EVALUATING *GETARUNS* PARSER WITH GREVAL TEST SUITE

Rodolfo Delmonte
Ca' Garzoni-Moro, San Marco 3417
Università "Ca Foscari"
30124 - VENEZIA
Tel. 39-041-2349464/52/19 - Fax. 39-041-5287683
E-mail: delmont@unive.it - website: project.cgm.unive.it

Abstract

GREVAL, the test suite of 500 English sentences taken from SUSANNE Corpus and made available by John Carroll and Ted Briscoe at their website, has been used to test the performance of a symbolic linguistically-based parser called GETARUNS presented in (Delmonte, 2002). GETARUNS is a symbolic linguistically-based parser written in Prolog Horn clauses which uses a strong deterministic policy by means of a lookahead mechanism and a WFST. The 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: in this sense it is equivalent to an LR parser. The results obtained fare higher that the ones reported in (Preis, 2003) and this we argue is due basically to the symbolic rule-based approach: we reach 96% precision (coverage) and 84% recall (accuracy). We assume that from a psycholinguistic point of view, parsing requires setting up a number of disambiguating strategies, to tell arguments apart from adjuncts and reduce the effects of backtracking. To do that the system is based on LFG theoretical framework and uses Grammatical Functions information to help the parser cope with syntactic ambiguity. In the paper we shall comment on some shortcomings of the Greval corpus annotation and more in general we shall criticize some aspects of the Dependency Structure representation.

1. Introduction

In this paper we will present the parser used by the system GETARUN and discuss its performance with the test suite called GREVAL set up by Carroll & Briscoe. We will also discuss the mapping algorithm from LFG to Dependency Grammatical Relations (DGRs), which we have been obliged to develop in order to be able to evaluate our parser. Greval is a benchmark for parser evaluation based on Grammatical Relations in a Head Dependency Structure style output, i.e. a word based Head-Dependent flat representation enriched with Grammatical Relation information, where each relation is represented as follows,

Relation(introducer,head,dependent)
Relation(introducer,head,dependent,deep-relation)

where a deep relation is introduced basically for passive constructions, dative shift, and potentially other structures, according to the "Movement" approach invoked by chomskians. The annotation adopted by the authors is a surface level GRs approach where for instance, in cases of Locative Inversion as in sentence 284,

(1) Here, in the old days - when they had come to see the moon or displays of fireworks - sat the king and his court while priests, soldiers, and other members of the party lounged in the smaller alcoves between.

the SUBJect NP 'the king and his court" is assigned to DOBJ and then receives an additional deep-relation label as NCSUBJ to indicate its original deep structure position. However the same relation is "wrongly" marked as NCSUBJ in a subsequent case of Locative Inversion (the only other one, sentence 445), which we report below,

(2) In his stead is a milquetoast version known as the corporation.

where the inverted subject NP "a milquetoast version" is annotated straightforwardly as NCSUBJ. The inconsistency denounced by this case of double annotation lingers on other types of ambiguous GRs that we will comment below.

In our experiment, for reasons already explained in (Crouch et al., 2002) and further commented below, we restricted our mapping algorithm to all "pred-only" f-structures, i.e. only to semantic heads with primary GRs. This is because we assume that the most difficult task a parser is faced with when parsing a sentence is to operate the argument-adjunct distinction: thus the subset of GRs we will take into account is the following:

NCSUBJ, DOBJ, IOBJ, ARG_MOD, XCOMP, CCOMP

Difficulties in building up a comparable version of our output, include the inconsistencies present in the non typographical text distributed for the test. We also had to give up using the internally provided tool for evaluation because our system builds multiword expressions which are almost totally absent in the annotated Gold version. So even though the authors admit to the need of improving this aspect, its lack makes it impossible for real systems to use automatic evaluation tools.

1.1 Multiwords

Multiwords go from obvious cases such as United States, which is wrongly treated in the annotated corpus as two separate words – "state" modified by "united" – to prepositional and adverbial locutions some of which have been individuated but have been left with an SGML markup, as shown here,

accord<blank>to
in<blank>favour<blank>of
even<blank>if
no<blank>one
a<blank>few
out<blank>of
in<blank>order<blank>to
at<blank>least
so<blank>that
for<blank>example
along<blank>with
in<blank>front<blank>of
because<blank>of
as<blank>to
e.<blank>g.

In addition to these 15 multiwords, we produced over 220 nouns, adverbials and adjectives which contributed in an important manner to disambiguate both syntactic and semantic processing, as well as to facilitate tagging.

1.2 Mapping from LFG to DGRs

However the most difficult part of the work was the mapping itself. The mapping from LFG to DGRs requires the setting up of a principled distinction between GRs.

The main difficulties were due to the treatment of OBLiques vs. PP adjuncts: LFG erases the preposition of oblique arguments which is no longer available for comparison, on the contrary, the preposition is preserved in case it is semantically needed to identify the semantic role associated to PP adjuncts. In DGRs instead we have two options reported below and taken from the online Readme document,

ncmod(type, head, dependent) is the most common relation type and is used to encode PP, adjectival/adverbial modification, nominal modifiers, and so on. **ncmod** is also used to indicate a particle `modifier' in a verb particle combination, e.g. Bill looked the word up is encoded as ncmod(prt, look, up); *of in consist of is treated as a particle.*

iobj(type, head, dependent) is the relation between a predicate and a non-clausal complement introduced by a preposition;

iobj(in, arrive, Spain)	*arrive in Spain*
iobj(into, put, box)	*put the tools into the box*
iobj(to, give, poor)	*give to the poor*

The definitions do not respect the actual decisions taken in the annotation process because only adjective and verb predicates are assigned complement labels like nsubj, dobj and iobj. Nouns are only assigned generic modifiers, apart from 9 cases of "wrongly" labeled IOBJ complements which we comment in more detail below. On the contrary, in our representation, possessor relations and other agent-like relations in noun modification are computed as SUBJ. The remaining "of" headed PPs are all computed as OBJ in case the noun is a deverbal predicate and is transitive; other prepositions introduce OBLiques and ADJuncts again according to the governing noun.

As to verb and adjective predicates, in order to be able to sort out NCMODs from IOBJs, all subcategorization relations must be suitably encoded, be it obligatory or optional ones, as can be understood from the examples of IOBJs listed above and taken from the online accompanying document. In particular, ARRIVE would be computed as taking a Locative PP complement at the same level of PUT, whereas only the latter requires the complement obligatorily. As to the third example, GIVE, a ditransitive verb allowing Dative Shift, its indirect object would be computed as a case of IOBJ thus collapsing an important linguistic distinction existing between OBLiques and Indirect Objects.

However, given the generic criteria followed by the annotators, we still wanted to verify the consistency of annotation of IOBJs, so we checked with COMLEX subcategorization frames whether the

relation would be predicted or not. When collecting the IOBJs of the corpus we soon discovered that 9 IOBJS constitute cases of non verbal complementation which we report below,
(iobj to akin future) ;;; non-verbal complementation
(iobj to adherence principle)
(iobj to applicability people)
(iobj of independent pressure)
(iobj of independent volume)
(iobj notorious for disregard)
(iobj to atune Whig)
(iobj to key acceptance)
(iobj on ruin country)

then, there is one dubious case of IOBJ: in sentence 316 the ellipsed governing adjective predicate "atune" is done away with and the IOBJ relation is assigned to the verb BE,

(3) Indeed, the old Jeffersonians were far more atune to the Hamilton oriented Whigs than they were to the Jacksonian Democrats.

(ncsubj be Jeffersonian _)
(xcomp _ be atune)
(iobj to atune Whig)
(ncsubj be they _)
**(iobj to be democrat)

From the search in COMLEX of the remaining 127 predicates governing IOBJ relations, we derived 20 predicates missing the preposition required for the complement discriminative choice. In more than one case the choice of complement (iobj) vs. adjunct (ncmod) is highly disputable. This situation makes the comparison and evaluation of IOBJs very uncertain and bound to low scoring as happened with the parsers included in the test reported under (Preis, 2003). As an additional remark, from the definition it would appear that OBLiques are treated as DOBJ of a preposition particle which is in turn treated itself as **ncmod**. To better clarify the issue we partially report the annotation of example 244 from the corpus, where we italicize the relevant relations,

(4) Meanwhile, the experts speak of wars triggered by false pre-emption, escalation, unauthorized behavior and other terms that will be discussed in this report.

ncsubj(speak, expert, _)
dobj(speak, war, _)
ncsubj(trigger, war, obj)
arg_mod(by, trigger, pre-emption, subj)
arg_mod(by, trigger, escalation, subj)
arg_mod(by, trigger, behaviour, subj)
arg_mod(by, trigger, term, subj)
ncsubj(discuss, term, obj)
ncmod(prt, speak, of)

So we decided that in our evaluation we treat as IOBJs both those actually produced by our parser and matched directly with the Gold corpus, as well as those which appear in the Gold corpus as DOBJ governed by a preposition NCMOD and also those that have been computed as NCMODs directly, as long as the head and the dependent are identical.
We then individuated a number of mismatches in the Gold annotation which would not receive a suitable mapping in our output, which were then interpreted as mistakes by Ted Briscoe (p.c.) in particular cases of secondary predication for the class of ECM verbs (consider, believe, term, etc) which were treated as OBJ2, as well as the relation of DOBJ associated to complements of verb HAVE, which we compute as copulative verb.

(5) Sunday he had added, We can love Eisenhower the man, even if we considered him a mediocre president... but there is nothing left of the Republican Party without his leadership.

ncsubj(add, he, _)
ncsubj(love, we, _)
dobj(love, Eisenhower, _)
ncsubj(consider, we, _)
dobj(consider, he, _)
obj2(consider, president, _)

For this reason we didn't include data for OBJ2 in our comparison which required too many changes in our parser architecture in order to have the appropriate mapping. As to clause level GRs, we computed both open and closed sentential (CCOMP) and small clause (XCOMP) complements, but we did not compute open adjuncts – XMODS – which again did not seem to be easily comparable in our mapping. As a matter of fact, these clausal complements have not been separated in the published test and only figure as CLAUSAL. For comparison reasons we had to erase all subject relations made available by our LFG-based representation for all open predicative complements which were however not represented in the Gold manually annotated GREVAL corpus.

2. Parsing and Robust Techniques

As far as parsing is concerned, we purport the 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. Our parser is a rule-based deterministic parser in the sense that it uses a lookahead and a Well-Formed Substring Table to reduce backtracking. It also implements Finite State Automata in the task of tag disambiguation, and produces multiwords whenever lexical information allows it. In our parser we use a number of parsing strategies and graceful recovery procedures which follow a strictly parameterized approach to their definition and implementation. Recovery procedures are also used to cope with elliptical structures and uncommon orthographic and punctuation patterns. A shallow or partial parser, in the sense of (Abney, 1996) is also implemented and always activated before the complete parse takes place, in order to produce the default baseline output to be used by further computation in case of total failure.

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 analysis for English has also been implemented and used for OOV words. The system uses a core fully specified lexicon, which contains approximately 10,000 most frequent entries of English. In addition to that, there are all lexical forms provided by a fully revised version of COMLEX. 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 – 270,000 lexical entries - in which we used 75 semantic classes similar to those provided by CoreLex.

All parser rules from lexicon to c-structure to f-structure amount to 7532 rules, thus subdivided:

1. Calls to lexical entries - morphology and lexical forms: 3865 rules

2. Syntactic and semantic rules in the parser proper: 2617 rules

All syntactic/semantic rules: 6482 rules

4. Semantic Rules for F-Structure

Lexical Rules for Consistency and Control:- 439 rules

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

Quantifier Raising and Anaphoric Control:- 441 rules

All semantic f-structure building rules: 1050 rules

The parser itself is made up of 51,000 lines of code. This does not take into account the code for the lexicon – with fully specified subcategorization frames - and the dictionary for morphological decomposition: 6600 entries for the lexicon and 76,000 entries for the dictionary. These are all consulted at runtime. Eventually the semantics from the WordNet and other sources derived from the web make up three hash-tables for 5 Mb overall sitting on the hard disk and accessed when needed.

Our training corpus for the complete system is made up 200,000 words and is organized by a number of texts taken from different genres, portions of the UPenn WSJ corpus, test-suits for grammatical relations, narrative texts, and sentences taken from COMLEX manual.

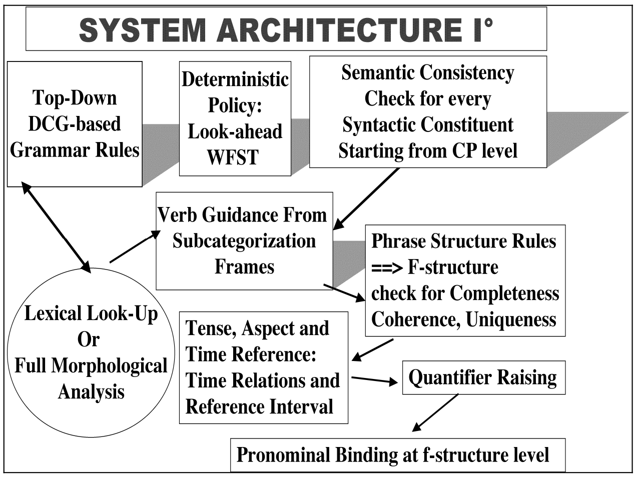
SYSTEM ARCHITECTURE I°
Top-Down DCG-based Grammar Rules
Deterministic Policy: Look-ahead WFST
Semantic Consistency Check for every Syntactic Constituent Starting from CP level
Verb Guidance From Subcategorization Frames
Phrase Structure Rules ==> F-structure check for Completeness Coherence, Uniqueness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

Fig.1 GETARUNS' LFG-Based Parser

2.1 Lookahead and FSA

One of the important differences we would like to highlight is the use of topdown lookahead based parsing strategies. The following list of preterminal 14 symbols is used:

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. av=adverb
14. x=punctuation

Tab. 1: Preterminal symbols used for lookahead

As has been reported in the literature (see Tapanainen and Voutilainen 1994; Brants and Samuelsson 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 depending on the tagset). Lookahead in our system copes with most cases of ambiguity: however, we also had to introduce a 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. Besides, all FSA may be augmented by tests related to linguistic properties needed for disambiguation; some such tests are,

□ check subcategorization frame for current word

This is used for [n,v], [a,n,v] ambiguity classes followed by a preposition, or followed by “that”

□ check for gerundive verb form

This is used to check for –ing endings of words

□ check for auxiliary and modals

This is used to disambiguate [n,v], [a,n,v] ambiguity classes when preceded by an auxiliary or a modal

□ check for noun belonging to factive class

This is used to disambiguate “that” [a,c,r] ambiguity class when preceded by a governing noun

□ check for verbs of saying

This is used to disambiguate verbs preceded or followed by punctuation marks

3. GETARUNS: a Linguistically and Psychologically Based Parser

The parser 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 phrase-structure rules in a given sequence 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.

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 use of strategies calls for psychologically related disambiguation processes which are strictly bound to linguistic parameters. For instance, English is a language that freely allows compless (complementizer-less) complement and relative clauses. Being the sentence the highest recursive structural level, it is plausible that English speakers will adopt some strategy in order to avoid falling in a garden path - thus freezing the parser. Another peculiar feature of English regards the inherent ambiguity of Past Tense/Past Participle verb forms, exception made for irregular verbs which however only constitute a small subset in the verb lexicon of English amounting in our case to some 20,000 entries. Seen that Reduced Relative Clauses are headed by the past participle verb form, and that Participial Adjuncts may be attached to any NP head nouns quite consistently; and seen also that is very hard to apply strict subcategorization tests for participial SUBJect - or deep OBJect in case of passives - with good enough confidence we assume that such tests will only be performed in case the parser is at the complement level of the SUBJect NP. The reason for this being that we need to prevent as much as possible failures at the I_bar level. For this reason we pass grammatical function information down into the NP complement level in order to be used for that purpose.

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, 2000a; Delmonte, 2000b).
As said above, English allows an empty Complementizer for finite complement and relative clauses, two structures which contribute a lot of indeterminacy to the parsing process. However, in our system, this can be nicely accomodated by using linguistic information to prevent the rule to be entered by the parser. 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 in association with local lookahead to 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.
The grammar formalism implemented in our system is not fully compliant with the one suggested by LFG theory (Bresnan, 2001), in the sense that we do not use a specific Feature-Based Unification algorithm but a Prolog-based parsing scheme. On the other hand, Prolog language gives 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 will need here to make clear what we intend for "LFG-related grammar organization": as said above, we are not following LFG theory strictly in that the parser is not organized as would all constraint unification-based parsers, with a context-free or context-augmented grammar that produces constituents which are then passed to a unification mechanism to check for consistency, uniqueness and coherence in LFG terms, or simply to check for feature agreement and subcategorization constraints satisfaction. In our parser the grammar is organized by Grammatical Functions which call syntactic constituents. Each Grammatical Function call passes functional information to the constituent level which is paramount to the lookahead mechanism and makes available syntactic constraints of a higher level than the constituent itself.

3.1 The Organization of Grammar Rules

The grammar is divided up into five main levels:
- the complex utterance level, where choices are made for subordinate/coordinate, direct/indirect discourse markers, or as simple assertion;
- the utterance level, where choices are made for detecting a question vs. assertion;
- the simple assertion level, where we may have assertions with a sentential subject; a verbal structure as a gerundive as subject; a fronted OBJect NP as focalized constituent or a Locative Inversion sentence structure.

In case none of the previous structures are detected the parser enters the CP level of canonical sentences where Aux-to-Comp may take place.
All sentential fronted constituents are taken at the CP level. Adjuncts at this level may be of many different kinds, some of them conflicting with the same role the constituent may have at a lower level. For instance, a vocative NP may be present, fronted PP complements, as well as various types of parenthetical structures: here again, the parser must be told at which level of computation it is actually situated in the grammar. This is done again by passing down the corresponding grammatical function. When the parser leaves this level of computation it will enter the canonical IP level where the SUBJect NP must be computed, either as a lexically present nominal head or as a string passed in the Extraposition variable. Then again a number of ADJuncts may be present between SUBJect and verb, and they can be adverbials and parentheticals. When this level is left the parser enters the I_bar level where it is expecting a verb in the input string. This can be a verb complex with a number of internal constituents, but the first item must be definitely a verb. In case there is none, a number of fail-soft and recovery strategies are tried to check whether the parser has taken a participial as ADJunct or as reduced relative clause in a previous parse which is passed down to perusal. Also in case there was no nominal element available at IP level a number of recovery strategy are tried to check whether the parser has taken an Appositive or a Vocative in a previous parse. So all the previously parser material is either passed down or is recorded in a WFST to be used in an upper level in case of failure at a lower level.
The parser is strictly top-down, depth-first, one-stage parser with backtracking: differently from most

principle-based parsers presented in (Berwick, Abney, and Tenny, 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.

What are the main advantages of performing a topdown lookahead driven parse as compared to unification procedures applied to a LR parse table as commented in (Carroll, 2000)? First of all, our grammar is not a CF grammar seen that CF rules are multiplied by all different sentence positions at which they may occur: in order to do that, a NP may be called by a SUBJect, an OBJect, a NCOMPlement, an APPosition, a VOCative etc. so that different properties and constraints may be associated with each NP realization.

Consider now the wellknown case of COMPlementizerLess complement clauses in English as represented by the following example (all examples are taken from Greval):

(6) A Yale historian, writing a few years ago in The Yale Review, said We in New England have long since segregated our children.

And now consider the case in which a sentential complement is used in subject position, as in

(7) That any sort of duty was owed by his nation to other nations would have astonished a nineteenth century statesman.

No rule accessing sentential complement would be used to look for subject complement clauses which are only accessed at sentence level as a special case of sentence structure. In order to take care of compless complement clauses the parser checks subcategorization frames, then in case the complementizer is missing, it activates a check for the semantic typology of verb predicates which allow the complementizer to be omitted, which coincides with bridge verbs or non-factive verbs.

Now consider the case in which the complement clause follows the object (direct,indirect) NP, as in

(8) He told the committee the measure would merely provide means of enforcing the escheat law which has been on the books since Texas was a republic.

This strategy is organized in a similar way to checking for the attachment of a PP complement following a NP. The complementation information is turned into a “that” word in case of sentential complement, and into the whole set of prepositions subcategorized by the verb with PP complements. These words are used as Cues-set to prevent the NP from entering relative clause rules, or any PP headed with one of the prepositions listed in the Cues-set, after the head has been taken and the parser is in the complement block of rules. The Cues-set is passed as a list from the I_bar level down to the vp level and into the object NP if any.

Now consider cases in which the parser has to choose between an OBJect NP/Sentence complement SUBJect in case the verb is compless as shown in:

(9) Mitchell decried the high rate of unemployment in the state and said the Meyner administration and the Republican controlled State Senate Must share the blame for this.

Or this sentence where the complement is started by a comma, and a vocative,

(10) A man must be able to say, Father, I have sinned, or there is no hope for him.

As said above, we ascertain the verb belongs to the semantic class of non-factive verbs and then look for a finite verb ahead before allowing the Sentence complement rule to be fired. Other similarly difficult cases that can be adequately treated in our parser are shown below,

(11) I told him what Liston had said and he said Liston was a double crosser and said anything he (Liston) got was through a keyhole.

where both the complement clause and the following relative clause are compless. Getting the final results reported in Table 2. with Greval took us one month/man work to account for rules and lexical entries missing in the parser. Some such coverage problems were caused by sentences like 12 and 13 below,

(12) Wagner replied, Can't you just see the headline City Hooked for $172,000 ?

(13) Yet, I responded, could not similar things be said about the art of the past ?

Or an imperative/exhortative followed by a question as in,

(14) Take Augustine's doctrine of grace given and grace withheld : have you pondered the dramatic qualities in this theology ?

or a subordinate clause followed by a question as in,

(15) If he attaches little importance to personal liberty, why not make this known to the world ?

Hard sentences to parse were the following ones,

(16) Battalion Chief Stanton M. Gladden, 42, the central figure in a representation dispute between the fire fighters association and the teamsters union, suffered multiple fractures of both ankles.

(17) He bent down, a black cranelike figure, and put his mouth to the ground.

where in 16 there are two long parentheticals fairly hard to process before the main verb comes; in 17 the appositive comes after the verb and not after head noun, the pronoun “he”.

New lexical entries had to be added to account for special multiwords basically in the area of grammatical function words. We also added some new lexical multiwords which caused the FSA disambiguation problems.

4. Evaluation and Discussion

The parser parses 89% of all text top down: then it parses 9.3% of the remaining linguistic material bottom up and adds it up to the parsed portion of the current sentence. That may produce wrong results in case a list has been partially parsed by the top down parser. But it produces right results whenever any additional complete subordinate or coordinate sentence structure has been left over – which constitutes the majority of cases. Overall almost the whole text – 98.3% - is turned into semantically consistent structures which have already undergone Pronominal Binding at sentence level in their DAG structural representation.

We find it very important to remark the fact that the performance of our parser is mainly to be appreciate for the high coverage. None of the statistically and stochastically based parser reported under (Preis, 2003) reached such a high score.

ALL-RELS	GOLD	VENICE	CORRECT	%PRECIS	RECALL/ GOLD	RECALL/ GUESS
xcomp	361	344	274	95.29	75.90	79.65
ncsubj	1038	1038	883	100.00	85.06	85.06
iobj	158	126	99	79.74	62.65	78.57
dobj	409	394	331	96.33	80.92	84.01
argmod	38	30	26	78.94	68.42	86.66
ccomp	81	80	62	98.76	76.54	77.5
TOTAL	2085	2012	1675	96.49	80.33	83.25

Table 2: Grammatical Relations produced by GETARUNS with Precision and Recall

For the sake of comparison we also report the main data taken from the table presented under (Preis, 2003) to allow the reader to appreciate the results of our parser. In particular, the number of main Grammatical Relations treated in the previous test is half the number in comparison with ours. If we look at the easiest GR to parse, i.e. the NCSUBJ GR, we see that the number of cases found by the best parser in the previous test is by far lower that our result. The highest case of precision is for DOBJ in the BU parser which reaches 88.42% which is 8 points lower that our result. In absolute terms, limiting the comparison to the two most frequent GRs, the best parser – BU - has found 361 DOBJs against our 394; 891 NCSUBJs against our 1038.

GF	oec	BC - PRECIS	BC - RECALL	BU - PRECIS	BU - RECALL	CH - PRECIS	CH - RECALL	C1 - PRECIS	C1 - RECALL	C2 - PRECIS	C2 - RECALL
dobj	409	85.83	78.48	88.42	76.53	84.43	75.55	86.16	74.57	84.85	75.31
iobj	158	31.93	67.09	57.75	51.9	27.53	67.09	27.09	69.62	27.01	70.25
ncsubj	1038	81.99	82.47	85.83	72.93	81.8	70.13	79.19	65.99	81.29	69.46
obj2	19	27.45	73.68	46.15	31.58	61.54	42.11	81.82	47.37	61.54	42.11
arg_mod	41	-	-	75.68	68.29	78.12	60.98	82.86	70.73	82.86	70.73
clausal	403	43.27	52.61	75.79	71.46	62.19	43.67	50.57	32.75	49.11	27.30

Table 3: GR Precisions and Recalls as derived from (Preis, 2003)

The recall is also accordingly lower in absolute terms: there are 277 cases of correct DOBJs in BU against our 331, and 702 cases of correct NCSUBJs – in this case it is the BC parser that gets best recall – against our 883. And 277 and 702 are slightly better than chance - 67.72 and 67.63 respectively.

The impression one gets from the performance of statistically and stochastically based parsers is that they are inherently unable to cope with deep linguistic information. They are certainly impossible to undergo substantial improvements. On the contrary, rule based parsers would benefit from additional subcategorization frames as in our case: and for all those constructions which require setting up of new additional peripheral rules in the grammar, they would typically increase their coverage, as did our parser.

As a last comment, we started evaluating subsets of GREVAL corpus with the online version of "Connexor" dependency parser, on the assumption that that version would be identical or even better than the one commercially available. We did that because this parser is regarded the best dependency parser on the market. We tried out a subset of 50 sentences, and on a first perusal of the output we discovered that only 40 sentences contained correct, and fully connected representations. The remaining 10 sentences either presented unconnected heads, or misconnected ones due to wrong attachments. Some remarks on the possible reasons for that:

- ❑ bottom up local parsing techniques are good at coping with typically hard to parse structures for a top down parser like coordinate structures but they are bad at computing long distance dependencies;
- ❑ they are good at computing attachment whenever it is local, but they make mistakes when there are extraposed elements;
- ❑ dependency parsing does not seem to obey to generally accepted grammaticality principles like the obligatoriness of SUBJect constituents, nor the need to provide some landing site for extracted wh- elements in relative and interrogative clauses;
- ❑ control structures like small clauses for predicative complements and adjuncts are all attached locally, which is not always the case.

So, even though word-level parsing may be more effective as to the number of connections (constituents) safely produced, without leaving off any fragment or skimmed fragment, it is nonetheless faced with the hard task of recomposing clause level control mechanisms which in a top down constituency-based parser are given for granted.

The F-measure derived from our P and R according to the usual formula:

$$F1(r,p) = \frac{2rp}{r+p}$$

is 89.38%, which is by far higher than the 75% reported in (Crouch et al., 2002) as being the best result obtained by linguistic parsers today.

We are currently experimenting with a "mildly" topdown/bottomup version of the parser in which rather than starting from Clause level we search recursively for Arguments and Adjuncts. In other words, we look for fully semantically interpreted constituents in which choice for argumenthood has already been partially performed. In addition to collecting Arguments/Adjuncts, the new parser scatters in the output list punctuation marks and coordinate/subordinate words which are deemed responsible to determine clause boundaries. To that aim, we devised a procedure for clause creation under the restriction that a main tensed Verb constituent complex has been found. This can be iterated on the input list and the procedure may decide to fuse portion of the output list which have been left stranded without independent clause status, and append it to the preceding prospective clause. Interpretation procedures follows by recovering subcategorization frames for the main tensed verb and assignment of grammatical function and semantic roles takes place. The Clause level procedure is then followed by an Utterance level procedure that produces simple utterance or complex ones – coordinated or subordinated – according to the Clause input list.

We experimented the new version of the parser with the Greval Corpus and discovered that in some cases it was much slower than the fully topdown version. However, we also recovered parsing time in highly ambiguous and complex sentences, where the “mildly” bottomup parser actually followed a totally linear behaviour: no increase in computation time resulted and the performance is only conditioned by the number of words/number of arguments-adjuncts/number of clauses to build. We haven’t been able to compute this proportion systematically but will do so in the future.

5. References

Abney, S. 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.

Berwick, R.C., S. Abney, and C. Tenny. 1991. Principle-Based Parsing: Computation and Psycholinguistics. New York: Kluwer Academic Publishers.

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

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Blackwells.

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

Delmonte R. 2002. GETARUN PARSER - A parser equipped with Quantifier Raising and Anaphoric Binding based on LFG, Proc. LFG2002 Conference, Athens, pp.130-153, at http://cslipublications.stanford.edu/hand/miscpubsonline.html.

Delmonte R. 2000a. Parsing Preferences and Linguistic Strategies, in LDV-Forum - Zeitschrift fuer Computerlinguistik und Sprachtechnologie - "Communicating Agents", Band 17, 1,2, (ISSN 0175-1336), pp. 56-73.

Delmonte R. 2000b. Parsing with GETARUN, Proc.TALN2000, 7° confèrence annuel sur le TALN,Lausanne, pp.133-146.

Preis J., 2003. Using Grammatical Relations to Compare Parsers, in Proc., EACL, Budapest, pp.291-298.

Richard Crouch, Ronald M. Kaplan, Tracy H. King, Stefan Riezler 2002. A Comparison of Evaluation Metrics for a Broad Coverage Stochastic Parser. In *Proceedings of the Workshop on "Parseval and Beyond", LREC'02*, Las Palmas, Spain, 67-74.

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