Building Domain Ontologies from Text Analysis: An Application for Question Answering

Rodolfo Delmonte

Department of Language Sciences
Università Ca' Foscari
Ca' Garzoni-Moro - San Marco 3417 - 30124 Venezia

Abstract. In the field of information extraction and automatic question answering access to a domain ontology may be of great help. But the main problem is building such an ontology, a difficult and time consuming task. We propose an approach in which the domain ontology is learned from the linguistic analysis of a number of texts which represent the domain itself. NLP analysis is done with GETARUNS system. GETARUNS can build a Discourse Model and is able to assign a relevance score to each entity. From Discourse Model we extract best candidates to become concepts in the domain ontology. To arrange concepts in the correct hierarchy we use WordNet taxonomy. Once the domain ontology is built we reconsider the texts to extract information. In this phase the entities recognized at discourse level are used to create instances of the concepts. The predicate-argument structure of the verb is used to construct instance slots for concepts. Eventually, the question answering task is performed by translating the natural language question in a suitable form and use that to query the Discourse Model enriched by the ontology.

1 Introduction

Textual information available on the web constitutes a gigantic encyclopaedia and is growing continuously, so it is important to have an effective tool to classify and extract information from non structured documents for the realization of Knowledge Management Systems and for efficient question answering systems [1,2,3,4].
The Semantic Web initiative [5] offers many basic languages to represent semantics. From the RDF [6] language, useful for annotating resources, to the OWL (Web Ontology Language) [7] proposed by the W3C to describe domain knowledge. Knowledge based and question answering systems often use annotation to add semantics to unstructured text. Nevertheless manual annotation of text is not applicable to the scale of the Web. For example the START QA system [8,9] is built on the basis of NLP techniques of text analysis but needs also manual annotation of texts. The KIM platform [10] is oriented towards a "Semantic Web" Information Extraction (IE) and allows semantic indexing, annotation and retrieval. It combines Natural Language Processing Tools with Semantic Web technologies to obtain annotation with respect to concepts and instances of a semantic repository. This platform is mainly oriented to annotate Named Entities (person, organization, location, dates etc.) and uses a manually constructed ontology. The OntoGenie

Delmonte R. (2006).
Building Domain Ontologies from Text Analysis: An Application for Question Answering.
In *Proceedings of the 3rd International Workshop on Natural Language Understanding and Cognitive Science*, pages 3-16
DOI: 10.5220/0002483200030016

platform [11] uses a domain ontology to automatically annotate web texts using WordNet as a bridge.
Two problems arise in these approaches. The main problem is building the domain ontology, a difficult and time consuming task. Moreover the same work must be redone when switching from one domain to another. Besides, it is not easy to associate keywords or terms extracted from documents with WordNet synsets.
The OntoLearn system [12] attempts to learn domain ontologies directly from web pages and documents. The system extracts important terms and relies on WordNet to arrange them in a hierarchy. The extraction of terms is based on statistical measure of "Domain Relevance" and "Domain Consensus".

1.1 Large-scale Syntactic-Semantic Indexing

Although full syntactic and semantic analysis of open-domain natural language text is beyond current technology, a number of papers have been recently published [29,30,31] showing that, by using probabilistic or symbolic methods, it is possible to obtain dependency-based representations of unlimited texts with good recall and precision. Consequently, we believe it should be possible to augment the manual-annotation-based approach with automatically built annotations by extracting a limited subset of semantic relations from unstructured text. In short, shallow/partial text understanding on the level of semantic relations, an extended label including Predicate-Argument Structures and other syntactically and semantically derivable head modifiers and adjuncts. This approach is promising because it attempts to address the well-known shortcomings of standard "bag-of-words" (BOWs) information retrieval/extraction techniques without requiring manual intervention: it develops current NLP technologies which make heavy use of statistically and FSA based approaches to syntactic parsing.
GETARUNS [12,13,14], a text understanding system (TUS), developed in collaboration between the University of Venice and the University of Parma, can perform semantic analysis on the basis of syntactic parsing and, after performing anaphora resolution, builds a discourse model. In addition, it uses a centering algorithm to individuate the topics or discourse centers which are weighted on the basis of a relevance score. This discourse model is used to individuate the candidates to become concepts in the domain ontology.
Using Getaruns we have built a prototype question answering system based on matching semantic relations derived from the question with those derived from the corpus of texts.
The steps involved in the process are:

a) Select a number of texts representative of a particular domain. (This may be done by filtering the results of a search engine).
b) Analyze these texts and extract the relevant terms.
c) Build a domain ontology, extablishing a hierarchy among these terms.
d) Populate the ontology with instances derived from predicate-argument structures and semantic relation obtained by Getaruns.
e) Build an Augmented Discourse Model corresponding to the thus instantiated ontology.

This approach is very well suited for another important application domain:

question/answering with students working at text/essay/summary understanding. In this case the domain is defined by the text and its related reference field: concepts and links in WordNet may in this case be directly filtered.
When dealing with web-based applications, the domain needs to be recovered from the query itself: this may lead to failures. In particular, in step 1 above, the texts filtered from the web – in the number of five/ten snippets – may be wrongly related to other domains, different from the ones required by the query. The creation of the base ontology is then totally misled and will determine problems in finding the right answer. We discuss here below how Discourse Model can help in the choice of the appropriate hierarchical relations.
This paper is organized as follows: in section 2 below we present GETARUNS, the NLP system and the Upper Module of GETARUNS; in section 3 we briefly describe the ontology builder; in section 4 we show one short example of text understanding question/answering; in section 5 we describe an experiment with web-based question answering.

2 GETARUNS – The TUS

GETARUN, the system for text understanding, produces a semantic representation in xml format, in which each sentence of the input text is divided up into predicate-argument structures where arguments and adjuncts are related to their appropriate head. Consider now a simple sentence like the following:
(1) John went into a restaurant
GETARUNS represents this sentence in different manners according to whether it is operating in Complete or in Shallow 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/Shallow modality.
The system will produce a representation inspired by Situation Semantics where reality is represented in Situations which are collections of Facts: in turn facts are made up of Infons which are 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 each Argument has 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 and a performative operator if needed. Negation on the contrary is expressed in each fact.
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. In the current version of the system, we use Complete modality for tasks which involve short texts (like the students summaries and text understanding queries), where text

analyses may be supervisioned and updates to the grammar and/or the lexicon may be needed. For unlimited text from the web we only use partial modality. Evaluation of the two modalities are reported in a section below.

2.1 The Parser and the Discourse Model

As said above, the query building process needs an ontology which is created from the translation of the Discourse Model built by GETARUNS in its Complete/Partial Representation. GETARUNS, 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. The system works in Italian and English.
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. A shallow or partial parser 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.

2.2 Lexical Information

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. We show here below the architecture of the system. The grammar is equipped with a lexicon containing a list of 30000 wordforms derived from Penn Treebank. 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 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. Subcategorization information and Semantic Roles are then derived from a carefully adapted version of FrameNet and VerbNet.
Our “training” corpus is made up of 200,000 words and contains a number of texts taken from different genres, portions of the UPenn Treebank corpus, test-suits for

grammatical relations, and sentences taken from COMLEX manual. An evaluation carried out on the Susan Corpus related GREVAL testsuite made of 500 sentences has been reported lately [15] to have achieved 90% F-measure over all major grammatical relations. We achieved a similar result with the shallow cascaded parser, limited though to only SUBJect and OBJect relations on LFG-XEROX 700 corpus.

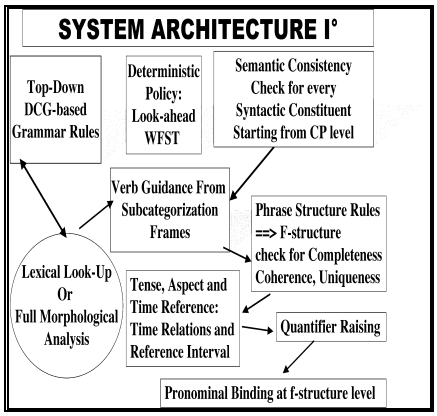

Fig. 1. GETARUNS' LFG-Based Parser.

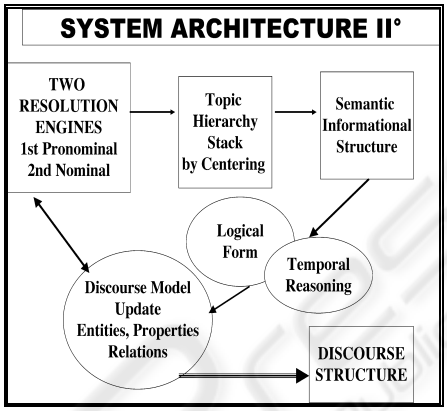

Fig. 2. GETARUNS' Discourse Level Modules.

2.3 The Upper Module

GETARUNS, as shown in Fig.2 has a 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. 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.

In each infon, Arguments have each a semantic identifier which is unique in the DM and is used to individuate the entity. 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 individuals 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. Whenever there is cardinality determined by a digit, its number is plural or it is quantified (depending on quantifier type) the referring expression is asserted as a SET. Cardinality is simply inferred in case of naked plural: 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.
The upper module of GETARUNS has been evaluated on the basis of its ability to perform anaphora resolution and to individuate referring expressions [16], with a corpus of 40,000 words: it achieved 74% F-measure.

3 The Ontology Builder

This module uses the output of Getaruns. In the first phase it builds a top level ontology. The ontology contains a limited number of classes: thing, man, event, state, location. The ranked list of the entities of the discourse produces the terms that are candidates to become classes of the ontology.
Verb predicates become subclass of events or states. Subcategorization and semantic role of verbs are used to define slots of verb classes.
Also the entities found in the discourse model with an high rank become classes.
To give a hierarchical structure to the classes we use WordNet [27]. From the term we individuate the corresponding WordNet synset and by using the hypernim link we find the superclass. The superclass may be a top level class like man, location etc. or a class obtained from the discourse model.
The main problem we have to face is to disambiguate words to find the correct synset. This task is a very thorny problem: our approach is that of using semantic categories associated with lexical entries. These semantic categories should match with categories associated to some hypernym of the synset: in case the semantic category is unique no ambiguity problem will ensue. In this case, the disambiguation procedure will only select those hypernyms that satisfy requirements imposed by the unique semantic identifier. When more than one semantic category is present, the synset is recursively searched for a non-empty intersection of concepts which will best satisfy the categories, thus pruning those items that do not match the given set: this is done

taking into account all arguments of the same predicate, and the predicate itself.
At this point we have built a class structure that is representative of the domain of interest.
The ontology can be viewed using an ontology editor. An example of the ontology obtained is shown in fig.3. In particular the event "go" is shown with the two slots "agent" and "location"

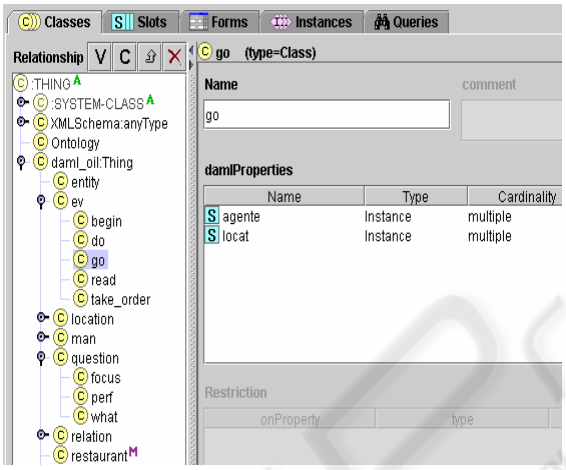

Fig. 3. Screenshot of the class built from the text. The roles of the class GO are shown.

We can now use the texts to populate the instance of the classes. This is made by recursively analysing the discourse structure and examining the individuals and the events and filling their slots. Information gathered in this way is then made available to the Augmented Discourse Model (hence ADM).
As will be discussed below, the ontology building process contributes different properties according to the type of application:
- in Q/A based on text and summary understanding, knowledge of the semantic field of application contributes a powerful disambiguation tool that allows access to WordNet hierarchy in a fully controlled manner. Different instances of the same concepts are neatly identified as cospecifications or coreferring items in the appropriate semantic relation;
- in Q/A based on web search, no preliminary domain information is made available to the Concept Builder to access WordNet and disambiguation is only worked out when enough information is available from the snippets analyzed by GETARUNS. As discussed below, in some cases, however, the ontology is beneficial.

4 Question Answering in Text Understanding

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 [21]. Here below is the original text which is followed by a short excerpt from the DM with the Semantic Database of Entities and Relations of the Text World.

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.

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)

4.1 Discourse Model for the Text Organized Sentence by Sentence

Here below we list an excerpt of the DM related to the most relevant sentences of the above text:

1.How Maple Syrup is Made
loc(infon1, id1, [arg:main_tloc, arg:tr(f2_es1)])
class(infon2, id2)
fact(infon3, Maple, [ind:id2], 1, id1, univ)
fact(infon4, inst_of, [ind:id2, class:edible_substance], 1, univ, univ)
fact(infon5, isa, [ind:id2, class:Syrup], 1, id1, univ)
ind(infon6, id3)
fact(infon7, inst_of, [ind:id3, class:plant_life], 1, univ, univ)
fact(infon8, isa, [ind:id3, class:Maple], 1, id1, univ)
in(infon9, id3, id2)
fact(id5, make, [agent:id2, theme_aff:id4], 1, tes(f2_es1), univ)
fact(infon13, isa, [arg:id5, arg:ev], 1, tes(f2_es1), univ)
fact(infon14, isa, [arg:id6, arg:tloc], 1, tes(f2_es1), univ)
fact(infon15, plu_perf, [arg:id6], 1, tes(f2_es1), univ)
fact(infon16, time, [arg:id5, arg:id6], 1, tes(f2_es1), univ)
fact(infon17, how, [arg:id5], 1, tes(f2_es1), univ)
before(tes(f2_es1), tes(f2_es1))
includes(tr(f2_es1), id1)
……..

7.The best time to collect sap is in February and March
ind(infon110, id32)
fact(infon111, best, [ind:id32], 1, univ, id7)
fact(infon112, inst_of, [ind:id32, class:time], 1, univ, univ)
fact(infon113, isa, [ind:id32, class:time], 1, univ, id7)
set(infon114, id33)
card(infon115, 2)
fact(infon116, inst_of, [ind:id33, class:time], 1, univ, univ)
fact(infon117, isa, [ind:id33, class:[march, February]], 1, univ, id7)
fact(id35, collect, [agent:id28, theme_aff:id24], 1, tes(finf1_es7), id7)
fact(infon118, isa, [arg:id35, arg:ev], 1, tes(finf1_es7), id7)
fact(infon119, isa, [arg:id36, arg:tloc], 1, tes(finf1_es7), id7)
fact(infon120, nil, [arg:id36], 1, tes(finf1_es7), id7)
fact(infon121, [march, February], [arg:id32], 1, univ, id7)
fact(id37, be, [prop:id35, prop:infon130], 1, tes(f1_es7), id7)
fact(infon122, isa, [arg:id37, arg:st], 1, tes(f1_es7), id7)
fact(infon123, isa, [arg:id38, arg:tloc], 1, tes(f1_es7), id7)
fact(infon124, pres, [arg:id38], 1, tes(f1_es7), id7)
fact(infon125, time, [arg:id37, arg:id38], 1, tes(f1_es6), id7)

during(tes(f1_es7), tes(f1_es6))
includes(tr(f1_es7), univ)
…….

12.The bucket has a cover to keep rain and snow out
class(infon218, id59)
fact(infon219, inst_of, [ind:id59, class:thing], 1, univ, univ)
fact(infon220, isa, [ind:id59, class:cover], 1, id53, id7)
fact(infon222, cover, [nil:id54], 1, id53, id7)
fact(id60, have, [actor:id54, prop:infon222, prop:id65], 1, tes(f1_es12), id7)
fact(infon223, isa, [arg:id60, arg:st], 1, tes(f1_es12), id7)
fact(infon224, isa, [arg:id61, arg:tloc], 1, tes(f1_es12), id7)
fact(infon225, pres, [arg:id61], 1, tes(f1_es12), id7)
fact(infon227, isa, [arg:id62, arg:rain], 1, tes(f1_es12), id7)
fact(infon228, isa, [arg:id63, arg:snow], 1, tes(f1_es12), id7)
fact(id65, keep_out, [agent:id54, theme_aff:id64], 1, tes(finf1_es12), id7)
fact(infon229, isa, [arg:id65, arg:pr], 1, tes(finf1_es12), id7)
fact(infon230, isa, [arg:id66, arg:tloc], 1, tes(finf1_es12), id7)
fact(infon231, pres, [arg:id66], 1, tes(finf1_es12), id7)
fact(infon232, time, [arg:id65, arg:id66], 1, tes(f1_es12), id7)
fact(infon233, coincide, [arg:id60, prop:id65], 1, tes(f1_es12), id7)
during(tes(f1_es12), tes(f1_es11))
includes(tr(f1_es12), id53)

4.2 Question-Answering

Coming now to Question Answering, the system accesses the ADM looking at first for relations, and 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)

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 contents expressed in 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 dictionary of synonyms. As to the fifth question, the system replies correctly.
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 output of the parser and the semantics (but see [13;14 ;16]); however, as far as anaphora resolution is concerned, the Higher Module computes the appropriate antecedent for the big Pro of the arbitrary SUBject of the infinitive in sentence n. 7, where the collecting action would have been left without an agent. This 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 ».
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.

5 GETARUNS Approach to WEB-Q/A

Totally shallow approaches when compared to ours will always be lacking sufficient information for semantic processing at propositional level: in other words, as happens with our “Partial” modality, there will be no possibility of checking for precision in producing predicate-argument structures.
Most systems would use some Word Matching algorithm to count the number of words appearing in both question and the sentence being considered after stripping stopwords: usually two words will match if they share the same morphological root after some stemming has taken place. Most QA systems presented in the literature rely on the classification of words into two classes: function and content words. They don't make use of a Discourse Model where input text has been transformed via a rigorous semantic mapping algorithm: they rather access tagged input text in order to sort best matched words, phrases or sentences according to some scoring function. It is an accepted fact that introducing or increasing the amount of linguistic knowledge over crude IR-based systems will contribute substantial improvements. In particular, systems based on simple Named-Entity identification tasks are too rigid to be able to match phrase relations constraints often involved in a natural language query.
We raise a number of objections to these approaches: first objection is the impossibility to take into account pronominal expressions, their relations and properties as belonging to the antecedent, if no head transformation has taken place during the analysis process.
Another objection comes from the treatment of the Question: it is usually the case that QA systems divide the question to be answered into two parts: the Question Target represented by the wh- word and the rest of the sentence; otherwise the words making up the yes/no question are taken in their order, and then a match takes place in order to identify most likely answers in relation to the rest/whole of the sentence except for stopwords.

However, it is just the semantic relations that need to be captured and not just the words making up the question that matter. Some systems implemented more sophisticated methods (notably [22;23;24]) using syntactic-semantic question analysis. This involves a robust syntactic-semantic parser to analyze the question and candidate answers, and a matcher that combines word- and parse-tree-level information to identify answer passages more precisely.

5.1 A Prototype Q/A System for the Web

We experimented our approach over the web using 450 factoid questions from TREC. On a first run the base system only used an off-the-shelf tagger in order to recover main verb from the query. In this way we managed to get 67% correct results, by this meaning that the correct answer was contained in the best five snippets selected by the BOW system on the output of Google API. However, only 30% of the total correct results had the right snippet ranked in position one.
Then we applied GETARUNS shallow on the best five snippets with the intent of improving the automatic ranking of the system and have the best snippet always position as first possibility. Here below is a figure showing the main components for GETARUNS based analysis.

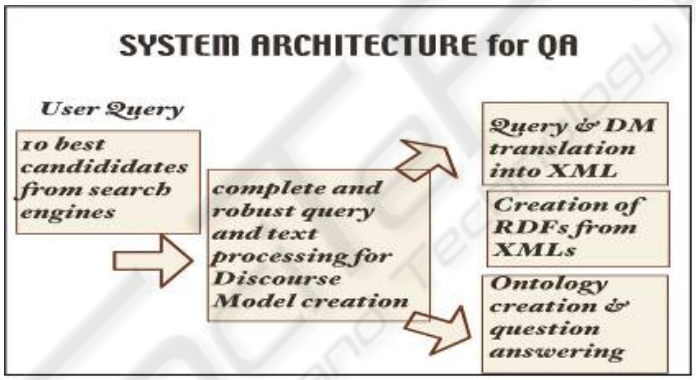

Fig. 4. System Architecture for QA.

We will present two examples and discuss them in some detail. The questions are the following ones:
Q: Who was elected president of South Africa in 1994?
A: Nelson Mandela
Q: When was Abraham Lincoln born?
A: Lincoln was born February_12_1809
The answers produced by our system are indicated after each question. Now consider the best five snippets as filtered by the BOWs system:

who/WP was/VBD elected/VBN president/NN of/IN south/JJ africa/NN in/IN 1994/CD
Main keywords: president south africa 1994
Verb roots: elect
Google search: elected president south africa 1994
1.On June 2, 1999, Mbeki, the pragmatic deputy president of South Africa and leader of the African National Congress, was elected

```
president in a landslide, having already assumed many of Mandela's
governing responsibilities shortly after Mandela won South Africa's
first democratic election in 1994.
2.Washington ? President Bill Clinton announced yesterday a doubling in
US assistance South Africa of $600-million (R2 160-million) over three
years, and said his wife Hillary would attend Nelson Mandela's
inauguration as the country's first black president.
3.Nelson Mandela, President of the African National Congress (ANC),
casting the ballot in his country's first all-race elections, in April
1994 at Ohlange High School near Durban, South Africa.
4.Newly-elected President Nelson Mandela addressing the crowd from a
balcony of the Town Hall in Pretoria, South Africa on May 10, 1994.
5.The CDF boycotted talks in King William's Town yesterday called by
the South African government and the Transitional Executive Council to
smooth the way for the peaceful reincorporation of the homeland into
South Africa following the resignation of Oupa Gqozo as president.
```

Notice snippet n.1 where two presidents are present and two dates are reported for each one: however the relation “president” is only indicated for the wrong one, Mbeki and the system rejects it. The answer is collected from snippet no.4 instead. As a matter of fact, after computing the ADM, the system decides to rerank the snippets and use the contents of snippet 4 for the answer. Now the second question:

when/WRB was/VBD abraham/NN lincoln/NN born/VBN
Main keywords: abraham lincoln
Verb roots: bear
Google search: abraham lincoln born

```
1. Abraham Lincoln was born in a log cabin in Kentucky to Thomas and
Nancy Lincoln.
2. Two months later on February 12, 1809, Abraham Lincoln was born in a
one-room log cabin near the Sinking Spring.
3. Abraham Lincoln was born in a log cabin near Hodgenville, Kentucky.
4.Lincoln himself set the date of his birth at feb_ 12, 1809, though
some have attempted to disprove that claim .
5. A. Lincoln ( February 12, 1809 April 15, 1865 ) was the 16/th
president of the United States of America.
```

In this case, snippet n.2 is selected by the system as the one containing the required information to answer the question. In both cases, the answer is built from the ADM, so it is not precisely the case that the snippets are selected for the answer: they are nonetheless reranked to make the answer available.

6 System Evaluation

After running with GETARUNS, the 450 questions recovered the whole of the original correct result 67% from first snippet.
The complete system has been tested with a set of texts derived from newspapers, narrative texts, children stories. The performance is 75% correct. However, updating and tuning of the system is required for each new text whenever a new semantic relation is introduced by the parser and the semantics does not provide the appropriate mapping. For instance, consider the case of the constituent "holes in the tree", where the syntax produces the appropriate structure but the semantics does not map "holes" as being in a LOCATion semantic relation with "tree". In lack of such a semantic role information a dummy "MODal" will be produced which however will not generate the adequate semantic mapping in the DM and the meaning is lost.

As to the partial system, it has been used for DUC summarization contest, i.e. it has run over approximately 1 million words, including training and test sets, for a number of sentences totalling over 50K. We tested the "Partial" modality with an additional 90,000 words texts taken from the testset made available by DUC 2002 contest. On a preliminary perusal of samples of the results, we calculated 85% Precision on parsing and 70% on semantic mapping. However evaluating full results requires a manually annotated database in which all linguistic properties have been carefully decided by human annotators. In lack of such a database, we are unable to provide precise performance data. The system has also been used for the RTE Challenge and performance was over 60% correct [33].

7 Conclusion

The system we have developed is able to build a domain ontology starting from a deep linguistic analysis of text. The system is also able to answer a number of questions about the analyzed text.
We are completing the experiments with the analysis of a larger number of texts to verify the scalability of our approach. We are also evaluating the quality of the ontologies generated when we shift from a complete to a partial/shallow parsing.

References

1. Brill, E., Lin, J., Banko, M., Dumais, S., & Ng, A.: Data-Intensive Question Answering. In E. M. Voorhees & D. K. Harman (eds.), The Tenth Text Retrieval Conference (TREC 2001). 122-131.
2. Ciravegna, F.: (LP)2, an Adaptive Algorithm for Information Extraction from Web related Texts. In: Proc. IJCAI-2001 Work. on Adaptive Text Extraction and Mining (2001)
3. Riloff, E.: A Case Study in Using Linguistic Phrases for Text Categorization on the WWW. In: AAAI/ICML Work. Learning for Text Categorization (2001)
4. Litkowski, K. C.: Syntactic Clues and Lexical Resources in Question-Answering. In E. M. Voorhees & D. K. Harman (eds.), The Ninth Text Retrieval Conference (TREC-9). NIST Special Publication 500-249. Gaithersburg, MD., (2001) 157-166
5. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American (May 2001)
6. Lassila, O. and Swick, R. (eds.). Resource Description Framework (RDF) model and syntax specification. Available at http://www.w3.org/ TR/1999/REC-rdf-syntax-19990222.
7. OWL http://www.w3.org/2004/OWL/
8. Kahan, J. and Koivunen, M. Annotea: an open RDF infrastructure for shared web annotations, in Proceedings of WWW10 (May 2001)
9. Boris Katz, Jimmy J. Lin, Sue Felshin: The START Multimedia Information System: Current Technology and Future Directions, In Proceedings of the International Workshop on Multimedia Information Systems (MIS 2002)
10. Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, Miroslav Goranov: Towards Semantic Web Information Extraction, Workshop on Human Language Technology for the Semantic Web http://gate.ac.uk/conferences/iswc2003/ proceedings/popov.pdf (2003)

11. Chintan Patel, Kaustubh Supekar, and Yugyung Lee, OntoGenie: Extracting Ontology Instances from WWW, Workshop on Human Language Technology for the Semantic Web http://gate.ac.uk/conferences/iswc2003/proceedings/patel.pdf (2003)
12. R. Navigli and P. Velardi: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites, Computational Linguistics, (30-2), MIT Press, April, (2004)
13. Delmonte R.: Parsing Preferences and Linguistic Strategies, in LDV-Forum - Zeitschrift fuer Computerlinguistik und Sprachtechnologie - "Communicating Agents", Band 17, 1,2, (2000) 56-73
14. Delmonte R.: Parsing with GETARUN, Proc.TALN2000, 7° confèrence annuel sur le TALN, Lausanne, (2000) 133-146
15. Delmonte R.: Generating from a Discourse Model, Proc. MT-2000, BCS, Exeter, (2000) 25-1/10
16. Delmonte R., D. Bianchi: From Deep to Partial Understanding with GETARUNS, Proc. ROMAND 2002, Università Roma2, Roma, (2002) 57-71
17. Delmonte R.: 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 (2002)
18. Delmonte, R.: Getaruns: a Hybrid System for Summarization and Question Answering. In Proc. Natural Language Processing (NLP) for Question-Answering, EACL, Budapest, (2003) 21-28
19. Delmonte R.: Evaluating GETARUNS Parser with GREVAL Test Suite, Proc. ROMAND - 20th International Conference on Computational Linguistics - COLING, University of Geneva, (2004) 32-41
20. Delmonte R.: The Semantic Web Needs Anaphora Resolution, Proc.Workshop ARQAS, 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization, Venice, Ca' Foscari University, (2003) 25-32
21. Hirschman, L. Marc Light, Eric Breck, & J. D. Buger. Deep Read: A reading comprehension system. In Proc. A CL '99.University of Maryland (1999)
22. Hovy, E., U. Hermjakob, & C. Lin.: The Use of External Knowledge in Factoid QA. In E. M. Voorhees & D. K. Harman (eds.), The Tenth Text Retrieval Conference (TREC 2001). (2002) 644-652
23. Litkowski, K. C.: Syntactic Clues and Lexical Resources in Question-Answering. In E. M. Voorhees & D. K. Harman (eds.), The Ninth Text Retrieval Conference (TREC-9). (2001) 157-166
24. Litkowski, K. C.: CL Research Experiments in TREC-10 Question-Answering. In E. M. Voorhees & D. K. Harman (eds.), The Tenth Text Retrieval Conference (TREC 2001). (2002) 122-131
25. Ravichandran, D. & E. Hovy.: Learning Surface Text Patterns for a Question Answering System. Proceedings of the 40th ACL. Philadelphia, PA., (2002) 41-7
26. Schwitter R., D. Mollà, R. Fournier & M. Hess.: Answer Extraction: Towards better Evaluations of NLP Systems. In Proc. Works. Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems, Seattle, (2000) 20-27
27. WordNet www.cogsci.princeton.edu/~wn/
28. Jena http://www.hpl.hp.com/semweb/
29. Dan Klein and Christopher D. Manning: Accurate Unlexicalized Parsing. ACL, (2003) 423-430
30. D. Lin.: Dependency-based evaluation of MINIPAR. In Proceedings of the Workshop on Evaluation of Parsing Systems at LREC 1998. Granada, Spain, (1998)
31. Sleator, Daniel, and Davy Temperley: "Parsing English with a Link Grammar." Proceedings of IWPT '93, (1993)
32. Delmonte R., Sara Tonelli, Marco Aldo Piccolino Boniforti, Antonella Bristot, Emanuele Pianta: VENSES – a Linguistically-Based System for Semantic Evaluation, RTE Challenge Workshop, Southampton, PASCAL - European Network of Excellence, (2005) 49-52