Evaluating Students' Summaries with GETARUNS

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

Evaluating summaries is currently performed by the use of statistically-based tools which lack any linguistic knowledge and are unable to produce grammatical and semantic judgements (Landauer et al., 1997). However, summary evaluation needs precise linguistic information with a much finer-grained 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, 2003). The heart of the system is a rule-based top-down parser, which uses an LFG oriented grammar organization. Lately, a less constrained version of the parser for the application field of text summarization has been developed, which allows the system to recover gracefully from failures. To this end, the parser is couple with another concurrent parsing processes: a partial or shallow parse is always produced and used to recover from complete failures. 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. Evaluation taps information from the Discourse Model and uses Predicate Argument Structures (PAS) to detect students' understanding of the text to summarize. It also uses the output of the Anaphora Resolution Module to check for most relevant topics in the text which the student should have addressed in his/her summary. The system uses a Topics-Stack while processing the text in order to corefer referential expressions: The Topic-Stack Hierarchy gauges nominal heads as either Main, Secondary or Potential Topic. This grading is used as a score that allows the system to detect the most relevant entities in the text at the end of the computation.

1. Introduction

Currently available summary and essay evaluation systems are basically based on statistical and mathematical procedures which are used to assess students linguistic abilities. We are here referring to such tools as LSA-based Summary Street®. Latent Semantic Analysis (Landauer T. et al, 1997) is a statistical theory of meaning which tells the student "…how well your summary covers the information in the original text. It will tell you if your summary is too long for a good summary." It is unable to check for grammaticality issues, neither coherence nor cohesiveness is checked, and what is worse no semantic soundness can be checked. LSA techniques simply allow looking for semantic similarities through comparison of most frequent content words with a knowledge of the surrounding most frequent content words at sentence and paragraph level. LSA does not take into account content words with frequency of occurrence equal-lower than two; it does not take into account the order in which content words cooccur. It capture text similarity in terms of differences in word choice among different texts. It seems to me to be a too poor a way of characterizing meaning: on the contrary, the authors speak of LSA as a tool that "captures a great deal of the similarity of meanings expressed in discourse", and use that as "…the basis for performing automated scoring of essays". Seen that LSA does not take into account word order and discards such important elements as negation items it follows that there is no way to tell whether simple coocurrence indicates similarity of meaning. The experiment commented in the same article is very uncouth. At first glance, Landauer et al. seem to be concerned only with destroying what has gone on in Syntactic Theory and its contribution to the determination of meaning. However in the following paragraph they come up with the opposite statement:

> The fact that LSA can capture as much of meaning as it does without using word order shows that the mere combination of words in passages constrains overall meaning very strongly. How can this be? In addition to the contrary theoretical presumptions mentioned earlier, various intuitive and rational arguments suggest that such representations must fall far short of extracting as much meaning from text as do human readers. For instance, the following two sentences are identical for LSA, but have very different meanings for a human reader: "It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem."; "That day the office manager, who was drinking, hit the problem sales worker with a bottle, but it was not serious… Nonetheless, what such examples prove is only that a method that ignores word order

cannot always render completely correct comprehension. [ibid:417]

Here it would seem that Landauer et al. recognize that LSA cannot possible capture the same amount of meaning from a text as human beings do, and the conclusions partially confirm this. If LSA is unable to distinguish sentences telling just the opposite of one another, than looking into what has gone on in Syntactic and Semantic Theory might be useful.

Another line of research is represented by the work of J.Burstein et al. who make use of NLP tools without however trying to build any kind of logical form or semantic representation. In their latest work they take advantage of discourse structure as built from syntactic structure. Discourse structure is used to determine whether essays produced by a test-taker contain as their most relevant parts topics which are regarded important for a given prompt. This seems to be a more effective line of research that tackles a specific task: that of assessing whether the essay produced covers most of the important topics and also presents them in a salient manner. This latter aspect can be captured nicely by the traversal of rhetorical structures built on the basis of syntactic structure as the authors comment in their work. So even though this is just one of the aspects to be looked into by a human assessor and does in no way cope with the issue of semantic soundness, semantic coherence and cohesion, it may still be a useful tool in itself.

2. GETARUNS: a system for text understanding

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 (Delmonte, 2000). The system work in Italian and English.

The system is based on LFG theoretical framework (Bresnan, 2000) 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 S.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.

2.1 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. For a core portion of the training corpus, some 700 sentences encoding most grammatical structures of English, the system achieves 90% correct parse with the topdown parser and a remaining 9% with the bottomup parser.

To test the parser performance we used the "Grameval 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 show that the Top-Down Parser covers 80% of the overall texts; the Bottom-Up Parser is responsible approximately for 18%, while the knowledge-poor cascaded parser covers the remaining 2%. 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.

2.2 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.

3. 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.

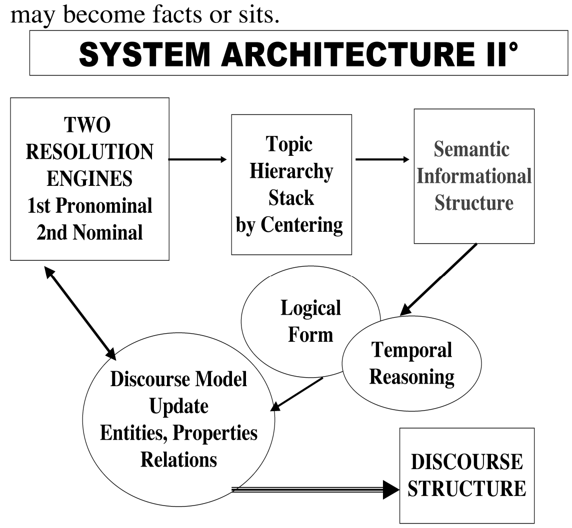

Fig.1 GETARUNS' Discourse Level Modules

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; 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.

3.1 The algorithm for temporal interpretation

In our model all entities and relations are assigned a spatiotemporal location which is made up of two indices: one index is bound to the main temporal location in the case of individuals. As to relations and modifiers of the main relation, the index for the time location is derived from the event/state time, i.e. TES. In turn, this index is in a given logical relation with the TR(Reference Time) of the previous text span, which is related to the discourse focus TF: it may be BEFORE, AFTER, MEETS, OVERLAP etc. with it. The reference of the current TR is furthermore asserted to be included in some main temporal location which could be present in the previous text span. Thus, individuals receive directly a time index which bind their existence in the world and their properties, such as roles, or attributes, to some main temporal location; in case of names associated to individuals, we assign them to a universal location, in other words, we assume that in the domain specified by the text that name is used unambiguously to refer to a single well defined individual. This bipartition is quite natural: entities such as individuals or sets are assumed to be in a given time location for a longer stretch than the actual relation in which they are currently involved. The latter might be a punctual event, thus having a very short duration in time. Event properties are thus different from entities properties as to their temporal location. As we shall see in this chapter, event properties are basically derived from the aspectual classification associate to the predicate, and only partly from the semantic role associated to the arguments of the predicate.

Temporal interpretation is computed compositionally from the local features associated to the verb tense and the static lexical label associated to aspect, plus a number of relevant information like definiteness of the subject NP, and of the Object NP, plus their number. These features contribute to assigning a first dynamic aspectual label to the sentence in which the verbal predicate is analysed. This allows the system to compute local temporal relation intersententially, taking into adequate account all the elements which contribute to the definition of temporal relations; these elements are essentially taken from Reichenbach's proposal. This computation is added to the f-structure of each clause, where it will have to face the complexity arising from the presence of subordinate, coordinate and complement clauses. Finally, the result of the utterance temporal computation is passed on to the discourse module for temporal reasoning which is based on Allen's proposal(Allen, 1983). At this level, temporal intervals are generated and are attached at nodes in a cluster. The logical notation introduced by Allen is perspicuous enough to allow for the overall representation of discourse structure, where reference intervals are generated by taking into account local logical relations.

In Allen's paper(1983) an interval-based temporal logic is introduced, together with a computationally effective reasoning algorithm. The temporal representation takes the notion of temporal interval as primitive; a notion of reference intervals is also introduced which is used to control the amount of deduction performed automatically by the system. By using reference intervals, the amount of computation involved when adding a fact in the temporal knowledge base can be controlled in a predictable manner, comments the author(ibid.,833). In our algorithm we also use Allen's explicit notation to express all the possible relations that can hold between two intervals by means of 13 relations.

Our implementation of Allen's algorithm takes as input Reichenbach's relations as they are independently computed at f-structure level. However, one of the main component's of the overall computation is the relation intervening between Time Focus and the Time Reference of the current utterance. Time Focus is a notion introduced by Webber(1985) which captures temporal information relevant for discourse and text analysis: it can be viewed as a stack which has on top the TR on which the discourse is currently focussing for any given utterance. The main call of our version of Allen's algorithm instantiates the current TF on top of the stack and determines the relation intervening between the TF and the TR of the current utterance. Subsequently, it determines the relation intervening between the TRs and it is asserted in the graph or cluster of relations and TRs computed so far. In order to do this, constraints are checked on whether two adjacent nodes may be included or not in the same Reference Interval. Finally the TF is updated if necessary.

3.2 Inferring Spatial Locations

Spatial locations in narrative texts and newspaper articles might vary according to the presence of one or more participants. In the stories that we worked on with GETARUN, both situations may be found: Avveduti's story is all located in Verona, even though more than one participant is present. On the contrary, The Story of the Three Little Pigs has a very high number of locations which may change according to what each character is doing in the story. The story sets the main spatial location right at the beginning, in the "countryside". In the second utterance the story introduces a new character which not necessarily is in the same location: however, this is explicitly stated. Thus, the same location is asserted and maintained up to the point in the story in which houses have been built and they can become suitable locations for the little pigs. At first the current location for the set of two lazy little pigs is their little house. When the perspective moves to the wolf, the spatial location returns to the previous one in which the wolf was located in preceding stretch of text, and we highlighted this by italicizing the wolf. Then again a change in perspective, and the two lazy little pigs are the current topics of discourse: no location is actually expressed and the system assumes that the current location is the one that they share with other character in focus, the wolf. Suddenly, they manage to "get in" somewhere. We understand that they entered their little brother's little house, even though this is not explicitly stated in the text. In order to infer the current location, we recover some previous relation in which the two lazy little pigs where involved as subject/actor directing themselves towards some location: this is the "running" event in one of the previous utterances. In this way, we may safely infer that now they entered the house they were running towards, i.e. their brother's little house. Again, a change of perspective and the wolf is highlighted: he is not in the same location as the little pigs, and no explicit location is present in the current utterance. Thus we assume that the wolf is again in the same location he was before. Notice now that a location is expressed, "to the ground": but this is not to be computed as a new location, in as far as the ground is included as part of the main current location, "the countryside". This clearly constitutes external world knowledge, which is present in our domain knowledge. Finally, the perspective moves again to the three little pigs which are in the solid brick house: this is the location previously associated to the two lazy little pigs. Recency, is activated in order to compute the right association.

At this point we might state the following rule for spatial locations:

Inferential Rule for Spatial Locations

Topic-consistency: A location should be consistent with the current topic

A. Old Locations : there is a previous main location to be asserted

CASE 1 - No location is explicitly expressed in the current utterance

- The current location is topic-inconsistent, an old location should be recovered
- Infer a suitable location for the current topic from the previous stretch of text
- The topic changes but the previous location is deictically assigned to the new topic

CASE 2 - There is a location explicitly expressed in the current utterance

- The new location has already been associated with the current topic in the previous stretch of text
- The new location is inferentially included in the current main location
- The new location is inferentially a suitable location which came into existence in the previous stretch of text

B. New Locations : a new main location is asserted

1. Assert a new location whenever it is explicitly stated in the current utterance.

4. Getaruns at work

We will show how GETARUNS computes the DM by presenting the output of the system for a quite wellknown text: The Story of Three Little Pigs in Italian – which we include in the Appendix - and show how the system produces Focussed Summaries, lists of most relevant characters and PAS for each such character. We will only present a portion of the sentence by sentence creation of the DM for lack of space.

C'erano una volta tre fratelli porcellini che vivevano felici nella campagna.

```
loc(infon2, id1, [arg:main_tloc, arg:volta])
loc(infon3, id2, [arg:main_sloc, arg:campagna])
set(infon4, id3)
card(infon5, id3, 3)
fact(infon6, fratello, [ind:id3], 1, id1, id2)
fact(infon7, inst_of, [ind:id3, class:animale_cibo], 1, univ, univ)
fact(infon8, isa, [ind:id3, class:porcellino], 1, id1, id2)
ind(infon9, id2)
fact(infon10, inst_of, [ind:id2, class:[luogo]], 1, univ, univ)
fact(infon11, isa, [ind:id2, class:campagna], 1, id1, id2)
set(infon12, id4)
card(infon13, id4, 3)
fact(infon14, inst_of, [ind:id4, class:uomo], 1, univ, univ)
fact(infon15, isa, [ind:id4, class:fratello], 1, id1, id2)
fact(id5, vivere, [actor:id3, locativo:id2], 1, tes(f2_po01), id2)
fact(infon19, isa, [arg:id5, arg:st], 1, tes(f2_po01), id2)
fact(infon20, isa, [arg:id6, arg:tloc], 1, tes(f2_po01), id2)
fact(infon21, imp, [arg:id6], 1, tes(f2_po01), id2)
fact(infon22, time, [arg:id5, arg:id6], 1, tes(f2_po01), id2)
fact(infon23, felice, [arg:id5], 1, tes(f2_po01), id2)
fact(id7, esserci, [tema_nonaff:id3], 1, tes(f3_po01), id2)
fact(infon25, isa, [arg:id7, arg:st], 1, tes(f3_po01), id2)
fact(infon26, isa, [arg:id8, arg:tloc], 1, tes(f3_po01), id2)
```

fact(infon27, imp, [arg:id8], 1, tes(f3_po01), id2)
fact(infon28, time, [arg:id7, arg:id8], 1, tes(f3_po01), id2)
fact(infon29, isa, [arg:id1, arg:volta], 1, tes(f3_po01), id2)
fact(infon30, nil, [arg:id7, non_punct:id1], 1, tes(f3_po01), id2)
includes(tr(f3_po01), id1)
contains(tes(f3_po01), td(f3_po01))
Questi allora, per proteggersi dal lupo, decisero di costruirsi ciascuno una casetta.
fact(infon79, poss, [porcellino, id3, id18], 1, id9, id2)
set(infon80, id18)
card(infon81, id18, 3)
fact(infon82, piccolo, [ind:id18], 1, id9, id2)
sit(infon83, isa, [ind:id18, class:casa], 1, id9, id2)
fact(infon84, inst_of, [ind:id18, class:cosa], 1, univ, univ)
fact(id19, costruire, [agente:id3, tema_eff:id18], 1, tes(finf18_po03), id2)
fact(infon90, isa, [arg:id19, arg:ev], 1, tes(finf18_po03), id2)
fact(infon91, isa, [arg:id20, arg:tloc], 1, tes(finf18_po03), id2)
fact(infon92, pres, [arg:id20], 1, tes(finf18_po03), id2)
fact(infon93, time, [arg:id19, arg:id20], 1, tes(finf18_po03), id2)
fact(id21, decidere, [actor:id3, prop:id19], 1, tes(f1_po03), id2)
fact(infon94, isa, [arg:id21, arg:st], 1, tes(f1_po03), id2)
fact(infon95, isa, [arg:id22, arg:tloc], 1, tes(f1_po03), id2)
fact(infon96, pass_rem, [arg:id22], 1, tes(f1_po03), id2)
fact(infon97, time, [arg:id21, arg:id22], 1, tes(f1_po03), id2)
fact(infon98, allora, [arg:id21], 1, tes(f1_po03), id2)
fact(id23, proteggere, [agente:id3, esperiente:id3, malef:id10], 1, tes(finf15_po03), id2)
fact(infon100, isa, [arg:id23, arg:pr], 1, tes(finf15_po03), id2)
fact(infon101, isa, [arg:id24, arg:tloc], 1, tes(finf15_po03), id2)
fact(infon102, pres, [arg:id24], 1, tes(finf15_po03), id2)
fact(infon103, time, [arg:id23, arg:id24], 1, tes(finf15_po03), id2)
fact(infon104, per, [arg:id21, purpose:id23], 1, tes(f1_po03), id2)
includes(tr(f1_po03), id9)
overlap(tes(finf15_po03), tes(f3_po01))
…….
Ma ecco che improvvisamente il lupo apparve alle loro spalle.
set(infon279, id61)
card(infon280, id61, 5)
fact(infon281, inst_of, [ind:id61, class:cosa], 1, univ, univ)
fact(infon282, isa, [ind:id61, class:spalla], 1, id9, id2)
fact(id62, apparire, [actor:id10], 1, tes(f1_po07), id2)
fact(infon284, isa, [arg:id62, arg:st], 1, tes(f1_po07), id2)
fact(infon285, isa, [arg:id63, arg:tloc], 1, tes(f1_po07), id2)
fact(infon286, pass_rem, [arg:id63], 1, tes(f1_po07), id2)
fact(infon287, time, [arg:id62, arg:id63], 1, tes(f1_po07), id2)
fact(infon288, improvvisamente, [arg:id62], 1, tes(f1_po07), id2)
fact(infon291, a, [arg:id62, nil:id61], 1, tes(f1_po07), id2)
fact(infon296, perf, [arg:id64, [ecco_che]:id62], 1, id9, id2)
includes(tr(f1_po07), id9)
during(tes(f1_po07), tes(f3_po01))

Questo intanto si leccava già I baffi pensando al suo prossimo pasto così invitante e saporito.
fact(infon336, poss, [lupo, id10, id77], 1, id9, id2)
ind(infon337, id77)
fact(infon338, prossimo, [ind:id77], 1, id9, id2)
in(infon339, id77, id3)
fact(infon340, isa, [ind:id77, class:pasto], 1, id9, id2)
fact(id78, leccare_baffo, [esperiente:id10], 1, tes(f1_po09), id2)
fact(infon342, isa, [arg:id78, arg:st], 1, tes(f1_po09), id2)
fact(infon343, isa, [arg:id79, arg:tloc], 1, tes(f1_po09), id2)
fact(infon344, imp, [arg:id79], 1, tes(f1_po09), id2)
fact(infon345, time, [arg:id78, arg:id79], 1, tes(f1_po09), id2)
fact(infon346, intanto, [arg:id78], 1, tes(f1_po09), id2)
fact(infon347, già, [arg:id78], 1, tes(f1_po09), id2)
fact(id80, pensare, [actor:id10, tema:id77], 1, tes(fgerund51_po09), id2)
fact(infon351, isa, [arg:id80, arg:st], 1, tes(fgerund51_po09), id2)
fact(infon352, isa, [arg:id81, arg:tloc], 1, tes(fgerund51_po09), id2)
fact(infon353, pres, [arg:id81], 1, tes(fgerund51_po09), id2)
fact(infon354, time, [arg:id80, arg:id81], 1, tes(fgerund51_po09), id2)
fact(infon355, coincide, [arg:id78, arg:id80], 1, tes(f1_po09), id2)
includes(tr(f1_po09), id9)
during(tes(fgerund51_po09), tes(f3_po01))

The portion of the DM we have included is intended to show how the system copes with important semantic issues such as the representation of negation, anaphora resolution and PASs.

5. Evaluating Summaries with PAS and Topic Relevance Scoring

The context in which the system is used for evaluation is with summary creation exercises in the area of newspaper stories for students of Foreign Languages and Literatures. The student is presented with a short text of approximately 800-1000 words length and is asked to shorten it to no more than 200 words by picking up the most relevant portions of it. He/she is also prompted to use the lexicon present in the text as much as possible. Our evaluation system will be able to tap WordNet for lexical items not present in the original text, but used by the student to create the summary : it will be looking for synonyms, hypernyms, hyponyms and other semantic relations in order to uderstand whether the new lexical item has been introduced with the appropriate meaning relation. However, this is done, starting from a complete semantic representation of the text, which constitues the knowledge representation needed to cope with all semantic relations expressed in the text. Without such a representation, searching WordNet for synonyms would be a nonsensical operation. Systems such as the ones used for « bag-of-words » approaches would be unable to look for the appropriate meaning by simply selecting most frequent content words in context.
Coming now to the actual tools used for evaluation, the system will produce a list of most relevant entities at the end of the computation, which includes for each entity all the properties asserted in the DM. In the case of the current text, this is the list generated:

List of most important characters of the text :

I due fratelli porcellini, grassi oziosi pigri teneri

Il lupo, terribile

Il fratello jimmi porcellino, grasso tenero

La casa, piccola

The list is organized in order of relevance so that the most relevant character comes first. The system may then generate PAS as an answer to a question related to one specific character – this in case it is working in tutoring and coaching modality. The question is computed by the system in the same way in which the text has been processed, except that it will not be part of the same DM . Suppose we query the system with the question « che cosa è accaduto al lupo ?/what happened to the wolf », this would be represented as follows:

User Question Discourse Model

q_ind(infon1, id1)
q_fact(infon2, inst_of, [ind:id1, class:animale_feroce], 1, univ, univ)
q_fact(infon3, isa, [ind:id1, class:lupo], 1, univ, univ)
q_fact(infon4, isa, [ind:id1, class:lupo], 1, univ, univ)
q_ind(infon5, id2)
q_fact(infon6, isa, [ind:id2, class:cosa], 1, univ, univ)
q_fact(infon7, inst_of, [ind:id2, class:cosa], 1, univ, univ)
q_fact(infon8, focus, [arg:id2], 1, univ, univ)
q_fact(id3, accadere, [tema_nonaff:id2, tema_nonaff:id1], 1, univ, univ)
q_fact(infon11, isa, [arg:id3, arg:ev], 1, univ, univ)
q_fact(infon12, isa, [arg:id4, arg:tloc], 1, univ, univ)
q_fact(infon13, pass_pross, [arg:id4], 1, univ, univ)
q_fact(infon14, time, [arg:id3, arg:id4], 1, univ, univ)
q_fact(infon18, perf, [arg:id5, ask:id3], 1, univ, univ)

There are three inportant elements of meaning representation that would have to be represented in the PAS list in order to gauge student's understanding of the text : Negation, Modality, the Narrative Sequence or Temporal Ordering of events as narrated by the text. The latter semantic element relies on the Temporal Reasoning component of the system, which in turns, relies on the aspectual-tense lexical and linguistic analysis elaborated by the parser, as well as the James Allen algorithm as explained above. The Narrative Sequence is in turn responsible for weak Causal Relation between events. Information about Causality and Relevance is reported in another level of linguistic representation we call Discourse Structure which is organized in terms of Rhethorical Structures and Semantic Relations. Stretches of discourse are assigned a Discourse Move and may thus be interpreted as being dependent of other structures or simply be independent and be attached to the Root. For instance the one below is a snapshop of a small portion of the Story of the Three Little Pigs :

```
C19/Questo intanto si leccava già i baffi C20/pensando al suo prossimo pasto così invitante e saporito.
```

up:to(7-16)
clause:9-19
topics:[secondary:id32:porcellino, expected:id9:lupo]
main_fact:leccarsi([id9:lupo], 1, id2)
ref_int:tint(tes(f4_po8), [f3_po9])
temp_rel:started_by(tes(f3_po09), tes(f4_po08))
disc_rel:elaboration
disc_str:17-[18, 19]
disc_dom:subjective
p_o_view:lupo

same_level:from(9-19)
clause:9-20
topics:[secondary:id32:porcellino, expected:id9:lupo]
main_fact:pensare([id9:lupo, id32:porcellino], 1, id2)
ref_int:tint(tes(f4_po8), [f3_po9, fger4_po9])
temp_rel:after(tes(fger4_po09), tes(f4_po08))
disc_rel:elaboration
disc_str:17-[18, 19, 20]
disc_dom:objective
p_o_view:lupo

The same argument applies to the next UP move, which is again a topic alternation but it has no consequences on the temporal sequence of events.

```
C21/Finalmente i porcellini riuscirono a raggiungere la loro casetta C22/e vi si chiusero dentro C23/sbarrando la porta.
```

up:to(8-18)
clause:10-21
topics:[main:id32:porcellino, secondary:id9:lupo]
main_fact:riuscire([id32:porcellino, id77:raggiungere], 1, id35)
ref_int:tint(tes(f5_po10), [f4_po8])
temp_rel:after(tes(f3_po10), tes(f4_po08))
disc_rel:result
disc_str:18-[21]
disc_dom:objective
p_o_view:narrator

down:down(10-21)
clause:10-22
topics:[main:id32:porcellino, secondary:id9:lupo]
main_fact:chiudere([id32:porcellino, id32:porcellino], 1, id35)
ref_int:tint(tes(f5_po10), [f4_po8, f5_po10])
temp_rel:after(tes(f5_po10), tes(f3_po10))
disc_rel:evidence
disc_str:21-[22]
disc_dom:objective
p_o_view:narrator

same_level:from(10-22)
clause:10-23
topics:[main:id32:porcellino, secondary:id9:lupo]
main_fact:sbarrare([id32:porcellino, id76:porta], 1, id35)
ref_int:tint(tes(f5_po10), [f4_po8, f5_po10, fger61_po10])
temp_rel:after(tes(fger61_po10), tes(f5_po10))
disc_rel:result
disc_str:18-[21, 22, 23]
disc_dom:objective
p_o_view:narrator

Clauses may be label either as UP,DOWN,SAME_LEVEL or ROOT(the first clause of the text). Knowledge about dependent discourse relation is contained in the slot « disc_str ». Each UP move defines a stretch of discourse where a new topic is introduced and may thus be used to ask the student : « what happened when the little pigs entered their house ? ».
The following would be the list of PAS extracted from the DM representation and generated :

List of PAS for the « wolf » entity :

vivere [actor:lupo, locativo:luogo]
proteggere [agente:porcellino, esperiente:porcellino, malef:lupo]
apparire [actor:lupo]
leccare_baffo [esperiente:lupo]
pensare [actor:lupo, tema:porcellino]
pensare [actor:lupo, tema:porcellino, prop:penetrare]
mettersi [actor:lupo, prop:osservare]
notare [esperiente:lupo, prop:essere]
soffiare [agente:lupo]
avere_calcagna [esperiente:porcellino, agente:lupo]
arrivare [tema_aff:lupo]
riempire [agente:lupo, tema_aff:polmone, materia:aria]
cominciare [agente:lupo, prop:soffiare]
accasciarsi [esperiente:lupo, prop:in]

5.1 Evaluating GETARUNS approach to Essay Evaluation

We have just started experimenting with the system in simulation, using summaries produced by students on paper and comparing the judgements expressed by human tutors with the output of the algorithm: the results are encouraging. As proposed in Burstein (2000:56), we compare test summary with the full text and then evaluate the level of relevance by gauging the number of predicate-argument structures used and their semantic appropriateness.

6. References

Allen J.F.(1983), Maintaining Knowledge about Temporal Intervals, *Communications of the ACM*, 26,11, 832-843.

Bianchi D., Delmonte R., Temporal Logic in Sentence and Discourse, in Proc.SIMAI'96, 226-228.

Bianchi D., R. Delmonte(1999), Reasoning with A Discourse Model and Conceptual Representations, Proc. VEXTAL, Unipress, pp. 401-411.

Delmonte R., Contextual Reasoning and Inferential Processing, in Proc.SIMAI'96, 229-231.

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.

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

Burstein, J. and Wolska, M. (2003). Toward Evaluation of Writing Style: Overly Repetitious Word Use in Student Writing. In Proceedings of the 11th Conference of the EACL. Budapest,35-42.

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

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

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. (2000), Generating from a Discourse Model, *Proc. MT-2000*, BCS, Exeter, pp.25-1/10.

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

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E., (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In M. G. Shafto & P. Langley (Eds.), Proceedings of the 19th annual meeting of the Cognitive Science Society (pp. 412-417). Mahwah, NJ: Erlbaum.

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

Webber B. L.(1988), Tense as discourse anaphor, Computational Linguistics, 113-122.

Appendix

The Story of the Three Little Pigs

C'erano una volta tre fratelli porcellini che vivevano felici nella campagna. Nello stesso luogo però viveva anche un terribile lupo che si nutriva proprio di porcellini grassi e teneri. Questi allora, per proteggersi dal lupo, decisero di costruirsi ciascuno una casetta. Il maggiore, Jimmi, che era saggio, lavorava di buona lena e costruì la sua casetta con solidi mattoni e cemento. Gli altri, Timmy e Tommy, pigri e oziosi se la sbrigarono in fretta costruendo le loro casette con la paglia e con pezzetti di legno. I due porcellini pigri passavano le loro giornate cantando e suonando una canzone che diceva chi ha paura del lupo cattivo. Ma ecco che improvvisamente il lupo apparve alle loro spalle. Aiuto, aiuto gridarono i due porcellini e cominciarono a correre più veloci che potevano verso la loro casetta per sfuggire al terribile lupo. Questo intanto si leccava già I baffi pensando al suo prossimo pasto così invitante e saporito. Finalmente i porcellini riuscirono a raggiungere la loro casetta e vi si chiusero dentro sbarrando la porta. Dalla finestra cominciarono a deridere il lupo cantando la solita canzoncina: chi ha paura del lupo cattivo. Il lupo stava intanto pensando al modo di penetrare nella casa. Esso si mise ad osservare attentamente la casetta e notò che non era davvero molto solida. Soffiò con forza un paio di volte e la casetta si sfasciò completamente. Spaventatissimi I due porcellini corsero a perdifiato verso la casetta del fratello. "Presto, fratellino, aprici! Abbiamo il lupo alle calcagna". Fecero appena in tempo ad entrare e a tirare il chiavistello. Il lupo stava già arrivando deciso a non rinunciare al suo pasto. Sicuro di abbattere anche la casetta di mattoni il lupo si riempì i polmoni di aria e cominciò a soffiare con forza alcune volte. Non ci fu niente da fare. La casa non si mosse di un solo palmo. Alla fine esausto il lupo si accasciò a terra. I tre porcellini si sentivano al sicuro nella solida casetta di mattoni. Riconoscenti i due porcellini oziosi promisero al fratello che da quel giorno anche essi avrebbero lavorato sodo.

Once upon a time there were three little pigs who lived happily in the countryside. But in the same place lived a wicked wolf who fed precisely on plump and tender pigs. The little pigs therefore decided to build a small house each, to protect themselves from the wolf. The oldest one, Jimmy who was wise, worked hard and built his house with solid bricks and cement. The other two, Timmy and Tommy, who were lazy settled the matter hastily and built their houses with straw and pieces of wood. The lazy pigs spent their days playing and singing a song that said, "Who is afraid of the big bad wolf?" And one day, lo and behold, the wolf appeared suddenly behind their backs. "Help! Help!", shouted the pigs and started running as fast as they could to escape the terrible wolf. He was already licking his lips thinking of such an inviting and tasty meal. The little pigs eventually managed to reach their small house and shut themselves in, barring the door. They started mocking the wolf from the window singing the same song, "Who is afraid of the big bad wolf?" In the meantime the wolf was thinking a way of getting into the house. He began to observe the house very carefully and noticed it was not very solid. He huffed and puffed a couple of times and the house fell down completely. Frightened out of their wits,the two little pigs ran at breakneck speed towards their brother's house. "Fast, brother, open the door! The wolf is chasing us!" They got in just in time and pulled the bolt. Within seconds the wolf was arriving, determined not to give up his meal. Convinced that he could also blow the little brick house down, he filled his lungs with air and huffed and puffed a few times. There was nothing he could do. The house didn't move an inch. In the end he was so exhausted that he fell to the ground. The three little pigs felt safe inside the solid brick house. Grateful to their brother, the two lazy pigs promised him that from that day on they too would work hard.