preferences which according to J.Fodor's latest work,

are typological in nature. It appears that speakers of English prefer to adopt a Minimal Attachment strategy while this is not so per speakers of Romance languages. In particular, in the case of Relative Clause Attachment, this might be related to the influence of Latin on Romance language: Italian allows relative clauses as indipendent sentences to be attached in the discourse, just like Latin does.

A. Omissibility of Complementizer
o NP vs S complement
o S complement vs relative clause
B. Different levels of attachment for Adjuncts
o VP vs NP attachment of pp
o Low vs high attachment of relative clause
C. Alternation of Lexical Forms
o NP complement vs main clause subject
D. Ambiguity at the level of lexical category
o Main clause vs reduced relative clause
o NP vs S conjunction
E. Ambiguities due to language specific structural proprieties
o Preposition stranding
o Double Object
o Prenominal Modifiers
o Demonstrative-Complementizer Ambiguity
o Personal vs Possessive Pronoun
F. Clitic Pronouns
G. Aux-to-Comp

2.4.1 Linguistically-plausible relaxation techniques

With the grammar listed above and the parameters we are now in a position to establish apriori positions in the parser where there could be recovery out of loop with ungrammatical structures with the possibility to indicate which portion of the input sentence is responsible for the failure. At the same time, parsing strategies could be devised in such a way to ensure recovery from local failure. We will start by commenting on Parsing Strategies first and their implementation in our grammar.

The parser has been built to simulate the cognitive processes underlying the grammar of a language in use by a speaker, taking into account the psychological nuances related to the wellknown problem of ambiguity, which is a pervading problem in real text/communicative situation, and it is regarded an inseparable benchmark of any serious parser of any language to cope with.

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 (see Pitchett.

Differently from what is asserted by global or full paths approaches (see Schubert, 1984; Hobbs et al., 1992), 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. 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 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.

3. 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 1million 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 2. below,

Tab. 2 Shallow Parsing & Statistical Approaches
(Data from D.Hindle & M.Roth, Structural Ambiguity and Lexical Relations)

Structural Types	Nos. Preps.	% Preps.	Tot. Preps.
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**	

As the data reported above clearly show, most of the prepositional phrases are constituted by arguments of Noun, rather than of Verb. As to the remaining data, adjuncts are represented approximately by the same amount of cases, 11% of the sample text.
At first we collected all information on prepositions as a whole and then we searched into our Treebank looking for their relations as encoded in syntactic constituent structure. Here below we report data related to prepositions for the whole corpus: notice that in Italian as in English, preposition "of"/di would be used mainly as a Noun argument/modifier PP.

Tab. 3.Shallow Parsing & Statistical Approaches

Venice Italian Corpus
1 million tokens
All prepositions - 54 different types or wordforms: 170,000 occurrences
Argument-like prepositions **71.1 %**

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

Adjunct-like prepositions **23.2%**

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:

- The number of prepositional phrases in Italian texts is four times bigger than 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 big difference between the two languages can be traced back in the behaviour of PP noun arguments, which in turn can be traced back to a language specific typalogical difference: the existence of prenominal modifiers in English and not in Italian – or at least, not yet substituted by the use of postnominal modification.

All PPs & Types of Adjs. Vs. Args.	Tot. Preps.	Tot. Types	% Types	% Preps
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%**	

Tab. 4 Quantitative Syntactic and Semantic Distribution of PPs in VIC

3.1 Two mechanisms at work

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 the lookahead stack, which is 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. The following list of preterminal 14 symbols is used:

Tab. 5 Preterminal symbols used for lookahead

1. v=verb-auxiliary-modal-clitic-cliticized verb
2. n=noun – common, proper;
3. c=complementizer
4. s=subordinator;
5. e=conjunction
6. p=preposition-particle
7. a=adjective;
8. q=participle/gerund
9. i=interjection
10. g=negation
11. d=article-quantifier-number-intensifier-focalizer
12. r=pronoun
13. b=adverb
14. x=punctuation

As has been reported in the literature (see Tapanainen, 1994; Brants, 1995), English is a language with a high level of homography: readings per word are around 2 (i.e. each word can be assigned in average two different tags). Lookahead in our system copes with most cases of ambiguity: however, we also had to introduce some disambiguating tool before the input string could be safely passed to the parser. Disambiguation is applied to the lookahead stack and is operated by means of Finite State Automata. The reason why we use FSA is simply due to the fact that for some important categories, English has unambiguous tags which can be used as anchoring in the input string, to reduce ambiguity. I'am now referring to the class of determiners which is used to tell apart words belonging to the ambiguity class [verb,noun], the most frequent in occurrence in English.

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. Any already parsed NP/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 contents of the structure in a table lookup storage which is then accessed whenever there is an attempt on the part of the parser to build up a similar constituent. In order to match the input string with the content of the stored 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. 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 the 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.

Lookahead is used in a number of different ways: it may impose a wait-and-see policy on the topdown strategy or it may prevent following a certain rule path in case the stack does not support the first or even second match:

a. to prevent expanding a certain rule

b. to prevent backtracking from taking place by delaying retracting symbols from input stack until there is a high degree of confidence in the analysis of the current input string.

It can be used to gather positive or negative evidence about the presence of a certain symbol ahead: symbols to be tested against the input string may be more than one, and also the input word may be ambiguous among a number of symbols. Since in some cases we

extend the lookahead mechanism to include two symbols and in one case even three symbols, possibilities become quite numerous.
Consider now failure and backtracking which ensues from it. Technically speaking, by means of lookahead we prevent local failures in that we do not allow the parser to access the lexicon where the input symbol would be matched against. It is also important to say that almost all our rules satisfy the efficiency requirement to have a preterminal in first position in their right-hand side. This is usually related to the property belonging to the class of Regular Languages. There are in fact some wellknown exceptions: simple declarative sentence rule, yes-no questions in Italian. Noun phrase main constituents have a multiple symbols lookahead, adjectival phrase has a double symbol lookahead, adverbial phrase has some special cases which require the match with a certain word/words like "time/times" for instance. Prepositional phrase requires a single symbol lookahead; relative clauses, interrogative clauses, complement clauses are all started by one or more symbols. Cases like complementizerless sentential complements are allowed to be analysed whenever a certain switch is activated.
Suppose we may now delimit failure to the general case that may be described as follows:
- a constituent has been fully built and interpreted but it is not appropriate for that level of attachment:
failure would thus be caused only by semantic compatibility tests required for modifiers and adjuncts or lack of satisfaction of argument requirements for a given predicate.
Technically speaking we have two main possibilities:
A. the constituent built is displaced on a higher level after closing the one in which it was momentarily embedded.
This is the case represented by the adjunct PP "in the night" in example 16 that we repeat here below:
(5) The thieves stole the painting in the night.
The PP is at first analysed while building the NP "the painting in the night" which however is rejected after the PP semantic features are matched against the features of the governing head "painting". The PP is subsequently stored on the constituent storage (the WFST) and recovered at the VP level where it is taken as an adjunct.
B. the constituent built is needed on a lower level and there is no information on the attachment site.
In this case a lot of input string has already been consumed before failure takes place and the parser needs to backtrack a lot before constituents may be safely built and interpreted.
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 WFST 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.
As said above, the lookahead procedure is used both in presence and in absence of certain local requirements for preterminals, but always to confirm the current choice and prevent backtracking from taking place. As a general rule, one symbol is sufficient to take the right decision; however in some cases, more than one symbol is needed. In particular when building a NP, the head noun is taken at first by nominal premodifiers, which might precede the actual head noun of the NP. The procedure checks for the presence of a sequence of at least two nouns before consuming the current input token. In other cases the number of preterminals to be checked is three, and there is no way to apply a wait-and-see policy.
Reanalysis of a clause results in a Garden Path(GP) in our parser because nothing is available to recover a failure that encompasses clause level reconstruction: we assume that GP obliges the human processor to dummify all naturally available parsing mechanisms, like for instance lookahead, 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. The same applies to our case which involves interaction between two separate modules of the grammar.
As an example, consider processing time 3.8 secs with strategies and all mechanisms described above

activated, as compared to the same parse when the same are disactivated – 6.5 secs, in relation to the following highly ambiguous example taken from a legal text:
(6) Producer means the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part and any person who by putting his name, trade mark or other distinguishing feature on the product presents himself as its producer.
Computation time is calculated on a Macintosh G4.
In more detail, suppose we have to use the information that "put" is a verb which requires an oblique PP be present lexically in the structure, as results from a check in its lexical form. We take the verb in I position and then open the VP complement structure, which at first builds a NP in coincidence with "the book". However, while still in the NP structure rules, after the head has been taken, a PP is an option freely available as adjunct.
We have implemented two lookahead based mechanisms which are used in the PP building rule and are always triggered, be it from a position where we have a noun as head and we already built part of the corresponding constituent structure; be it from a position where we have a verb as head and we want to decide whether our PP will be adequate as argument rather than as adjunct - in the latter case it will become part of the Adjunct Set.
The first one is called,
- Cross Compatibility Check (CCC)
This mechanism requires the head semantic features or inherent features to be checked against the preposition, which in turn activates a number of possible semantic roles for which it constitutes an adequate semantic marker. For instance, the preposition "on" is an adequate semantic marker for "locative" semantic role, this will cause the compatibility check to require the presence in the governing heading of inherent or semantic features that allow for location. A predicate like "dress" is computed as an object which can be assigned a spatial location, on the contrary a predicate like "want" is computed as a subjective intensional predicate which does not require a spatial location. However, in order to take the right decision, the CCC must be equipped with the second mechanism we implemented;
The second one is called,
- Argument Precedence (AP)
The second mechanism we implemented allows the parser to satisfy the subcategorization requirements in any NP constituent it finds itself at a given moment if the parsing process. Suppose that after taking "put" as the main verb, this mechanism is activated, by simply copying the requirements on PP oblique locative present in the lexical form associated with the predicate "put" in the lexicon, in the AP. As soon as the NP "the book" is opened, after taking "book" as N at the head position, the parser will meet the word "on", which allows for a PP adjunct. While in the P head position, the parser will fire the CCC mechanism first to see whether the preposition is semantically compatible, and in case it is, the second AP mechanism will be fired. This will cause the system to do the following steps:
i. check whether the requirements are empty or not;
ii. and in case it is instantiated, to control the semantic role associated with it;
iii. to verify whether the P head is a possible semantic marker for that semantic role: in our case, "on" is a possible semantic marker for "locative" semantic role;
iv. finally to cause the parser to fail on P as head of a PP adjunct of the head noun;
v. produce a closure of NP which obeys Minimal Attachment principle.

3.2 Some examples

In the texts reported below we give empirical evidence for the need to use lexical information in order to reduce parsing loads resulting from backtracking procedures: we mark decision points with a bar,
(7) Council directive | **of** july 1985 | **on** the approximation | **of** the laws, | regulations and | administrative provisions | **of** the Member States | concerning liability | **for** defective products.
At the first boundary we have "of" which is non semantically marked and no prediction is available, so that the default decision is to apply Late Closure, which turns out to be the correct one. When the second preposition is found we are in the NP of the PP headed by "of", and we have taken the date "1985": this will cause the CCC to prevent the acceptance of the preposition "on" as a semantically compatible marker thus preventing the construction of the NP headed by "approximation".
Notice, that in case that would be allowed, the NP would encompass all the following PPs thus building a very heavy NP: "the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products". In case the parser had a structure monitoring strategy all this work would have to be undone and backtracking would have to be performed. Remember that the system does not possibly know where and how to end backtracking unless by trying all possible available combination along the path. In our case, the presence of a coordinate structure would render the overall process of structure recoverability absolutely untenable.
Another important decision has be taken at the boundary constituted by the participial head "concerning": in this case the CCC will take the inherent features of the head "States" and check them with the selection restrictions associated in the lexical form for the verb "concern". Failure in this match will cause the NP "the Member States" to be closed and will allow the adjunct to be attached higher up with the coordinated head "laws, regulations and administrative provisions". In this case, all the

inherent features are collected in a set that subsumes them all and can be used to fire CCC.
Notice that the preposition "for" is lexically restricted in our representation for the noun "liability", and the corresponding PP that "for" heads interpreted as a complement rather than as an adjunct. We include here below the relevant portion of each utterance in which the two mechanisms we proposed can be usefully seen at work. We marked with a slash the place in the input text in which, usually when the current constituent is a NP a decision must be taken as to whether causing the parser to close (MA) or to accept more text (LC) is actually dependent upon the presence of some local trigger. This trigger is mostly a preposition; however, there are cases in which, see (11), (12), (14)., (15), the trigger is a conjunction or a participle introducing a reduced relative clause. Coordinate NPs are a big source of indecision and are very hard to be detected if based solely on syntactic, lexical and semantic information. For instance, e. can be thus disambiguated, but h. requires a matching of prepositions; In the case represented by (15) we put a boundary just before a comma: in case the following NP "the Member State" is computed as a coordination - which is both semantically, syntactically and lexically possible, the following sentence will be deprived of its lexical SUBJect NP - in this case, the grammar activates a monitoring procedure independently so that backtracking will ensue, the coordinate NP destroyed and the comma computed as part of the embedded parenthetical (which is in turn an hypothetical within a subordinate clause!!). Notice also that a decision must be taken in relation to the absolutives headed by a past participle which can be intended as an active or a passive past participle: in the second case the head noun would have to be computed as an OBJect and not as a SUBJect. The following examples are small fragment from biggest sentences which are use to enforce our point:
(8) a differing degree of protection of the consumer | **against** damage caused by a defective product | **to** his health or property
(9) in all member states | **by** adequate special rules, it has been possible to exclude damage of this type | **from** the scope of this directive
(10) to claim full compensation for the damage | **from** any one of them
(11) the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part | **and** any person
(12) The liability of the producer | **arising** from this directive
(13) any person who imports into the community a product | **for** sale, hire or any form of distribution | **in** the course of his business
(14) both by a defect in the product | **and** by the fault of the injured person
(15) However, if... the commission does not advise the Member State | **concerned** that it intends submitting such a proposal | **to** the council | , the Member State

3.3 Principles of Sound Parsing

o Principle One: Do not perform any unnecessary action that may overload the parsing process: follow the Strategy of Minimal Attachment;
o Principle Two: Consume input string in accordance with look-ahead suggestions and analyse incoming material obeying the Strategy Argument Preference;
o 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;
o 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;
o Principle Five: Erase short-memory stack as soon as possible, i.e. whenever clausal constituents receive Full Interpretation.
oStrategy 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.
oStrategy Minimal Attachment: whenever Functional Preference allows it apply a Minimal Attachment Strategy.
The results derived from the application of Principle Four are 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 can be, and that in order to interpret major constituents as arguments of some predicate we need to have 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.

3.4 Graceful Recovery Actions from Failures

As we discussed above, recovery from garden-path requires a trial and error procedure, i.e. the parser at first has to fail in order to simulate the garden-path effect and then the recovery will take place at certain conditions.

Now consider the well-known case of Reduced Relatives which have always been treated as a tough case (but see Stevenson & Merlo[15]). From an empirical point of view we should at first distinguish cases of subject attachement reduced relatives from all other cases, because it is only with subject attachement that a garden-path will actually ensue. This is easily controllable in our parser given the fact that NPs are computed by means of functional calls. In this way the information as to where the NP is situated in the current sentence analysis is simply a variable that is filled with one of the following labels: subj, obj, obj2, obl, adj, ncomp, where the last label stands for predicative open complements. Again from a purely empirical point of view, we also visited the WSJ corpus in order to detect cases of subject attachment vs all other cases for reduced relatives and we came up with the following figures:

SUBJECT-ATTACHEMENT 530
OTHERS 2982
Total 3512

From the total number we must subtract present participle cases of reduced relatives which do not constitute ambiguous words: the total number is lowered down to 340. Subject-attachment thus constitute the 9.68% of all cases, a certainly negligible percentage. In addition, 214 of all subject-attachment are passive participles and lend themselves to easy computation being followed by the preposition "by". So there will reasonably be only 116 possible candidates for ambiguous reduced relatives. The final percentage comes down 3.3% which is very low in general, and in particular when computed over the whole 1 million occurences, it comes down to a non classifiable 0.0116%. The same results can be obtained from an investigation of the Susanne Corpus, where we found 38 overall cases of reduced relatives with ambiguous past participles, 0.031% which is comparable to the 0.035% of the WSJ.

If we look into matter closely, then we come up with another fairly sensible and easily intuitive notion for reduced relatives disambiguation: and it is the fact that whenever the governing Noun is not an agentive, nor a proto-agent in any sense of the definition (see [15]), no ambiguity may arise simply because non agentive nominal governors may end up with an ambiguous interpretation only in case the verb is used as ergative. However, not all transitive verbs can be made ergatives and in particular none of the verbs used in WSJ in subject-attachment for reduced relatives can be ergativized apart from "sell". We report here below verb-types, i.e. verb wordforms taken only once. As can be easily seen none of the verbs are unergative nor unaccusatives.

Tab. 6 List of 27 verb-types used in WSJ in subject-attached reduced relatives

accused	afforded	based	boosted
bought	canceled	caught	caused
completed	contacted	derived	designed
filed	honed	involved	led
listed	made	managed	owned
paid	purchased	related	represented
requested	***sold***	unsettled	

If we look at the list of 36 verb-types used in Susanne Corpus we come up with a slightly different and much richer picture:

Tab. 7 List of 36 verb-types used in SUSANNE in subject-attached reduced relatives

altered	become	***bent***	***burned***
charged	clouded	compared	***cooled***
cut	deserted	distilled	***dominated***
estimated	fed	figured	filmed
focused	frozen	internalized	intertwined
known	left	made	***opened***
posted	proposed	puckered	put
removed	reported	seen	shown
shut	soiled	studied	torn

The number of ergativazable verbs increases and also the number of verb types which is strangely enough much higher than the one present in WSJ. We also underlined verbs that can be intransitivized, thus contributing some additional ambiguity. In some cases, the past participle is non ambiguous, though, see "frozen, seen, shown and torn". In some other cases, the verb has different meanings with different subcategorization frames: this is case of "left".

In any case, the parser will procede by activating any possible disambiguation procedure, then it will consider the inherent semantic features associated to the prospective subject: in order to be consistent with a semantic classification as proto-agent, one of the following semantic classes will have to be present: "animate, human, institution, (natural) event, social_role, collective entity".

In the affirmative case, and after having checked for the subject position/functional assignment, the analysis will proceed at NP internal adjunct modifier position. If this is successful, the adjunct participial clause will be interpreted locally. Then the parser will continue its traversal of the grammar at i_double_bar position, searching for the finite verb.

In case no finite verb is available, there will be an ensuing failure which will recovered gracefully by a recovery call for the same main constituent expected by the grammar in that position. Two actions will take place:

- the current input word will have to be a nonfinite verb;
- the already parser portion of the input sentence must contain a possibly ambiguous finite verb;
- this token word should correspond to the predicate lemma heading the modifier adjunct clause computed inside the NP which is scanned to search for the appropriate structural portion.

The first two actions are carried out on the lookahead stack, while the third action is carried out on the NP structure already parsed and fully interpreted by the parser.

References

[1] Steve Abney, (1996), Part-of-Speech Tagging and Partial Parsing, in Ken Church, Steve Young, and Gerrit Bloothooft, eds. Corpus-Based Methods in Language and Speech, Kluwer Academic Publishers, Dordrecht, 118-136.

[2] K.Church(1988). A stochastic parts program and noun phrase parser for unrestricted texts, in Proc.2nd Conference on Applied Natural Language Processing, Austin, Texas, 136-143.

[3] DeRose S.(1988), Grammatical category disambiguation by statistical optimization, Computational Linguistics, 14(1), 31-39.

[4] Garside R.(1987), The CLAWS word-tagging system, in Garside R., F.Leech and G.Sampson (eds), The Computational Analysis of English, Longman, 30-41.

[5] K.Church, W.Gale, P.Hanks, & D.Hindle (1989). Parsing, word associations and typical predicate-argument relations in IWTP, 389-98.

[6] R.Dale, Symbolic Approaches to Natural Language Processing, in R.Dale, H.Moisl, H.Somers (2000). Handbook of Natural Language Processing, Marcel Dekker, New York, 1-10.

[7] S.Armstrong-Warwick (1993). Preface, Computational Linguistics 19(1), Special Issue on Using Large Corpora: I, iii-iv.

[8] J.R.Hobbs et al. SRI International: Description of the FASTUS system used for MUC-4. In Proc. 4th Understanding Conference, San Mateo, CA, Morgan Kaufmann, 268-275.

[9] Dan Klein and Christopher D. Manning. 2002. Conditional Structure versus Conditional Estimation in NLP Models. 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 9-16.

[10] John A.Carroll, Statistical Parsing, in R.Dale, H.Moisl, H.Somers (2000). Handbook of Natural Language Processing, Marcel Dekker, New York, Chapt.22, 525-43.

[11] Stephan Oepen, Dan Flickinger, Chris Manning, Kristina Toutanova, LinGO Redwoods - A Rich and Dynamic Treebank for HPSG, webpage of the project.

[11] Fred Karlsson & Lauri Karttunen, Shallow Parsing, in Giovanni Varile, Antonio Zampolli, 1995. Survey of the State of the Art in Human Language Technology Editorial Board: Ronald A. Cole, Editor in Chief, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, Victor Zue, Chapter 3: Language Analysis and Understanding, 113-114.

[12] Fernardo Pereira, Sentence Modeling and Parsing, in Giovanni Varile, Antonio Zampolli, 1995. Survey of the State of the Art in Human Language Technology Editorial Board: Ronald A. Cole, Editor in Chief, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, Victor Zue, Chapter 3: Language Analysis and Understanding, 113-114.

[13] Ted Briscoe, Robust Parsing, in Giovanni Varile, Antonio Zampolli, 1995. Survey of the State of the Art in Human Language Technology Editorial Board: Ronald A. Cole, Editor in Chief, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, Victor Zue, Chapter 3: Language Analysis and Understanding, 113-114.

[14] Alan Agresti. 1990. Categorical Data Analysis. John Wiley & Sons, New York.

[15] Hana Filip, 1998. REDUCED RELATIVES: LEXICAL CONSTRAINT-BASED ANALYSIS, Proceedings of the Twenty- Fourth Meeting of the Berkeley Linguistic Society, 1-15.

[15] Stevenson, S. and P. Merlo. 1997. “Lexical Structure and Parsing Complexity”. Maryellen C. MacDonald (ed.) *Lexical Representations and Sentence Processing*. Psychology Press.,349-399.

[16] Bresnan, Joan. 2001. Lexical-Functional Syntax. Blackwells.

[17] Delmonte R. (2000), SLIM Prosodic Automatic Tools for Self-Learning Instruction, Speech Communication 30, 145-166.

[18] R.Sproat(ed.), 1998, Multilingual Text-to-Speech Synthesis: The Bell Labs Approach, Dordrecht, Kluwer Academic, Introduction, i-iv.

[19] Delmonte R.(1987), Grammatica e ambiguità in Italiano, **Annali di Ca' Foscari** XXVI, 1-2, pp.257-333.