Text Understanding with GETARUNS for Q/A and Summarization

Rodolfo Delmonte
Department of Language Sciences
Università Ca' Foscari
Ca' Garzoni-Moro - San Marco 3417 - 30124 VENEZIA
e-mail: delmont@unive.it website - http://project.cgm.unive.it

Abstract

Summarization and Question Answering need precise linguistic information with a much higher coverage than what is being offered by currently available statistically based systems. We assume that the starting point of any interesting application in these fields must necessarily be a good syntactic-semantic parser. In this paper we present the system for text understanding called GETARUNS, General Text and Reference Understanding System (Delmonte, 2003a). The heart of the system is a rule-based top-down DCG-style parser, which uses an LFG oriented grammar organization. The parser produces an f-structure as a DAG which is then used to create a Logical Form, the basis for all further semantic representation. GETARUNS, has a highly sophisticated linguistically based semantic module which is used to build up the Discourse Model. Semantic processing is strongly modularized and distributed amongst a number of different submodules which take care of Spatio-Temporal Reasoning, Discourse Level Anaphora Resolution.

1. Introduction

GETARUNS, the system for text understanding developed at the University of Venice, is equipped with three main modules: a lower module for parsing where sentence strategies are implemented; a middle module for semantic interpretation and discourse model construction which is cast into Situation Semantics; and a higher module where reasoning and generation takes place (Delmont & Bianchi, 2002) .

The system is based on LFG theoretical framework (Bresnan, 2001) 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 dominating 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 (Delmonte, 2002). In other words, 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 cross compatibility tests on the basis of selectional restrictions of main predicates and adjuncts heads.

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. In that case partial semantic mapping will take place where no Logical Form is being built and only referring expressions are asserted in the Discourse Model – but see below.

1.2 The Binding Module

The output of grammatical modules is then fed onto the Binding Module(BM) which activates an algorithm for anaphoric binding in LFG terms using f-structures as domains and grammatical functions as entry points into the structure. Pronominals are internally decomposed into a feature matrix which is made visible to the Binding Algorithm(BA) and allows for the activation of different search strategies into f-structure domains. Antecedents for pronouns are ranked according to grammatical function, semantic role, inherent features and their position at f-structure. Special devices are required for empty pronouns contained in a subordinate clause which have an ambiguous context, i.e. there are two possible antecedents available in the main clause. Also split antecedents trigger special search strategies in order to evaluate the set of possible antecedents in the appropriate f-structure domain. Eventually, this information is added into the original f-structure graph and then passed on to the Discourse Module(DM). We show here below the architecture of the parser.

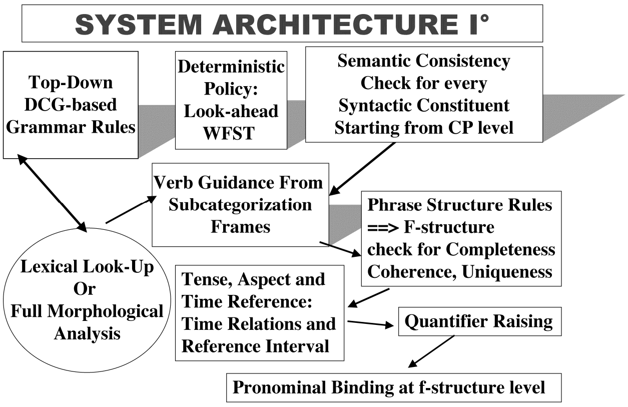
SYSTEM ARCHITECTURE I°
Top-Down DCG-based Grammar Rules
Deterministic Policy: Look-ahead WFST
Semantic Consistency Check for every Syntactic Constituent Starting from CP level
Verb Guidance From Subcategorization Frames
Phrase Structure Rules ==> F-structure check for Completeness Coherence, Uniqueness
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

1.3 Lexical Information

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.

Our training corpus which 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, and sentences taken from COMLEX manual.

To test the parser performance we used the “Greval Corpus” made available by John Carroll and Ted Briscoe which allows us to measure the precision and recall against data published in (Preis, 2003). The results obtained are a 90% F-measure which is by far the best result obtained on that corpus by other system, ranging around 75%. Overall almost the whole text - 98% - is turned into semantically consistent structures which have already undergone Pronominal Binding at sentence level in their DAG structural representation. The basic difference between the complete and the partial parser is the ability of the first to ensure propositional level semantic consistency in almost every parse, which is not the case with the second.

2. The Upper Module

GETARUNS, has a highly sophisticated linguistically based semantic module which is used to build up the Discourse Model. Semantic processing is strongly modularized and distributed amongst a number of different submodules which take care of Spatio-Temporal Reasoning, Discourse Level Anaphora Resolution, and other subsidiary processes like Topic Hierarchy which will impinge on Relevance Scoring when creating semantic individuals. These are then asserted in the Discourse Model (hence the DM), which is then used to solve nominal coreference together with WordNet. The system uses two resolution submodules which work in sequence: they constitute independent modules and allow no backtracking. The first one is fired whenever a free sentence external pronoun is spotted; the second one takes the results of the first submodule and checks for nominal anaphora. They have access to all data structures contemporarily and pass the resolved pair, anaphor-antecedent to the following modules. Semantic Mapping is performed in two steps: at first a Logical Form is produced which is a structural mapping from DAGs onto of unscoped well-formed formulas. These are then turned into situational semantics informational units, infons which may become facts or sits. Each unit has a relation, a list of arguments which in our case receive their semantic roles from lower processing – a polarity, a temporal and a spatial location index.

2.1 Logical Form Creation and Semantic Mapping

In order to produce a semantic interpretation from the output of the parser we adopt a uniform meaning representation which is a structured Logical Form(LF). In other words we map our f-structures into a linear formalism that can capture the basic meaning of the structural units of grammatical representation. We assume that parsing has made explicit predicate-argument relations as well as subordination and adjunction in f-structure representation: no ambiguity has been left to decide in the semantics, seen that all constituents have been assigned a preferential reading.

LF representations are used to generate a semantic analysis for an utterance: in this sense, they represents its interpretation in context and also its truth conditions. In fact, the system generates a situation semantics mapping directly from LF, and that is used to update the Discourse Model with new discourse entities or new properties of already existing entities.

LF is basically a flat version of f-structure, where the main verb predicate is raised at the higher node, and arguments and adjuncts are stripped off of useless information w.r.t. semantic mapping. In order to produce a semantic interpretation of each utterance we proceed as follows:

A. we start from DAGs(Direct Acyclic Graphs) available for each utterance, i.e. f-structures, and perform pronominal binding and anaphora resolution at discourse level. Our f-structures are enriched with Semantic Roles which are derived from our augmented Lexical Forms by a match with the head Noun inherent features and selectional restrictions. Semantic match is also performed for Adjuncts, which require an intermediate Preposition and Verb semantic consistency check for all PP adjuncts. Semantic Roles may undergo a transformation in the semantic mapping from LF to Infons in case of idiomatic expressions, and in case of unexpressed Obligatory Arguments;

B. each CLAUSE in a DAG is turned into a well-formed-formula with restricted unscoped quantification, positive literals, no variables except for those introduced at a syntactic level. The LF transducer looks for the starting node which is the propositional node, where mood and tense are available. All arguments are searched first, by traversing the DAG looking for grammatical functions; only semantically referential arguments are considered, non referential ones are erased (notice that f-structures containing semantic role Form (corresponding to “there” existential subject, or pleonastic “it”) are excluded from LF;

C. after argument f-structures are mapped in appropriate logical terms, i.e. by computing internal adjuncts and/or arguments, the algorithm looks for sentence level adjuncts. In LFG, both arguments and adjuncts may be computed in two different ways: open or predicative, closed or non-predicative. These two syntactic constructions receive a different treatment in the semantics: in particular, closed adjuncts have only a modifying import on the Event variable associate to the main predicate. On the contrary, open adjuncts have both an Event variable and an argument variable which they modify: this information is represented in f-structure by the presence of an internal Subject variable functionally controlled by the governing head NP. An example will be reported below and discussed in details;

D. each wff is an expression of logical form which is made up of a predicate and a number of arguments, "p(arg$_1$, ..., arg$_n$), where 'p' is a constant and 'arg' may be a complex term. A term is made up of a quantifier, a variable and a restriction, "term(quant,var,restr)" where the quantifier may be a real natural language quantifier existing in a NP or a time operator like "time"; the variable is a syntactic index assigned to the phrase in the f-structure representation by the parser; the restriction is the structure on which the quantifier/operator takes scope which might coincide with the phrase or clause of f-structure representation or may be a logical expression built for that aim at logical form level, as happens for time formulas. In order to reach an adequate representation for our discourse model we generate a generic "situation" predicate for each tensed clause we compute, and we build a complex term for time-aspect representation.

E. In LF representation we use syntactic indices derived directly from f-structure. The mapping onto semantic representation has two effects: syntactic indices are substituted by semantic ones, where they already exist – and this is the case of anaphora resolution. In case of new entities, new semantic indices are generated.

F. Each term is enriched with Semantic Role information. As said above, Semantic Roles may undergo a transformation in the semantic mapping from LF to Infons in case of idiomatic expressions, and in case of unexpressed Obligatory Arguments. In the former case semantically empty arguments are assembled together to produce a non compositional meaning representation (see THERE_BE, as opposed to the BE predicate). The latter case regards both agentless passives and the Receiver or Goal of ditransitive verbs.

The following is the LF for the first utterance: John went into a restaurant.

```
wff(situation, [
    wff(go, [term(definite, sn2, wff(isa, [sn2, john])),
              term(definite, sn5, wff(isa, [sn5, restaurant])),
              term(event, f5, wff(and, [wff(isa, [f5, ev]),
                     wff(time, [f5, term(definite, t1,
                        wff(and, [wff(isa, [t1, tloc]),
                          wff(past, [t1])]))])]))])) 'term-event'])])
```

Generic 'isa' relations are introduced into wffs for NP's and the quantifier is represented by the translation of the content of the NP's specifier. Indefinite NP are turned into 'definite' operators in case no scope ambiguity in the clause may arise due to the absence of ambiguity inducing quantifiers. Tense specifications are transformed into complex terms with a semantic operator that translates the contents of aspect after the computations that have transformed the lexical static value of aspect into its corresponding dynamic propositional import. We use three different operators: event, process, state. These operators then have a complex restriction, represented by a conjoined number of wffs, where we indicate both the location in time - tloc - and its specificity.

This LF representation is then converted into a situational semantic representation where syntactic identifiers are turned into semantic identifiers and all logical predicates are omitted except for the conjunction 'and'. Semantic identifiers might be derived from the discourse model in case the linguistic form represents an entity already existing or known to the world of the DM. Situation semantics builds infons for each unit of information constituting the situation denoted by the proposition being represented in the formula. In addition, for each individual or set entity we record the semantic role already assigned at f-structure level by the grammar. A generic 'arg' is associated to arguments of time predicate. Notice then that a polarity argument has been added at the end of each expression.

```
sit(event, id4, go,
    [ind(definite, id3,
       and([infon(att, infon8, isa, [id3, john], [], 1)]), agent),
        ind(indefinite, id2,
              and([infon(att, infon9, isa,
                [id2, restaurant], [], 1)]), locat)],
              and([infon(att, infon10, isa, [id4, ev], [], 1),
           infon(att, infon13, time, [id4,
                 ind(definite, id5,
                 and([infon(att, infon11, isa, [id5, tloc], [], 1),
          infon(att, infon12, past, [id5], [], 1)]), arg)], [], 1)]), 1)
```

Finally the content of this representation is asserted in the DM as a set of 'facts' or 'sits' in case they are not already present. Factuality for situational types - events, processes and states - is computed from propositional level informational and semantic features. Semantic roles inherited from f-structure representation make explicit, in a declarative way, semantic relations which are not computed in the LF.

The final translation in the DM introduces the objects of our ontology which, as we said above are made up of the following literals: fact, sit, loc, ind, set, card, in, class. The structure of each situation semantic expression is different according to their semantic role: loc, locations has no polarity and no spatiotemporal location indices; ind, in, card, set, class are type denotators and have no internal structure. Fact and sit have an internal structure which is made up of the following arguments:
- an infon ranked number; a relational type specifier; a list of argument expressed as a feature role:identifier; a polarity, spatiotemporal indices.
Facts and sits corresponding to main propositional relations have no infon: in its place they have a semantic unique identifier.

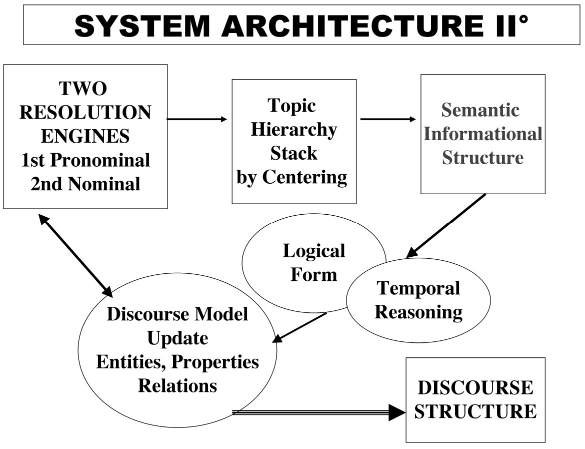
SYSTEM ARCHITECTURE II°
TWO RESOLUTION ENGINES 1st Pronominal 2nd Nominal
Topic Hierarchy Stack by Centering
Semantic Informational Structure
Logical Form
Temporal Reasoning
Discourse Model Update Entities, Properties Relations
DISCOURSE STRUCTURE

Fig.2 GETARUNS' Discourse Level Modules

2.2 Building the Discourse Model

In Situation Semantics where reality is represented in Situations which are collections of Facts: in turn facts are made up of Infons which information units characterised as follows:
Infon(Index, **Relation**(Property),
List of Arguments - with Semantic Roles,
Polarity - 1 affirmative, 0 negation,
Temporal Location Index,
Spatial Location Index)

In addition Arguments have each a semantic identifier which is unique in the Discourse Model and is used to individuate the entity uniquely. Also propositional facts have semantic identifiers assigned thus constituting second level ontological objects. They may be "quantified" over by temporal representations but also by discourse level operators, like subordinating conjunctions. Negation on the contrary is expressed in each fact. All entities and their properties are asserted in the DM with the relations in which they are involved; in turn the relations may have modifiers - sentence level adjuncts and entities may also have modifiers or attributes. Each entity has a polarity and a couple of spatiotemporal indices which are linked to main temporal and spatial locations if any exists; else they are linked to presumed time reference derived from tense and aspect computation. Entities are mapped into semantic individual with the following ontology: on first occurrence of a referring expression it is asserted as an INDividual if it is a definite or indefinite expression; it is asserted as a CLASS if it is quantified (depending on quantifier type) or has no determiner. Special individuals are ENTs which are associated to discourse level anaphora which bind relations and their arguments. Finally, we have LOCs for main locations, both spatial and temporal. If it has a cardinality determined by a number, it is plural or it is quantified (depending on quantifier type) it is asserted as a SET and the cardinality is simply inferred in case of naked plural, i.e. in case of collective nominal expression it is set to 100, otherwise to 5. On second occurrence of the same nominal head the semantic index is recovered from the history list and the system checks whether it is the same referring expression:

- in case it is definite or indefinite with a predicative role and no attributes nor modifiers nothing is done;
- in case it has different number - singular and the one present in the DM is a set or a class nothing happens;
- in case it has attributes and modifiers which are different and the one present in the DM has none, nothing happens;
- in case it is quantified expression and has no cardinality, and the one present in the DM is a set or a class, again nothing happens.

In all other cases a new entity is asserted in the DM which however is also computed as being included in (a superset of) or by (a subset of) the previous entity.

2.3 GETARUNS at work

As said at the beginning, this paper is concerned with an hybrid approach to text understanding which is based on the concurrent use of complete NLP techniques with shallow and partial ones in heavily linguistically demanding tasks such as the one posed by summarization and question answering. This approach should be taken as a proposal in line with current NLP research in unrestricted texts that assumes that partial processing can be more suitable and useful for better satisfaction of certain requirements. In particular, morphological analysis is a prerequisite in order to better cope with Out of Vocabulary Words(OOW) by means of guessing techniques based on morphological rules; statistical processing – or finite state automata as is the case with our system - is assumed to be essential for tagging disambiguation. As to syntactic parsing, robust approaches should be adopted in order to allow for structure building in the case of local failures. Eventually, whenever required, partial semantic interpretation has to be carried out in order to execute anaphora resolution and a Discourse Model is built with a limited number of relations and properties. Partial semantic interpretation means that not all semantic relations will be detected and encoded appropriately in a sense better specified below. Nonetheless, what is captured by partial analysis can still be useful to carry out such important tasks as anaphora resolution at discourse level and a rough evaluation of entity relevance in order to better grasp what topic has been the most relevant one.

Consider now a simple sentence like the following:

1. John went into a restaurant

This might be represented by Ternary Expressions (Katz, 1997) as follows:
<John go restaurant>
<GO <SUBJ John>, <OBL restaurant>>
GETARUNS represents the same sentence in different manners according to whether it is operating in Complete or in Partial modality. In turn the operating modality is determined by its ability to compute the current text: in case of failure the system will switch automatically from Complete to Partial modality.

The system will produce the following representations:

```
loc(infon2, id1, [arg:main_tloc, arg:tr(f1_r01)])
loc(infon3, id2, [arg:main_sloc, arg:restaurant])
ind(infon4, id3)
fact(infon5, inst_of, [ind:id3, class:man], 1, univ, univ)
fact(infon6, name, [john, id3], 1, univ, univ)
ind(infon7, id4)
fact(infon8, isa, [ind:id4, class:restaurant], 1, id1, id2)
fact(infon9, inst_of, [ind:id4, class:place], 1, univ, univ)
fact(id5, go, [agent:id3, locat:id4], 1, tes(f1_r01), id2)
fact(infon12, isa, [arg:id5, arg:ev], 1, tes(f1_r01), id2)
fact(infon13, isa, [arg:id6, arg:tloc], 1, tes(f1_r01), id2)
fact(infon14, past, [arg:id6], 1, tes(f1_r01), id2)
fact(infon15, time, [arg:id5, arg:id6], 1, tes(f1_r01), id2)
```

So in case of failure at the Complete level, the system will switch to Partial and the representation will be deprived of its temporal and spatial location information as follows:

```
ind(infon4, id3)
fact(infon5, inst_of, [ind:id3, class:man], 1, univ, univ)
fact(infon6, name, [john, id3], 1, univ, univ)
ind(infon7, id4)
fact(infon8, isa, [ind:id4, class:restaurant], 1, id1, id2)
fact(infon9, inst_of, [ind:id4, class:place], 1, univ, univ)
fact(id5, go, [agent:id3, locat:id4], 1, univ, id2)
```

In order to test the performance of the system in text understanding we refer to such application fields as Question/Answering and Summarization. They are by far the best benchmark for any system that aims at showing how good the semantic mapping has been.

We will show how GETARUNS computes the DM by presenting the output of the system for the "Maple Syrup" text made available by Mitre for the ANLP2000 Workshop(see Hirschmann et al, 1999). Here below is the original text which is followed by the DM only relatively to the linguistic material needed to answer the five questions, though.

How Maple Syrup is Made

Maple syrup comes from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a "sugar" maple tree.

Sugar maple trees make sap. Farmers collect the sap. The best time to collect sap is in February and March. The nights must be cold and the days warm.

The farmer drills a few small holes in each tree. He puts a spout in each hole. Then he hangs a bucket on the end of each spout. The bucket has a cover to keep rain and snow out. The sap drips into the bucket. About 10 gallons of sap come from each hole.

Discourse Model for sentences 6 and 7

6. Farmers collect the sap

```
class(infon100, id28)
fact(infon101, inst_of, [ind:id28, class:man], 1, univ, univ)
fact(infon102, isa, [ind:id28, class:farmer], 1, univ, id8)
fact(id29, collect, [agent:id28, theme_aff:id24], 1, tes(f1_es6), id8)
fact(infon105, isa, [arg:id29, arg:ev], 1, tes(f1_es6), id8)
fact(infon106, isa, [arg:id30, arg:tloc], 1, tes(f1_es6), id8)
fact(infon107, pres, [arg:id30], 1, tes(f1_es6), id8)
during(tes(f1_es6), tes(f1_es5))
includes(tr(f1_es6), univ)
```

7. The best time to collect sap is in February and March

```
ind(infon112, id31)
fact(infon113, inst_of, [ind:id31, class:substance], 1, univ, univ)
fact(infon114, isa, [ind:id31, class:sap], 1, univ, id8)
in(infon115, id31, id24)
ind(infon116, id32)
fact(infon117, best, [ind:id32], 1, univ, id8)
fact(infon118, inst_of, [ind:id32, class:time], 1, univ, univ)
fact(infon119, isa, [ind:id32, class:time], 1, univ, id8)
set(infon120, id33)
card(infon121, 2)
fact(infon122, inst_of, [ind:id33, class:time], 1, univ, univ)
fact(infon123, isa, [ind:id33, class:[march, February]], 1, univ, id8)
fact(id35, collect, [agent:id28, theme_aff:id31], 1, tes(finf1_es7), id8)
fact(infon126, isa, [arg:id35, arg:ev], 1, tes(finf1_es7), id8)
fact(infon127, isa, [arg:id36, arg:tloc], 1, tes(finf1_es7), id8)
fact(infon128, nil, [arg:id36], 1, tes(finf1_es7), id8)
fact(infon130, [march, February], [arg:id32], 1, univ, id8)
fact(id37, be, [prop:id35, prop:infon130], 1, tes(f1_es7), id8)
fact(infon131, isa, [arg:id37, arg:st], 1, tes(f1_es7), id8)
fact(infon132, isa, [arg:id38, arg:tloc], 1, tes(f1_es7), id8)
fact(infon133, pres, [arg:id38], 1, tes(f1_es7), id8)
during(tes(f1_es7), tes(f1_es6))
includes(tr(f1_es7), univ)
```

3. Question-Answering

Coming now to Question Answering, the system accesses the DM looking for relations at first then for entities : entities are searched according to the form of the focussed element in the User DataBase of Question-Facts as shown below with the QDM for the first question:

User Question-Facts Discourse Model

```
q_loc(infon3, id1, [arg:main_tloc, arg:tr(f1_free_a)])
q_ent(infon4, id2)
q_fact(infon5, isa, [ind:id2, class:who], 1, id1, univ)
q_fact(infon6, inst_of, [ind:id2, class:man], 1, univ, univ)
q_class(infon7, id3)
q_fact(infon8, inst_of, [ind:id3, class:coll], 1, univ, univ)
q_fact(infon9, isa, [ind:id3, class:sap], 1, id1, univ)
q_fact(infon10, focus, [arg:id2], 1, id1, univ)
q_fact(id4, collect, [agent:id2, theme_aff:id3], 1, tes(f1_free_a), univ)
q_fact(infon13, isa, [arg:id4, arg:pr], 1, tes(f1_free_a), univ)
q_fact(infon14, isa, [arg:id5, arg:tloc], 1, tes(f1_free_a), univ)
q_fact(infon15, pres, [arg:id5], 1, tes(f1_free_a), univ)
```

The system knows that the « focus » argument is « who » with semantic id, id2, and is an entity belonging to the semantic class of « man », this latter informantion being derived from the syntactic structure of the corresponding sentence where the interrogative pronoun has bound an empty category in the SUBJect of the verb « COLLECT » of the main clause : this in turn has allowed the parser to pass the selectional restrictions associated in the lexicon with the corresponding lexical frame for the verb « COLLECT ». Search of the answer is performed by looking into the DM for the best Infon that matches the question: at first, the system looks for the same relation « collect », then it looks for the entity corresponding to the semantic role of the Focus in the question, the Agent. If the first action doesn't succeed, the well-known « semantic bottleneck » will cause the system to search for synonyms in the WordNet synset at first, then in a more generic dictionary (2 million correlations for

some 30,000 entries) of quasi-synonyms or concepts belonging to the same semantic field.
Then, the system tries to pick up the entity that is the Agent, which in our case is id28 (as shown in the DM for sentence 6), by searching the entity ontological identifiers – set, ind, ent. When the corresponding fact is found, the predicate (FARMER) is passed to the Generator that builds the reply sentence.

As to the current text, it replies correctly to all questions. As to question 4, at first the system takes « come from » to be answered exhaustively by sentence 14 ; however, seen that « hole » is not computed with a « location » semantic role, it searches the DM for a better answer which is the relation linguistically expressed in sentence 9, where « holes » are drilled « in each tree ». The « tree » is the Main Location of the whole story and « hole » in sentence 9 is inferentially linked to « hole » in sentence 14, by a chain of inferential inclusions. In fact, come_from does not figure in WordNet even though it does in our generic dictionary of synonyms. As to the fifth question, the system replies correctly.

1. Who collects maple sap? (Farmers)
2. What does the farmer hang from a spout? (A bucket)
3. When is sap collected? (February and March)
4. Where does the maple sap come from? (Sugar maple trees)
5. Why is the bucket covered? (to keep rain and snow out)

Another possible « Why » question could have been the following : « why is the tree called a "sugar" maple tree », which would have received the appropriate answer seen that the corresponding sentence has received an appropriate grammatical and semantic analysis. In particular, the discourse deictic pronoun « This » has been bound to the previous main relation « use » and its arguments, so that they can be used to answer the « Why » question appropriately.

There is not enough space here to comment in detail the parse and the semantics (but see Delmonte 2000d); however, as far as anaphora resolution is concerned, the Higher Module computes the appropriate antecedent for the big Pro, i.e. the empty SUBject of the infinitive in sentence n. 7, where the collecting action would have been left without an agent. This resolution of anaphora is triggered by the parser decision to treat the big Pro as an arbitrary pronominal and this information is stored at lexical level in the subcategorization frame for the name « time ».

With question n.4 the text only makes available information related to « maple syrup ». As said above, we start looking for relations, and the « come from » relation has a different linguistic description as SUBJect/ Theme_Unaffected argument – i.e. « SAP » -, what we do is to try and see whether there is some inferential link between « sap » and « syrup » in WordNet. This fails, seen that WordNet does not link the two concepts explicitly. However both are classified as « substance » thus allowing the required inference to be fired – both are also taken as synonyms in our generic dictionary. The final question does not constitute a problem seen that the relation «cover» has become a semantic relation and is no longer a noun or a verb. Also worth noting is the fact that the question is not a real passive, but a quasi-passive or an ergative construction, so no agent should be searched for. Our conclusion is that the heart of a Q/A system should be a strongly restrictive pipeline of linguistically based modules which alone can ensure the adequate information for the knowledge representation and the reasoning processes required to answer natural language queries.

3.1 Answering Generic Question

An important issue in QA is answering generic questions on the "aboutness" of the text, questions which may be answered by producing appropriate headlines or just a title. In our system, given the concomitant work of anaphora resolution modules and the semantic mapping into predicate-argument structures, this can be made as follows. The system collapses all entities and their properties, relations and attributes, after the text has been fully analysed, by collecting them for each ontological type under each semantic identifier. At the same time, each semantic id receives a score for topichood thus allowing a ranking of the entities. Here below we list the most relevant entities of the text reported above:

```
entity(set,id8,30,facts([
card(infon23, id8, 5),
fact(infon24, sugar_maple, [ind:id8], 1, T, P),
fact(infon25, inst_of, [ind:id8, class:plant_life], 1, T, P),
fact(infon26, isa, [ind:id8, class:tree], 1, T, P),
fact(id11, come, [actor:id2, locat:id8], 1, T, P),
fact(id25, make, [agent:id8, theme_aff:id23, patient:id24], 1, T,
P)])).

entity(class,id30,77,facts([
fact(infon114, inst_of, [ind:id30, class:man], 1, T, P),
fact(infon115, isa, [ind:id30, class:farmer], 1, T, P),
fact(id39, drill, [agent:id30, theme_aff:id38], 1, T, P),
fact(id42, put, [agent:id30, theme_aff:id41, locat:id38], 1, T, P),
fact(id48, hang, [agent:id30, theme_aff:id44], 1, T, P)])).

entity(ind,id13,10,facts([
in(infon48, id13, id9),
fact(infon46, inst_of, [ind:id13, class:substance], 1, T, P),
fact(infon47, isa, [ind:id13, class:sugar], 1, T, P),
fact(id14, make, [agente:id2, tema_aff:id13], 1, T, P),
fact(*, inst_of, [ind:id13, class:maple], 1, T, P),
fact(*, isa, [ind:id13, class:maple], 1, T, P),
fact(*, isa, [ind:id13, class:sugar_maple], 1, T, P),
fact(*, of, [arg:id10, specif:id13], 1, T, P)])).
```

Where starred facts are inherited by the inclusion relation specified by the "in" semantic predicate. For instance, the fact constituted by a "specifying" relation between "sugar" and "maple" as
fact(infon34, of, [arg:id10, specif:id9], 1, univ, univ)
becomes a starred fact inherited by id13 in force of the inclusion relation,
in(infon48, id13, id9)

In this way, an appropriate answer to the question "What is the text about" can be generated directly from the entity list by picking up relations and properties of the most relevant individuals,sets and classes (Delmonte, 2000).

4. The Experiment

We downloaded the only freely available corpus annotated with anaphoric relations, i.e. Wolverhampton's Manual Corpus made available by Prof. Ruslan Mitkov on his website. The corpus contains text from Manuals at the following address,
http://clg.wlv.ac.uk/resources/corpus.html

To compare our results with the SGML documents we created a Perl script that extracted all referring expressions and wrote the output into a separate file. The new representation of the SGML files looked now like a list of records each one denoted by an index a dash and the text of the referring expression. In case of complex referring expressions we had more than one index available and so we translated the complex referring expression into a couple or a triple of records each one denoted by its index. The final results were 75% F-measure - complete results are published in (Delmonte, 2003b).

5. Conclusions

Results reported in the experiment above have been aimed to show the ability of the system to cope with what has always been regarded as the toughest task for an NLP system to cope with, that of reference resolution which is paramount in any system of Q/A. We have not addressed the problem of summarization for lack of space: however hints have been addressed by the issue of answering Generic Questions.

We are currently experimenting with automatic ontology building from the DM into a Protegé database which is then used to answer queries from the web (Delmonte, 2003b). By weaving natural language into the basic fabric of the Semantic Web, we can begin to create an enormous network of knowledge easily accessible by both machines and humans alike. Furthermore, we believe that natural language querying capabilities will be a key component of any future Semantic Web system. By providing "natural" means for creating and accessing information on the Semantic Web, we can dramatically lower the barrier of entry to the Semantic Web. Natural language support gives users a whole new way of interacting with any information system, and from a knowledge engineering point of view, natural language technology divorces the majority of users from the need to understand formal ontologies. As we have tried to show in the paper, this calls for better NLP tools where a lot of effort has to be put in order to allow for complete and shallow techniques to coalesce smoothly into one single system. GETARUNS represents such a hybrid system and its performance is steadily improving.

6. References

Abney, A. 1996. Part-of-Speech Tagging and Partial Parsing, in Ken Church et al., eds. Corpus-Based Methods in Language and Speech, Kluwer Academic Publishers, Dordrecht.

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

Delmonte R., 2003a. Getaruns: a hybrid system for summarization and question answering, in Proc. Workshop "NLP for Question Answering" in EACL, Budapest, 21-28.

Delmonte R., D. Bianchi. 2002. From Deep to Partial Understanding with GETARUNS*, Proc. ROMAND 2002*, Università Roma2, Roma, 57-71.

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

Delmonte R. 2000. Generating from a Discourse Model, *Proc. MT-2000*, BCS, Exeter, 25-1/10.

Delmonte R., 2003b. The Semantic Web Needs Anaphora Resolution, Proc.Workshop ARQAS, 2003 International Symposium on Reference Resolution and Its Applications to Q/A and Summarization, Venice, Ca' Foscari University, 25-32.

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

Hirschman, L. Marc Light, Eric Breck, & J. D. Buger. 1999. Deep Read: A reading comprehension system. In *Proc. A CL '99*.University of Maryland.

Katz, B. 1997. Annotating the World Wide Web using natural language. In RIAO '97.