SHUMI

Support to *Hu*man *M*achine *I*nteraction

Project number	
Project title	SHUMI
Deliverable type	Final Report restricted to project participants

Title of deliverable	*Detailed study* accompanying Final Report		
Release number	1.0		
Contractual date of delivery	July 5th, 2004		
Actual date of delivery	September 24th, 2004		
Nature of the deliverable	Technical Report		
Author(s)	Maria Teresa Pazienza Marco Pennacchiotti Michele Vindigni Fabio Massimo Zanzotto	**Organization**	University of Roma Tor Vergata

List of contents

1. Scope of this document

Aim of this document is to analyse how high level technologies on Natural Language Processing (NLP) could support the development of a system for searching information inside textual documents to reply experts' needs during the design of future space missions. In fact, in the context of *Concurrent Design Facility* (CDF), a team of experts from several disciplines jointly works to design future space missions. Such an activity needs a fast and effective interaction of involved disciplines: it requires experts access to several kinds of documentations, among which scientific papers, studies, internal reports, etc., produced by experts of related disciplines all over the World.

A robust system supporting the team of experts in getting, at working time, state of the art information, could ensure consistent and high-quality results in shorter time and improve technical quality of agreed decisions.

Moreover, a system with such an ability and based on domain related linguistic knowledge could be also useful in supporting technical author in producing minutes of the design sessions and editing the final study documentation.

The intended use of the system is a kind of advisor, running basically in the background during a technical meeting about Spacecraft Design; by analyzing input information from resources connected to Spacecraft Design it should refer to specific problems. The system could work on a batch processing form, collecting a certain amount of input information and producing answers in a reasonable time, e.g. by the end of a daily meeting, or by successive morning.

After assessing the availability of relevant written documents, the resources proposed to deal with are firstly study reports and papers. Other information resources like emails between design experts, and short queries addressing the main points and keywords of some required information may be considered in a later study.

The system should retrieve relevant information in the form of e.g. references to related internal studies, citations, relevant web documents, etc. depending on the database (internal or Internet-related) on which the search engine will run. As a starting point the output information should consist of resources available either from internal study reports, or external resources available through the Internet.

It should be worth to pay most attention to the building blocks of the base of the system, i.e. identifying the main features of the space engineering domain and careful designing of the knowledge base, because this would form the basis of possibly later studies.

It is planned to use a machine learning approach to build the KB of the field, then the provided corpus will serve as a starting train collection.

The document is structured as follows. In Sec. 2 a comprehensive description of the state of the art *Information System* approaches and methodologies is presented, in order to make clear the general framework in which the proposed architecture lives. In particular, we present the three main classes of Information Systems: Information Retrieval (Sec.2.1.1), Information Extraction (Sec.2.1.2) and Question Answering (Sec.2.1.3).

In Sec. 3 we propose our work hypothesis, that is, we describe how we would tackle the problem of building the architecture of an automatic advisor able to support the design of a spacecraft mission. We sketch out three different architecture of growing complexity, starting from an Information Retrieval based platform (Sec. 3.1), and adding successively "active" (Sec.3.2), semantic (Sec.3.3) and ontological (Sec.3.4) components. The proposed architectures are strictly tight to the methodologies in use at RTV, and based on the well-assessed approaches we use to deal with Natural Language related problems. Sec.0 is thus intended to give a deep description and understanding of the enabling methodologies and technologies we develop here at RTV and that can be applied to the specific task of building the system architecture. In particular RTV approaches to syntactic and semantic analysis of textual sources (Sec. 4.2) and RTV methodologies to automatically learn domain ontologies (Sec. 4.3) (that both assume a central role in the proposed architectures) are extensively described.

In Sec.5 we finally present current technologies and solutions that could be used to build the architectures: specifically, IR systems (Sec.5.1) and Natural Language processing systems (Sec. 5.2).

2. Introduction

2.1 Natural Language oriented Information Systems

A robust system able to retrieve and propose to experts state of the art information, such as the one we will propose later on this study, should strongly base on methodologies and technologies to extract information from textual documents, that usually represent the specific, to the application, knowledge.

In this section it will thus be proposed a brief introduction to the existing approaches to the extraction of user's information need: *Information Retrieval*, *Information Extraction* and *Question Answering*. While the presented approaches differ with respect to techniques and methodologies adopted, they can all be classified as Natural Language oriented Information Systems, as their common aim is to retrieve (or extract) useful information from a set of informative sources (such as documents) expressed mainly in Natural Language.

Although a clean distinction among the three approaches can not be clearly outlined, as they are strictly intertwined both from the methodological and the technical point of view, they will be separately described, highlighting differences and commonalities.

2.1.1 Information Retrieval

2.1.1.1 Basic Concepts

Information Retrieval (IR) can be roughly defined as *"the discipline that deals with retrieval of unstructured data, especially textual documents, in response to a query or topic statement, which may itself be unstructured […] or which may be structured, e.g. a boolean expression."* [Greengrass, 2002].

In order to define IR more rigorously it can be useful to clarify which is its actual aim, which is the type of user information needs it satisfies and what kind of information sources it uses during the retrieval process. From this point of view it is necessary to underline a first distinction between Information Retrieval and Data Retrieval (DR).

Both IR and DR aim to retrieve a specific user information need, which is specified through a *query*, from a set of *records* containing information. Records can be *structured* or *unstructured*. In the first case records have a well defined syntax, which underlies a definite semantic. Records in a DBMS are classical example of structured data: indeed, each component of a record carries a specific and well defined semantic information. On the other hand, unstructured records don't have such a strict relation between syntax and semantic; moreover syntax itself is not always well defined and the information carried can be highly semantically ambiguous. Example of unstructured records are natural language text (*documents*), audio, video etc.

DR deals with structured records: a user query can be thus intended as a specific and clear expression of the user need, which requires the retrieval of the correct answers, that is, all the records that satisfy the query. Being the information semantically well-defined inside the records, the retrieval process simply consist in matching the query semantic need with the semantic content of the records (this is how a DBMS query works).

IR deals with unstructured data, mainly natural language documents, organized together in a repository called *collection*. The structure of such data is loose and minimal, and usually consist in a rough division in chapters, paragraphs and sentences; at most, documents can be organized using a markup languages such as HTML or SGML. In this framework the task of retrieving useful data becomes more difficult. The user query itself, usually expressed through natural language, can generally remain ambiguous and loosely defined. In order to deal with such vagueness and unstructuredness, document and query representation and interpretation must be carefully investigated.

A further distinction can be made between *Document Retrieval* and *Information Extraction* (IE). Both disciplines can be intended as different type of IR, even though in recent years IE has became an independent field, thus identifying IR with Document Retrieval (we will assume such identification from now on). The aim of Document Retrieval is to retrieve *documents* relevant to a given query, while IE returns to the user the actual *information* responding to the user information need (extracted from one or more documents). Such fundamental distinction implies a differentiation in methodologies and techniques used to retrieve data. IR (Document Retrieval) usually needs only statistical and probabilistic theories in order to satisfy a user query, while IE is based more on texts semantic and syntactic analysis, being its task harder and more specific.

2.1.1.2 Documents representation

Representation of documents (*logical view of the document*) plays a central role in the IR systems, being the main source of information.

The simplest way to represent documents is to generate, manually or automatically, a set of index terms or keywords able to carry the most relevant information embedded in the document. The task can be performed both by human experts or by the IR system itself, through a lexical and syntactic text analysis.

Documents can be also represented as *full text*, that is, via the whole set of words they contain. While this representation is more complete and accurate, it implies high computational costs, that even modern computers can not manage if the document collection is too large.

Between this two extremes other logical view can be taken into consideration, all of which rely on textual analysis of the text collection.

2.1.1.3 The retrieval process

The retrieval process is generally structured in a set of sub-processes:

1. Document collection définition
2. Creation of document logical view
3. Document indexing
4. Query analysis and modification
5. Query processing
6. Retrieved document ranking
7. User feedback and query refinement

1. The first task is related to the identification of the textual document collection over which the retrieval must be performed: documents can consist in a static and limited collection or in a wide and dynamic set of changing data (e.g. Internet). In both of these modalities, it is successively necessary to identify the most suitable document logical view, ranging from full text to index keywords. In most cases an intermediate representation is chosen: each document is broken into a subset of important terms (usually identified with words) derived from a full text analysis. In order to obtain such a subset, different operations are performed over the texts. *Stemming* extract the lexical root of each word, thus reducing to a unique grammatical form all the different lexical expression of a root-word present in the document. A *stop-list* can then be used to eliminate common words that carry a small semantic and discriminative power (like prepositions, articles and so on). *Noun grouping* eliminates verbs, adjectives and adverbs from the document view. At the end of this process, the union of the terms (*index terms* or *keywords*) obtained will represent the document during the retrieval process.
2. In a few IR models, once index terms are identified, it is necessary to assign a numeric weight to each term for every document in the collection; this weight measures how effective the term is in distinguishing the document from the others present in the collection (the same term thus assumes

different weights in every distinct document). Different weighting measures can be adopted, from naïve term frequency to more complex metrics.

3. Document indexing is a crucial step. The *index* essentially contains couples <term, document>, needed during the retrieval phase to quickly understand in which documents of the collection are present the terms expressed in the user's query. The most widely used type of index, thanks to his simplicity and good performance, is the *inverted file*. For each term the file contains an ordered list (*inverted list*) of the identifiers of all the documents that contain the indexed term. The *document id* is also usually associated with the term weight in the document. The performance of the retrieval process strongly depends on a good implementation of the index, that must guarantee a fast retrieval of the query words into the collection.

The retrieval process takes place over the document database represented by the collection of documents and the index. The user specifies her *information need* through a *query*, expressed in a specific query language, that can range from natural language to more strict representations such as regular expressions or a simple list of keywords. In all cases the terms used by the user must convey the semantics of the information need. This is a not trivial step, as the user can define his needs in a wrong or a poor way, compromising the whole retrieval process. Since the translation of the information need into a query is a task completely assigned to the user himself, a good IR system must thus be able to take into consideration this aspect, and to implement specific countermeasures, as techniques of user feedback analysis.

4. Before the query terms are searched into the collection index, they can be further processed by the system, through a set of query operations similar to those previously applied to the documents, or through additional interpretation step, as *query expansion* and *query refinement*, whose aim is both to better define the user need and to improve the retrieval results. Query expansion augment the terms used in the user query, identifying new terms to be searched, using statistical or semantic methods. In the first case new terms are selected analysing the documents retrieved by the simple query and using methods of user feedback. In the second case, new terms usually consists on synonyms of the query terms, or semantically related words, obtained through lexical-semantic resources like Wordnet.
5. Once the query has been expanded and refined, the IR system retrieves the documents that better fit to the user need. Different kind of retrieval methodologies exist and are currently used (statistical, probabilistic and semantic approaches): they will be summarized later on.
6. Retrieved documents are then ranked according to the likelihood of relevance, and finally presented to the user.
7. Sometimes, optionally, the user can give a feedback to the system, selecting the more interesting documents among those retrieved : the system can thus elaborate a further and more precise query based on this information.

2.1.1.4 Evaluation

The performance evaluation of an IR system is strongly related to the concept of *relevance*: a "good" IR system in facts returns to the user a set of "relevant" documents.

As stated before, the user information need is expressed through a query, whose semantic content can not be neither and precisely conveyed nor evaluated by the user itself, due to the intrinsic characteristic of vagueness and ambiguity of natural languages.

The concept of *relevance* itself is thus strongly subjective, as it represents the human judgement the user expresses on how well the retrieved document relate to the user information needs. While a precise definition of relevance is still matter of debate, two measures based on the concept of relevance have been elaborated and are commonly used to measure the performance of IR systems: *precision* and *recall*.

Precision is defined as "the ratio of relevant items retrieved respect to all retrieved items, or the probability given that an item is relevant that it will be retrieved" [Saracevic, 1995]. *Recall* is defined as "the ratio of relevant items retrieved respect to all relevant items in a file (a collection), or the probability given that an item is relevant that it will be retrieved" [Saracevic, 1995].

	Relevant Documents	Non-relevant Documents
Retrieved Documents	A	B
Non-Retrieved Documents	C	D

$$\Pr ecision = \frac{|A|}{|A| + |B|} \qquad \text{Re}call = \frac{|A|}{|A| + |C|}$$

A trade-off between these two measures can be usually intended as the expectation a user has: a set of retrieved documents that contains mostly useful documents and few uninteresting documents (precision), and that contains the widest part of interesting documents present in the whole collection (recall). This trade-off is expressed in the *F-measure*, which is a compromise between precision and recall:

$$F = 1 - \frac{1}{0{,}5\frac{1}{P} + 0{,}5\frac{1}{R}}$$

Since users can be more interest in precision or in recall, a flexible version of F-Measure, called *Effectiveness*, has been elaborated [Rijsbergen, 1979] :

$$E \approx 1 - \frac{1}{\alpha\left(\frac{1}{P}\right) + (1-\alpha)\frac{1}{R}}$$

where P is precision, R is recall, and α a user parameter (from zero to one) that expresses the degree of importance given to P or R.

More sophisticated ways to express Precision and Recall have been implemented, such as *non-interpolated and eleven-point averages* [Greengrass, 2000].

Other ways to evaluate IR system consider other aspects, as: the *efficiency* (average time interval between request and answer), *effort* (involved by the user to obtain request) and *coverage* (relevant material present in the whole collection).

2.1.1.5 Approaches to IR

Two main categories of approaches to Information Retrieval can be identified: *statistical* and *semantic*. Statistical approaches retrieve documents using statistical measures and techniques, applied to both the user query and the collection documents. Semantic approaches, on the other end, use semantic and syntactic methods to better understand the user query and to retrieve the correct documents by looking at semantic information.

Statistical approaches are today much more used than the semantic ones. They can be further divided in: *boolean* , *extended boolean*, *vector space* and *probabilistic approaches*. In all cases documents and query are both intended as a *set of terms*, extracted as full text or by index keywords. Some IR system identify terms in words, while others in groups of a number *n* of consecutive characters, called *n-grams*. Terms in the collection are usually organized in an *index*, as already described.

Boolean approach

In the classical boolean approach the query consist in a set of terms combined with classical boolean operators (such AND, OR, NOT) or ad-hoc operators (like the *proximity* operator that specifies how close two terms must be to satisfy the query). The query is evaluated according to the rules of boolean algebra; using the collection index, all the documents that satisfy the query expression are retrieved.

In such an approach a degree of relevance of the retrieved document can not be expressed: the document can be simply relevant or not-relevant, since the query is expressed by a boolean expression, that can be only *true* or *false*.

The classical Boolean method can be enhanced with the previously described text operation techniques (stemming, etc.) or by query expansion methods (such the use of synonymy information).

Another way to improve the boolean approach is to introduce terms weight (*Extended Boolean Approach*). One of the major drawbacks of the classical method is in fact the absence of a relevance ranking on the retrieved documents, that can be only divided into relevant and not-relevant, while a user would know the *degree* with which a document relates to his query. In order to produce a relevance ranking, some IR systems assign to the terms of the documents a weight, automatically calculated , which expresses the importance of the terms in the specific document (i.e., its usefulness to distinguish the document to other documents in the collection). Several measures have been proposed for assigning weight, and will be briefly described in the next sub-section. Weights are also given to the user query, by the user himself, giving to each term a value proportional to the importance it has for satisfying the information need. Using terms weights the IR system can then elaborate a *similarity measure* between the query and each document, returning to the user a ranked set of relevant documents; one of the more widely used measure is *p-norm*, introduced by [Salton and McGill, 1983].

Extended Boolean Approaches have shown to achieve better results compared to the Classical Boolean Approach and the Vector Space Model Approach (described later).

Vector Space Approach

In vector space approaches each document is represented by a set of terms, each one accompanied with a weight. Terms are extracted using the previously described text operations. The union of the terms extracted from all the documents of the collection can be seen as a *document space* [Salton and McGill, 1983], where each term is a dimension of the space whose value corresponds to the term weight. In this space, a document corresponds to a point, whose coordinates are the weights of the terms it contains.

The user query is expressed by a set of terms with an accompanying weight, or by a natural language expression. In the latter case the query is processed like a document. In this framework similarity among documents and relevance of a document to a query can be thus defined as a distance measure in the document space.

A final note on Vector Space approaches regards the possibilities of expressing in the vector space not only a list of terms, but also syntactical and semantic information, such as the Part of Speech of terms or their semantic generalization. In the first case, when Vector Space Model is applied to a space of terms, viewed as independent and lexical entities, the approach is called *Bag-of-Words* Model. In the latter case, the Vector Space represent not only the terms themselves, but also their syntactic and semantic information and other relation among terms, the approach can be defined as a *Semantic Model*..

This aspect will be further discussed in Sec. 2.1.2 and Sec. 4.3.

Probabilistic Approach

Probabilistic approaches make explicitly use of formal probability theory in order to estimate a probability of relevance for each document. Even if they are not widely used as the classical statistical approaches, they achieve performance that are comparable to the boolean methods.

Usually, documents and query are represented as set of terms, and the relevance computation consist in the calculation of the conditional probability *P(D|R)* that a document D is observed given an event R (i.e. the document is relevant to the given query). How this probability is calculated depends on the underlying model used (usually it is a function of the probability of occurrence of the terms in the relevant and non-relevant documents).

2.1.1.6 The TREC Conference

The Text REtrieval Conference (TREC) [Voorhees, 2003a] is the annual meeting, held since 1991 and sponsored the NIST (US National Institute of Standard and Technologies) and DARPA (US Defence Advanced Research Project Agency), to gather the state of the art of IR technologies and systems, through a series of area of focus (called *tracks*) which consist on competitions among different concurrent systems in performing a specific IR task Given a fixed test collection (consisting on textual documents) and given particular measure of performance (among which Precision and Recall), different systems are asked to perform a specific IR task, like finding relevant document in a specific domain or answer to user's natural language questions (*Question Answering*).

The analysis of the tracks and the performance results of the participating systems are thus a good indicator of the state of the art of the IR technologies and the future direction of research in the retrieval tasks.

2.1.1.7 Interesting Readings

- *Ed Greengrass*, **Information Retrieval: A Survey**, 2000. (on-line: http://www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf)

- *Keith van Rijsbergen*, **Information retrieval**. 1979. (on line: http://www.dcs.gla.ac.uk/Keith/Preface.html)

- *Gerard Salton and M J McGill*, **Introduction to Modern Information Retrieval**, McGraw-Hill Company, 1983.

- *R. Baeza-yates and B. Ribeiro-Neto*., **Modern Information Retrieval**, Addison-Wesley and ACM Press, 1999.

- *Karen Sparck Jones and Peter Willett (eds)*, **Readings in Information Retrieval**, Morgan Kaufmann, 1997.

2.1.2 Information Extraction

2.1.2.1 Basic Concepts

Information Extraction (IE) is the activity of analysing unstructured natural language texts (the *corpus*) in order to extract information about pre-specified types of events, entities or relationship [Pazienza, 1997] , [Pazienza, 1998] , [Pazienza, 2003].

Taking as input not previously indexed and possibly ambiguous text, IE techniques aim thus to produce an unambiguous an fixed-format output, corresponding to a pre-specified user information need. The information need is usually expressed through a particular structure called *template*, that is, a sort of information skeleton that must be filled by the IE system.

For example let assume a financial domain: a particular information need could be the extraction of all the company acquisitions described in a corpus. Such need can be expressed into the following template:

COMPANY ACQUISITION TEMPLATE	
Buyer	*[company]*
Acquired	*[company]*
Date	*[date]*
Price	*[money]*

The task of an IE system would be then to fill the template, that is, find all the company acquisitions, defining the company that was acquired, the buyer, the date and the price of the acquisition. The extraction of these kind of specific information is not an easy job:

- the extraction of precise data from unstructured and vague texts (as those in Natural Language) intrinsically needs a deep syntactic and semantic analysis of the corpus;
- the possible answers to the information need can take different lexical forms in the text.

Referring to previous IE template, sentences like “Microsoft bought IBM for 300M$” and “IBM was acquired by the Bill Gates company” drive the same semantic meaning, and should be thus both be extracted by the IE system to fill in the template with retrieved information..

2.1.2.2 IE and IR

As previously mentioned, IE and IR can be seen as different disciplines, if we intend IR in its more specific definition, that is, as Document Retrieval. The first difference lies in the output produced by IR and IE systems.

The aim of IR is to retrieve relevant documents from a text collection; the aim of IE is to extract a specific relevant information from documents. From this point of view the two disciplines can be seen as complementary: a general information extraction system could be intended as a cascade of both an IR subsystem and a IE subsystem, where the first one retrieves relevant documents that are then analysed by the IE system to extract the template need (see Fig. 1).

A second difference lies in the techniques used by the two disciplines. In order to identify relevant documents IR relies mainly on statistical and probabilistic analysis applied to documents and the whole collection: documents are seen simply as *bag of words*, that is, an unordered set of terms whose semantic meaning can be left aside. IE, on the other hand, needs a deep and precise understanding of the text in analysis: it thus relies on rich syntactic and semantic techniques (parsing, Named Entity recognition, etc.) able to detect specific information in the corpus and to understand the meaning and the syntactic-semantic role of the terms.

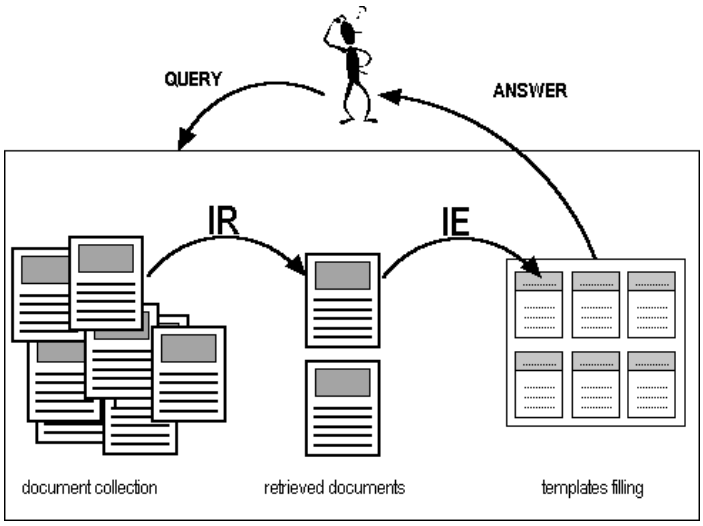


Fig. 1 A schema of IR-IE integration.

A third difference relates the interpretation of the user query. IR retrieve the documents that contain the exact terms indicated in the user query, permitting at most lexical variances of the words (stemming) or superficial semantic variation (as synonymy). IE looks in the corpus not only for expression lexically corresponding to the user query, but also for expression that have different lexical form, while containing the same meaning (as in the abovementioned example).

Thus, IE systems seem to be much more accurate and useful for the retrieving of a specific information need: the user doesn't have to analyse a whole set of retrieved documents looking for the needed information, as happens in IR, but simply he has to read the answer returned by the system. The problem with IE system is that unfortunately they are far from achieving the same performance as the IR systems, due to the inherent complexity of their task, that limits Precision and Recall, and due to the computational costs of their analysis, which makes the systems much slower than the IR. Moreover, while IR systems usually apply to open domain, most of the IE tools are still tuned for answering in specific domain (such as finance, medicine, news…) due to the use of certain semantic resources that are strongly domain dependent.

2.1.2.3 Types of IE

Starting from 1987 a cycle on annual conferences called MUC (Message Understanding Conference) and devoted to IE has been organized by the DARPA and the US Navy. Thanks to these conference IE has assumed a growing interest in the scientific community, and has allowed a sort of standardization in the IE framework. Today, as underlined in the MUC conferences, there are a few types of IE, called *tasks.* Each task refers to the extraction of a particular kind of information from a domain specific text corpus: actors, events and relations.

Named Entity Recognition (NE) Task

The aim of the NE task is to identify in the corpus all the names of people, places, organizations, dates and amount of money. All those entities are called *Named Entities*, and usually represent important concepts for the specific domain in analysis: the NE task is thus weakly domain dependent. NE is one of the oldest, most simple and most reliable task of IE: today it achieves performance that can be compared to human identification, reaching an accuracy in NE classification over 96%.

The following example show the tagged output text of a NE system:

```
<enamex type="organization"> Bridgestone Sport Co.
</enamex> said <timex type="date"> Friday </timex> it
has set up a joint venture in <enamex type="location">
```

Taiwan </enamex> with a Japanese trading house to produce golf clubs to be shipped to <pnamex> **Japan** </pnamex>.

Coreference Resolution (CO) Task

The aim of the CO task is to identify in the documents identity relationship between entities in texts: entities can be both NE and anaphoric references to NE. CO creates a sort of link between entities and their anaphoric reference in the documents. This task is useful in the definition of other task, such the following TE and ST. Today anaphora resolution system achieve performances slightly above 70%, far from the average human score, that is around 80%.

Template Element Production (TE) Task

TE uses both NE and CO to identify important entities in the corpus and to associate to each of them important descriptive information, thus creating a first simple form of template, called *template element.* Example of template elements will be shown later. As NE, also TE is weakly domain dependent. These kind of system achieve performance around 80%, while human reach 93%.

Scenario Template Production (ST) Task

The aim of ST is to extract relations among template elements, creating complex templates. These templates represent *events* and situation inherent to the domain scenario that involve important entities. The previous example of "company acquisition" is a simple form of scenario template production. ST is one of the most complex IE task and at the same time one of the most interesting and useful: it is in fact through a scenario template that a user express his information needs, and it is the task of the IE system to fill this template with the correct information. At present, ST system have performance around 60%, but the difficulty of the task is testified by the low human performance, around 80%.

In the following example (from [Gaizauskas and Wilks, 1998]) the TE and ST filling task in a financial domain are described.

a) DOCUMENT:

<DOC>

<DOCNO> 56 </DOCNO>

<HL> Who's news: Burns Fry Ltd. </HL>

<DD> 04/13/94 </DD>

<TXT>

BURNS FRY Ltd. (Toronto) -- Donald Wright, 46 years old, was named executive vice president and director of fixed incombe at this brokerage firm. Mr. Wright resigned as president of Merrill Lynch Canada Inc., a unit of Merrill Lynch& Co., to succeed Mark Kassirer, 48, who left Burns Fry last month. A Merrill Lynch spokeswoman said it hasn't named a successor to Mr. Wright, who is expected to begin his new position by the end of the month.

</TXT>

</DOC>

b) TEMPLATES ST:

<TEMPLATE> := *DOC_NR:* " 56 "

CONTENT: <SUCCESSION EVENT 56-1>

<SUCCESSION EVENT 56-1>:=

SUCCESSION_ORG: <ORGANIZATION 56-1>

POST: "executive vice president "

IN_AND_OUT: <IN_AND_OUT 56-1>

<IN_AND_OUT 56-1>

VACANCY REASON: OTH_UNK

<IN_AND_OUT 56-1>:=

IO_PERSON: <PERSON 56-2>

NEW STATUS: OUT

ON_THE_JOB: NO

<IN_AND_OUT 56-1>:=

IO_PERSON: <PERSON 56-1>

NEW STATUS: IN

ON_THE_JOB: NO

OTHER ORG: <ORGANIZATION 56-2>

REL_OTHER_ORG: OUTSIDE_ORG

c) TEMPLATES TE:

<ORGANIZATION 56-1>:=

ORG_NAME: "Burns Fry Ltd."

ORG_ALIAS: "Burns Fry"

ORG_DESCRIPTION: "this brokerage firm"

ORG_TYPE: COMPANY

ORG_LOCALE: Toronto CITY

ORG_COUNTRY: Canada

<ORGANIZATION 56-2>:=

	ORG_NAME:	"Merrill Lynch Canada Inc."
	ORG_ALIAS:	"Merrill Lynch"
	ORG_DESCRIPTION:	"a unit of Merrill Lync & Co."
	ORG_TYPE:	COMPANY
<PERSON 56-1>:=		
	PER_NAME:	"Donald Wright"
	PER_ALIAS:	"Wright"
	PER_TITLE:	"Mr."
<PERSON 56-2>:=		
	PER_NAME:	"Mark Kassirer"

2.1.2.4 A generic IE system

Every IE system has its own extraction strategies, text analysis and implementation techniques. Nevertheless., all IE system refer to a common abstract model, today reckoned as a standard for the development of IE architectures. This model, proposed in 1993 by J.R.Hobbes [Hobbes, 1993], has been called *generic IE system,* and it is defined as *"a cascade of modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically"*.

The ten modules identified by Hobbes are the following:

1) Text Zoner

The Text Zoner transforms the input text in segments, that can be paragraphs or macro-structures.

2) Preprocessor

This module transforms text segments into a sequence of sentences, identifying the sentence boundaries. Sentences are further divided into lexical items, that is, single words and punctuation elements. At each lexical items is then associated its grammatical role, called Part of Speech or *POS*. Collection of lexical POS tags are used to this aim, comprehending roles like *proper noun* (*NN tag*), *adjective* (*JJ tag*), *cardinal number* (*CD tag*) and so on. Furthermore, the Preprocessor recognizes and normalizes basic entities to their corresponding NE, like dates, persons and companies.

3) Filter

The aim of the Filter is to discard irrelevant sentences from the original text, making it shorter.

4) Preparser

This module groups, when needed, words into small structure of lexical elements that convey a unique idea or that are of common use. Systems thus can identify noun phrases and verb groups and attach appositives to their head nouns (like genitives and "of" prepositional phrases).

5) Parser

The Parser analyses each input sentence (represented by a sequence of lexical items and phrases called fragments) received from the preparser and tries to produces a parse tree, assigning to each item its syntactic role. At present, most of the parser for IE have abandoned full-sentence parser for a shallower form of parsing, whose aim is only to identify fragments.

6) Fragment Combiner

This module takes as input the parse trees produced by the parser: since usually parsers can not produce full parses for an entire sentence, its task is to combine the parse trees of each sentence.

7) Semantic Interpretation

With this module the semantic analysis of the input text begins. The Interpreter translate the parse tree fragments into semantic structures or logical forms or event frames. For example for each phrases can be produced a predicate that represent its syntactic role (e.g. "Prime Minister" is translated in person(Prime Minister)). Also lexical disambiguation usually takes place at this level.

8) Lexical Disambiguation

If not performed in the previous modules, lexical disambiguation is carried out, translating general or ambiguous predicates in new unambiguous predicates, through the use of manually produced rules or statistical methods.

9) Co-Reference resolution / Discourse Processing

This module identifies anaphoric and co-reference elements (pronouns, noun phrases and also events) linking different descriptions of the same entity present inside the text. Moreover, the Discourse Processor organizes the semantic structures produced by the Semantic Interpretation module into a unique knowledge base hierarchically organized (discourse model).

10) Template Generation

The semantic structures produced by the previous modules are finally matched with the templates manually created to satisfy the user need.

2.1.2.5 The Scenario

One of the major limitations of most IE systems is their strong dependence on the domain and on the type of information they are requested to extract. Indeed, in order to achieve good performances, an IE architecture must have not only a deep knowledge of the domain but also of type of events that the final user wants to be extracted (templates). This whole framework is called *scenario*. The strong dependence on the scenario is due to the techniques used by the IE system in specific modules (*parsing, semantic interpretation* and *discourse processing*). In order to produced useful information to fill the user templates, these modules must in facts rely on resources strictly tied to the template itself. In particular the three modules are responsible of identifying and extracting:

- *Entities* that take part to an event and their typology (Named Entity Recognition, NE)
- *Events* related to the specific scenario (Scenario Template Production, ST)

NE is usually carried out by the parser, while ST is performed by the Semantic Interpreter; finally, in the Discourse Processing phase entities and events extracted are ontologically organized, permitting an identification of the correct answer to the user template requests.

Recently a few IE system ([Riloff, 1996][Yangarber et al., 2000][Basili et al., 2002][Zanzotto, 2002]) have proposed a *scenario independent* approach to entity and event extraction. The assumption of these architectures is that the information extraction can be carried out not only in view of a specific user need, but can be driven by the corpus itself. In other terms, the events extracted from the corpus are those that seem to be *interesting*, where the *interesting degree* is computed looking at statistical and semantic information related to the corpus itself. The extraction is thus not driven from the outside (a specific user information need) but from the inside of the corpus: the scenario is then expressed by the corpus itself. The resulting IE architecture is thus more generic and domain independent.

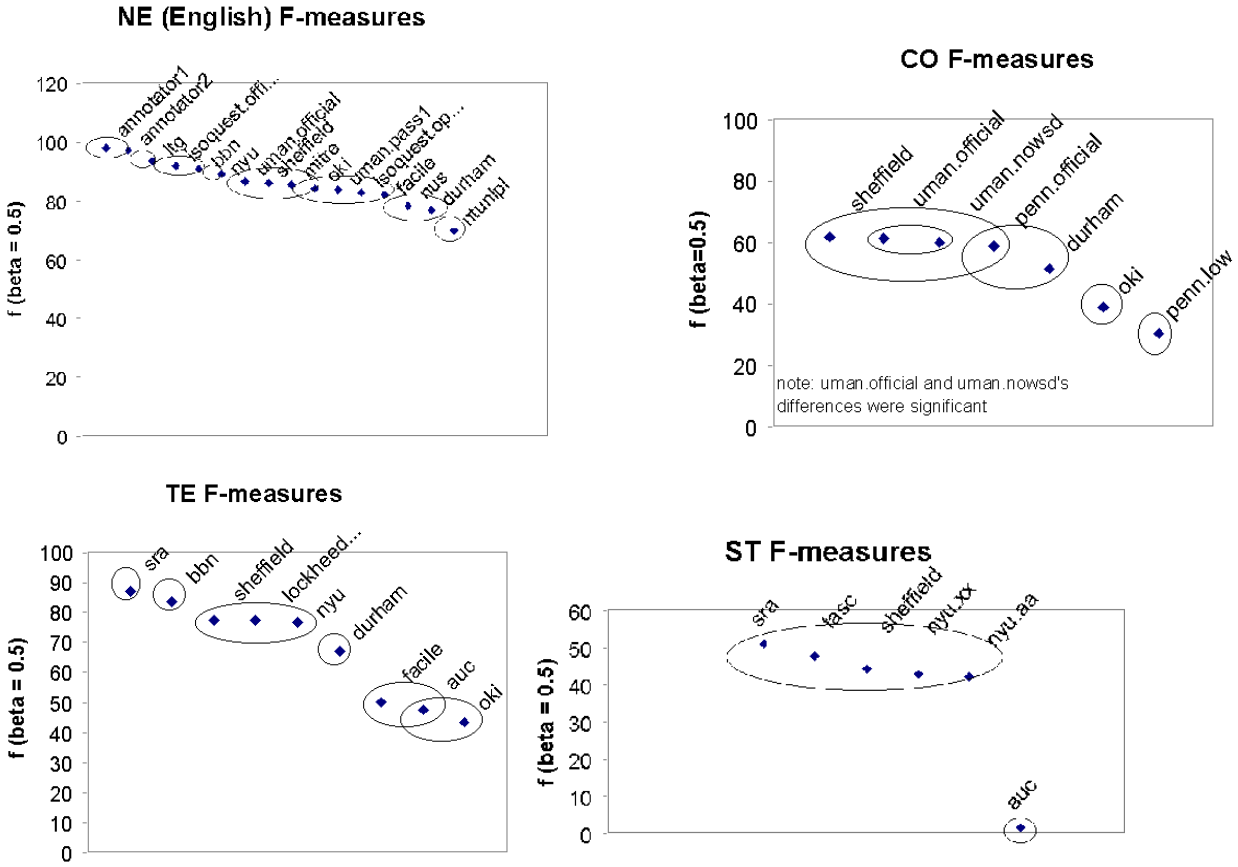
2.1.2.6 IE applications and integration

As demonstrated by the variety of systems presented at the MUC conferences [MUC, 1997], IE techniques can be applied to a wide range of specific and practical tasks. For example, IE systems are used in *news* applications, to extract or classify, from press agency news, relevant information of specific interest. In *finance* IE techniques are already used by a few companies to identify particular events like company acquisitions or joint ventures, analysing automatically huge collection of financial magazines and newspaper. *Medical* applications extract and gather information related to a patient, looking at medical summaries and reports, proposing, based on these information, a rough classification of the patient for medical auditing. Other applications have been developed in military intelligence, police, law and employment scenarios.

It is also interesting to analyse the integration of IE with other technologies. For instance, in *Natural Language Generation* the ontological representation of a document obtained by an IE system, could be used to produce a new shorter version of the document itself (a sort of abstract): a final user could thus apprehend the same information reading just a synthesis of the original document (*Text Summarization*). Being the ontological representation of a document language independent, it can also be used to produce a new document in a different language (*Machine Translation*): translation is thus performed through a meta-linguistic and semantic step (represented by the knowledge base) avoiding all the problems related to the *word-by-word* translation.

2.1.2.7 Past, present and future

Interest on IE started in the 1960s, with the implementation of the first applications devoted to template filling. But the real grow of the field begun in late 1980s, when DARPA funded competing research groups to pursue information extraction. Starting from1987 the annual Message Understanding Conference (MUC), sponsored by the US Navy and DARPA, have been the key events in driving the field of IE forward. The aim of the conferences is to gather, evaluate and compare the state of the art IE systems, using a common test bed of information extraction. Each year the MUC conference proposes a different domain over which the systems must measure their performance on the different IE tasks (NE, CO, TE, ST).Performance results (F-measure) on some tasks of the MUC-7 obtained by the participant systems are summarized in the following tables (from NIST website).



Besides MUC, other IE projects have been developed, most of which under the Linguistic Research and Engineer (LRE) initiative, funded by the European Commission: the LRE initiative is trying to push IE over the scenario perspective, encouraging the development of domain independent architecture easily reusable in different domains.

Reusability and domain independence is one of the key concepts that will be focused by the IE community in the future: in facts, porting IE system to new domains is still a serious bottleneck for state of the art systems. The development of user-centred adaptive systems is thus one of the major challenge for the future, that will show if IE technology will be widely applicable in real applications.

2.1.2.8 Interesting Readings

- *H. Cunningham,* **Information Extraction - a user guide**, Research Memo CS-97-02, University of Sheffield, Sheffield, 1997. (on line: http://www.dcs.shef.ac.uk/~hamish/IE)

- *R. Gaizauskas and Y. Wilks,* **Information Extraction: Beyond Document Retrieval**, In *Journal of Documentation, 54(1):70--105,* 1998.

- *M.T Pazienza (Ed.),* **Information Extraction in the Web Era**, *LNAI 2007,* Springer, 2002.

2.1.3 Question Answering

2.1.3.1 Basic Concepts

Question Answering (Q/A) can be intended as the process of automatically extracting specific answers to a question in Natural Language, using a collection of documents.

Thus, in Q/A the user information need is expressed in the most natural way, through a simple question that carries all the semantic need required. Such a straightforward and realistic approach from the user's point of view, implies, on the other hand, a huge effort in the process of question interpretation and answer retrieval.

A number of *dimensions* must be taken into consideration in order to define a Q/A system. The first one relates the *nature of the information* the system uses: knowledge can in fact codified either in a structured model (such a database) or in an unstructured form (e.g. free text); answer retrieval strategies strongly depend on the specific kind of adopted knowledge representation model.

Another important aspect of Q/A is the *nature of question*: a question can in fact request a specific fact or an informed opinion about a fact. As we will describe later, questions can range from simple *factoid* questions (that is questions completely based on facts *explicitly* represented in the corpus) to *speculative* questions, which imply reasoning and deductive mechanisms.

Nature of the answer is also a crucial issue: it can in fact consist in a mere extraction of a textual passage from a corpus document, or it can be a linguistic well-formed answer resulting from the semantic analysis of the whole corpus. In the latter case semantic and syntactic technique need to be implemented in order to extract from one or more documents all the information useful for the answer definition, and to produce a correct answer that gathers these information.

All the abovementioned aspects drive the choice of the techniques implemented in the system: complex type of information answer and question usually need sophisticated and knowledge intensive techniques, like syntactic parsing and semantic analysis. Shallower approach (i.e. *n*-grams) are instead required for simpler systems.

One of the major challenge in Q/A is that question and answer can have different surface structures, that is, the meaning is lexically represented in different ways. For instance the question *"When did Microsoft buy IBM?"* could be answered with a document sentence like *"IBM was acquired by Microsoft in 2005"* . This sentence, that semantically drives the correct information, has a lexical structure different from the question. This so called *lexical gap*, common also to IE (as already seen in the previous section) and other NLP applications , can be solved using appropriate lexical and semantic resources, using shallow reasoning and paraphrasing techniques.

A broad taxonomy of Q/A system can be thus outlined [Moldovan and Surdeanu, 2002] taking into consideration the previous points.

Class 1. *Q/A systems that process factual questions*

Using empirical methods based on keyword manipulation, these systems extract the answer to the user's questions as text literally extracted from one or more documents, and recombined together. An example of factual Q/A could be: *"Who is the President of the United States? J.W.Bush"*

Class 2. *Q/A systems with simple reasoning mechanism*

These systems make use of inferential reasoning in order to relate answer and question, through the use of ontologies, pragmatic knowledge and semantic resources (e.g. Wordnet). An example could be: *"How did Socrates died? Drinking poison"*

Class 3. *Q/A systems capable of answer fusion*

These systems are able to elaborate an answer putting together pieces of information scattered in more than one document: it can be simply requested to assemble a list of information, or to produce a sort of script. For example a typical question is: "*How can I assemble a bicycle?*"

Class 4. *Q/A systems capable of interaction*

In order to answer to a question the system takes into consideration the user interaction context, that is, it makes reference to information inside previous questions asked by the user and the corresponding answer.

Class 5. *Q/A with speculative questions*

These advanced systems should be capable of answering a question with an information that is not reported explicitly in the corpus documents. The answer is, in other terms, derived from reasoning applied to documents. Usually, a question is transformed into a set of queries that retrieve evidence regarding the question; these evidence are then gathered in the answer, through techniques based on reasoning by analogy. Most systems make use of ontology and case-based reasoning. An example question could be: *"Is the airline in trouble?"*

Apart class 1, all other Q/A systems require a huge amount of both specific knowledge and reasoning. As a consequence only class 1 systems are currently under implementation in several application contexts.

2.1.3.2 A generic Q/A architecture

Most Q/A systems broadly consist on three macro-modules: *query formulation*, *document retrieval* and *answer extraction/formulation*. Analysing different architectures, it is possible to further divide the three main modules in 10 smaller modules [Moldovan and Surdeanu, 2002], each one of those is responsible for a specific operation of the whole Q/A process.

Module 1. *Question keyword pre-processing [query formulation]*

All the important words of the user question are spell-checked and expanded with their lexical variants. For instance from the question :*"Who was the President of the United States in 1996 ?"* the lexical variants could be:, *The_President, USA, 96*.

Module 2. *Internal representation construction [query formulation]*

User question is parsed, stop words (like articles and prepositions) are removed, important concepts are identified and relations among concepts are retrieved. In the previous example important concepts are: *President, United States, 1996*.

Module 3. *Answer type identification [query formulation]*

Based on semantic and syntactic information and resources (e.g. Wordnet), the *answer type* of the question is identified. This type is then used by the following modules to retrieve the correct kind of answer. For instance, the *who* question in the example requests an answer type *person*. Classical type of answers are: *person, location, organization, quantity, date, who, where, which, what,* and *when*. Answer type classification (that is, given a question find its class among *k* classes) is an hard task, that can be performed both manually (using handcrafted rules) or automatically (using machine learning algorithms). In order to find the correct kind of answer to a specific answer type in the corpus documents, some systems apply to the corpus a pre-processing step of *predictive annotation*. This techniques consists in indexing a document with

concepts that are expected to be useful for searching the answer: a document is thus annotated using specific tags that identify concepts like *person* and *location* , that can be matched with the answer type.

Module 4. *Keyword selection [query formulation]*

A subset of the question concepts identified by Module 2 are gathered together to form the Boolean query that will be used by the document retrieval modules. The selection of important concepts is usually carried out using part of speech information. Thus, the query in the example will be: "*President* AND *The_President* AND *United States* AND *USA* AND *1996* AND *96*"

Module 5. *Keyword selection [query formulation]*

Keywords selected by the previous modules are expanded with lexical, syntactic and semantic alternations, that are then integrated in the Boolean query.

Module 6. *Retrieval [document retrieval]*

The query produced by Module 5 is processed by the retrieval engine, which return all the documents that contain its keywords. IR techniques are generally used to retrieve documents, but a few peculiarities of Q/A retrieval must be taken into consideration to adapt such techniques in the Q/A framework. First of all, in Q/A the information need is more specific than in ad hoc retrieval; secondly there are less relevant document to be retrieved, and the answer to the question is expressed very locally inside the document, roughly one or two sentences. That is why after document retrieval, documents are restricted to text passages where all the keywords are located. For example a retrieved passage could be: "*Bill Clinton was the President of the USA in 1996, when he decided to …*". This specific Q/A-IR techniques has been called *passage-based* retrieval, and is used in several Q/A systems. Other common IR techniques, like stemming and lemmatization, are also used to improve performance.

Module 7. *Passage filtering[document retrieval]*

Among all the retrieved passages, only those that satisfy the question semantic constraint are retained. In the example *1996* could be a semantic constraint on the date of answer.

Module 8. *Identification of candidate answers [answer extraction]*

Another passage filtering process is carried out: only those passages that contain the answer type requested by the question (Module 3) are selected. The answer type can be identified, for instance, through the use of Named Entity Recognizers. This technique works fine, but a problem arises when the passage that expresses the correct answer to the user's question doesn't contain the answer type requested: this is the case, for example, of co-reference expressions. For instance an answer to the previous example question could be contained in the passage "*He was President in 1996*", where the pronoun *he* is a co-reference to the correct answer type *Bill Clinton*, maybe expressed in a previous passage. As the example shows, passage that contain co-referring expression may not contain the terms the expression co-refer with. A solution to this problem is to improve retrieval of relevant passages by adding terms based on detectable co-reference and co-description relations. The use of co-reference resolution has shown to be useful, as many systems using this techniques improve Q/A performance results.

Module 9. *Answer ranking [answer extraction]*

Answers are ranked according to their relevance. Relevance can be computed in different ways (for example the distance in the passages between the keywords can be considered a relevant measure).

Module 10. *Answer formulation [answer extraction]*

Answers with highest relevance are selected and shown to the user as a literal fragment of the text extracted from the corpus or in a internally generated form.

2.1.3.3 Q/A, IR and IE

Many of the technologies used in IR and IE can be also applied to Q/A. In the following table a list of the main modules used by the three technologies is presented, underlying the common characteristic among them. In the table, features that are fully supported by the system are marked with *x*, features that are only partially supported with *y*.

Subsystem	Module	IR	IE	Q/A
Question processing	Keyword processing	X		X
	Question representation			X
	Answer prediction			X
	Keyword selection	X		X
	Keyword expansion	X		X
Document indexing and retrieval	Document indexing	X	Y	X
	Document search and retrieval	X	Y	X
	Document ranking	X	Y	X
Document processing	Morphological and lexical proc		X	X
	Extract relevant passages		X	X
	Syntactic parsing		X	X
	Name entity recognition		X	X
	Coreference		X	Y
	Discourse processing		X	X
	Semantic analysis			X
Use of world knowledge	WordNet		X	X
	Dictionaries		X	X
Use of domain knowledge	Domain ontologies		X	X
	Domain patterns		X	
	Domain coreference		X	
	Domain event merging		X	
Output extracting	Patterns		X	X
	Complex nlp techniques		X	X
	Merge		X	X
	Answer ranking			X

	Logic prover			X
	Answer justification			X
Output formatting	Template filling		X	
	Answer formulation		Y	X

In a Q/A system the role of the retrieval component and the role of the information extraction component can be more or less emphasized. There is then a common distinction between *IR-based* and *IE-based* systems. The former are built on top of an IR-engine, and thus strongly rely on statistical analysis of the documents; the former make use of more semantic methodologies and resources.

2.1.3.4 Past, present and future

While Q/A has received great attention in recent years, early works started to appear in the 1960's. As in modern systems, two distinct approaches to Q/A have been studied in the past: a database-oriented approach, and a text-based approach. Early example of Q/A database-oriented systems are *Baseball* [Green et al., 1963], used to answer questions about scores, teams and locations of baseball games, and *Lunar* [Woods, 1977], a complex system to access chemical data on lunar material (moon rocks and soil) compiled during the Apollo missions. Relying on a English parser, Lunar was able to translate Natural Language user's question into queries for the database , using syntactic and semantic rules. Research on database-oriented systems continued up to the 1980's, when the interest shifted to text-oriented approaches using IR and IE techniques.

Today the TREC annual conferences are the most important place to look at the state of the art of Q/A technologies. In fact, one of the TREC tracks specifically refers to Q/A (*Question Answering Track*): system are tested over a pool of questions of various type and a common corpus. In TREC-2003 [Voorhees, 2003b] 500 questions have been proposed, 410 of which were *factoid* questions, 50 *definition* questions and 40 *list* questions. Definition questions require the systems to retrieve a definition responding to queries like *"Who is Colin Powell?"* or *"What is mold?"*. List questions (like *"List the names of chewing gum*") require systems to assemble an answer from information located in multiple documents. More complex kind of questions (Class 5 previously defined), that imply a strong reasoning component inside the system, have not been taken into consideration by the TREC Q/A task up to 2003, since the development of such kind of systems is still a task to be explored.

Another important conference cycle is the European *CLEF* (Cross Language Evaluation Forum) [Peters et al., 2002], which has recently focused on an important open issue: *multilingual-Q/A*. In particular, CLEF tests systems over three different languages (Italian, Spanish and Dutch).

Other Q/A open issue regard the developing of domain specific systems (biomedicine, law, etc.) and the integration between the database-oriented approach and the open-domain Q/A, since a growing number of Internet site have useful information in some kind of structured format.

2.1.3.5 -Interesting Readings

- *L. Hirshman,R. Gaizauskas*, **Natural Language Question Answering: The view from here,** Natural Language Engineering 7(4), 2001.

- *Dan Moldovan, Mihai Surdeanu*, **On the Role of Information Retrieval and Information Extraction in Question Answering Systems**, In *Information Extraction in the Web Era*, M.T.Pazienza (ed), Springer, 2003, pag 129-147.

2.1.4 A global view of Information Systems approaches

In order to have a global and synthetic view of Information System approaches, it can be useful to define a two-dimension space, where one dimension represents the degree of specification of the information need and the other the type of information sources used by the specific Information Systems (i.e. the degree of structure of sources).

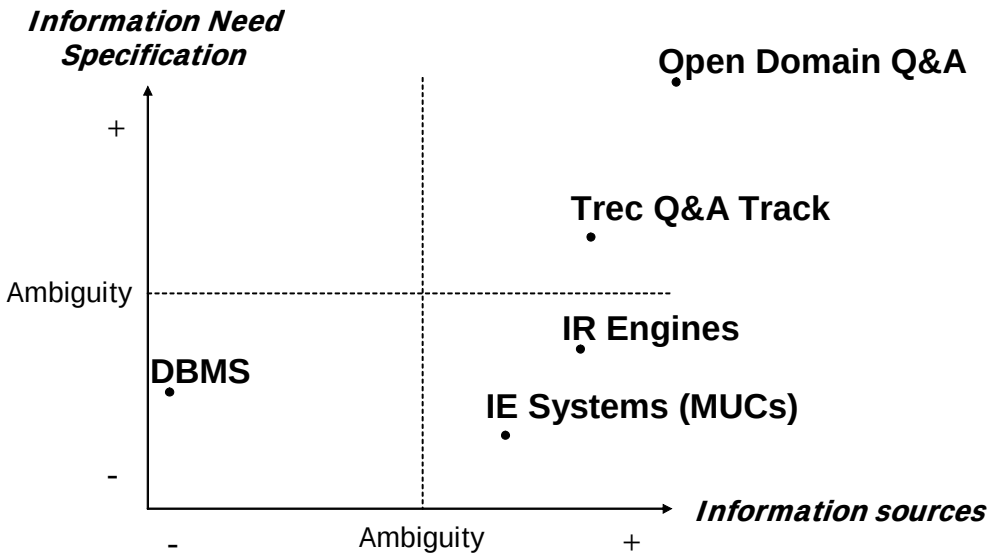


Fig. 2 Information System space

As the graphic in Fig. 2 shows, Database Management Systems (DBMS) rely on structured type of information sources (databases) and on fixed formalisms for conveying the information need. Thanks to this structured approach, DBMS don't have to face the problem of dealing with ambiguous expressions, which normally arise in Natural and loosely controlled Languages. On the other side, DBMS can only deal with the restricted set of structured sources; moreover the user must express his need in a specific language or grammar, whose learning process can not be straightforward. Because of that, more adaptable approaches are required for those applications whose aim is to deal with Natural Language and with the largest set of sources of information available. Relaxing the *no-ambiguity* constraint, IE, IR and Q/A systems are able to cope with unstructured or loosely structured type of information sources. Q/A systems, in particular, abandon a structured approach also during the specification of the information need: in facts, while IE and IR approaches still need queries expressed in a formal language, Q/A can deal with natural questions written by the user. A further distinction can be outlined between TREC-Q/A (that is, TREC competition-based system), where the information need is restricted only to a specific domain, and Open Domain Q/A, where the user can ask any kind of questions. The development and implementation of Open Domain Q/A is one of the key tasks in present and future NLP researches.

3. Automatic Assistant for Information Access: RTV proposals

The kind of information needed during the mission design is largely expressed within documents by means of natural language. Tools for retrieving and coherently organising documents are then necessary and complementary resources for a design environment such as the ESA Concurrent Design Facility[1] (CDF).

Each design process is a task asking for the cooperation of a team of experts with both different backgrounds and perspectives. In designing space missions, the goal of the process is to produce a spacecraft able to accomplish the envisioned mission. During this process a large quantity of knowledge is accessed and as a result a knowledge repository is produced: one of the preferred means to express this knowledge is natural language.

The access and the maintenance of such a large amount of textual material is then a problem with which we have to deal: a few solutions can be provided to facilitate the access the to the two following different knowledge repositories:

- pre-existing knowledge that may be expressed in internal documental databases or external knowledge sources such as the Web, etc.;
- knowledge "formalised" during the project of the space mission, i.e. documents, reports, and minutes produced during the design of the mission and of the actual spacecraft.

As described in Sec.2.1, methods for searching and organising documents have been largely investigated in Information Retrieval (IR) and in Natural Language Processing (NLP): thus, successful applications are nowadays widely available. The definition of a coherent system able to handle such amount of textual material is then possible.

An "ideal" assistant helping people in finding information they need should be able to "understand" users' information needs and to search the knowledge repositories for documents able to satisfy these specific needs. As an additional feature, it can offer the possibility of understanding the common "jargon" and terminology used in the design process. It is not possible to refer to a predefined, shared and agreed knowledge: in fact we have a team (that is a new expert community created for a specific task) composed by persons with different backgrounds willing to adhere to a new common operative scenario. It is plausible that some new concepts arise during the design process and assume a status of shared concept. This should be recognised and used to define the common understanding and to offer a means to retrieve documents whenever necessary.

A complete solution for building an information access assistant is widely depicted in Fig. 3, where several functional architectures are depicted. Details on of the related architecture as well as on methodological approaches are in both Sec.3 and Sec.4 .What we suggest for the complete architecture is a *proactive* system, that "listens" at the dialogues going on among the project participants (through the minutes of the meetings, for example) and extracts information needs. Later on, these are used to query information access systems able to retrieve documents where the information needs can be satisfied.

Once selected as relevant by final users, retrieved documents contribute to the definition of the knowledge relevant for the design project. These documents can be then used to extract an explicit conceptualisation of the mission. This explicit conceptualisation can be called *"mission ontology"* (see Fig. 3), that is the formal representation of the knowledge as it is embedded into the documents and present in expert background.

The mission ontology has a twofold scope. On the one side, it could somehow represent how the project contributed to the systematic representation of the knowledge about the space missions. On the other side, it could be a means to make a useful indexing of the documentations produced and gathered during the mission design.

The overall system could result in facilitating:

- the access to the project related documentation;

[1] http://www.esa.int/SPECIALS/CDF/index.html

- the access to external information;
- fixing the terminology and the knowledge built in the process (the ontology of the mission);
- the creation of a central view on the knowledge stored on the project related documentation using the proposed terminology.

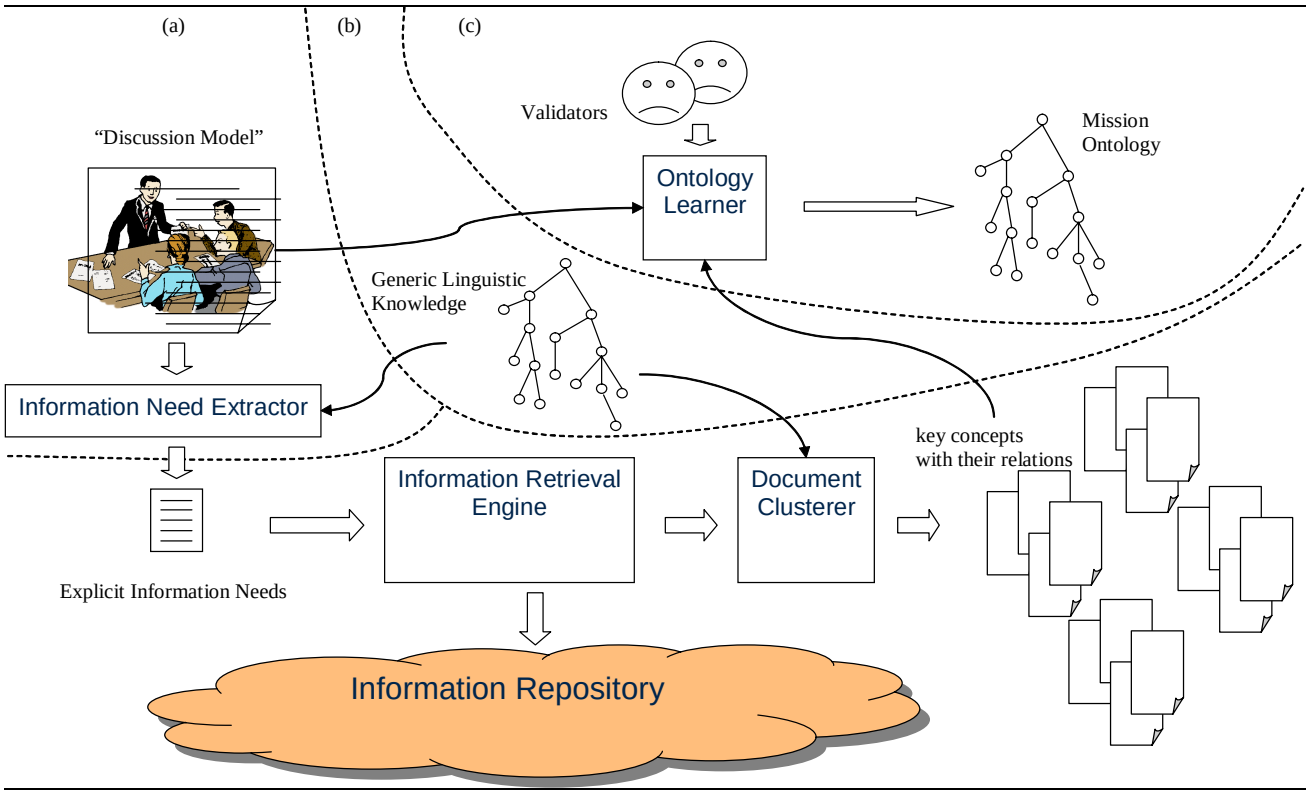


Fig. 3 A complete solution for an Automatic Assistant

Such a system can be realised with technologies ranging from Information Retrieval engines (based on the bag-of-word document model to ensure the coverage of the phenomena), to knowledge based systems using also complex natural language models. Either generic linguistic (such as WordNet [Miller, 1995]) or space mission semantic knowledge can be used to empower document clustering and to interpret ambiguous and unknown terms. General linguistic expressions are the common ground when domain specific communication fails.

Thanks to its modular structure, the overall described architecture is not required to be implemented only in the complete and very rich set of functionalities. It is possible to arrange several different architectures where more functionalities can be added passing by one to others. Starting with the IR engine as the "core" system, further capabilities can be added.

1. The "core" of the proposed system could then be the Information Retrieval engine plus the Document Clusterer. This is represented in the bottom of Fig. 3. The "core" system will answer to explicit information needs that are used to retrieve documents in the information repository, that will be then provided in topically coherent clusters (see Sec.3.1).

The additional capabilities that may be foreseen (as depicted in Fig. 3 under (a), (b), (c)), define different systems:

2. the system behaves an "active" information access system. This implies that the system will be able to "follow" a conversation and extract "implicitly" stated information needs (see Sec.3.2).
3. The system becomes more robust for lexical variations by using generic linguistic knowledge bases such as WordNet [Miller, 1995] or EuroWordNet [Vossen, 1998] (see Sec.3.3).

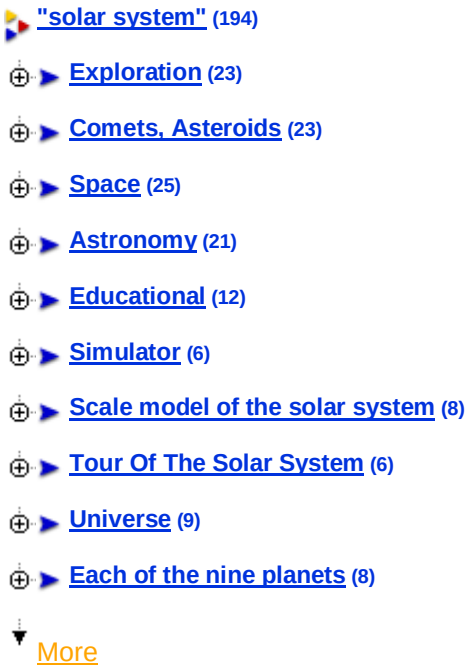
4. The system could be able to acquire an explicit model of the knowledge embodied in processed documents as well as produced by the process. This explicit knowledge model is what has been above called the "*mission ontology*". It represents the *memory* the system has about the structure of the mission (see Sec.3.4).

3.1 Information Retrieval and Clustering

The "core" of the proposed architecture, as described in the previous section, is based on both an IR engine and an automatic cluster components.

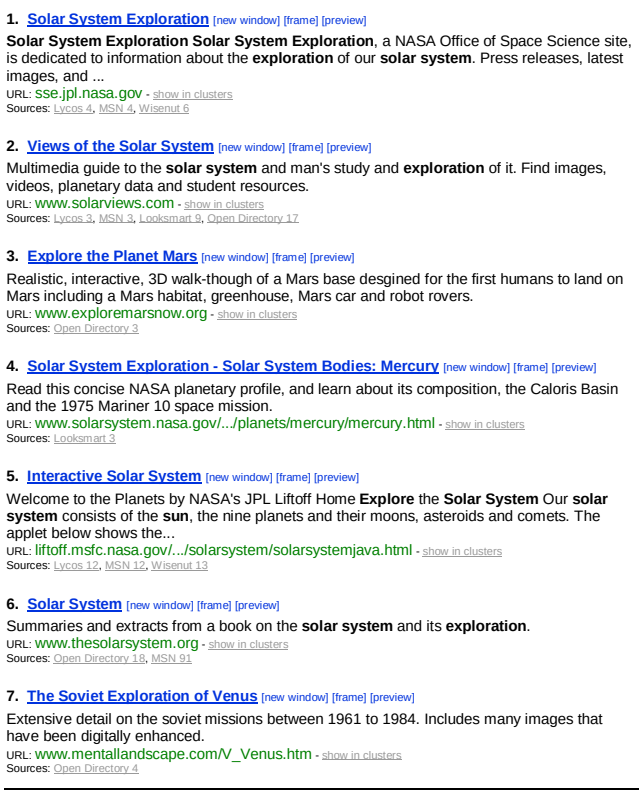
Clustering results of a given query is often seen as a way to better publish documents retrieved by an information retrieval engine. Documents are clustered according to their coherence with respect to the topic. A label, e.g. the most representative word, can be used to represent the cluster. This model seems to be very effective to present data while they have been coherently indexed: in fact, coherent documents are collected and shown. It is thus a powerful means to drive users to the relevant document.

Let us take for instance Vivisimo ©[2], an on-line system implementing the above architecture and let us query it with the keyword "solar system". Resulting documents are organised according to the following tree of labels:



showing the different clusters of documents. There is, for instance, a net separation of documents related to the *exploration* with respect to those related to an *educational* purpose. The first are related to the results of some explorative missions as it may be seen from the snippets of the firstly ranked documents (Fig. 4), while the others are pages dealing with the dissemination of the space related activities as it emerges in Fig. 5.

[2] www.vivisimo.it



1. **Solar System Exploration** [new window] [frame] [preview]
Solar System Exploration Solar System Exploration, a NASA Office of Space Science site, is dedicated to information about the **exploration** of our **solar system**. Press releases, latest images, and ...
URL: sse.jpl.nasa.gov - show in clusters
Sources: Lycos 4, MSN 4, Wisenut 6

2. **Views of the Solar System** [new window] [frame] [preview]
Multimedia guide to the **solar system** and man's study and **exploration** of it. Find images, videos, planetary data and student resources.
URL: www.solarviews.com - show in clusters
Sources: Lycos 3, MSN 3, Looksmart 9, Open Directory 17

3. **Explore the Planet Mars** [new window] [frame] [preview]
Realistic, interactive, 3D walk-though of a Mars base desgined for the first humans to land on Mars including a Mars habitat, greenhouse, Mars car and robot rovers.
URL: www.exploremarsnow.org - show in clusters
Sources: Open Directory 3

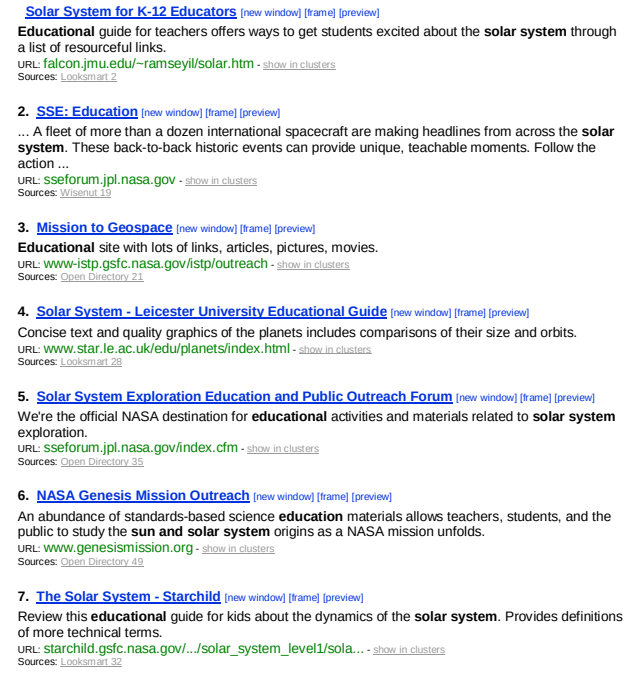
4. **Solar System Exploration - Solar System Bodies: Mercury** [new window] [frame] [preview]
Read this concise NASA planetary profile, and learn about its composition, the Caloris Basin and the 1975 Mariner 10 space mission.
URL: www.solarsystem.nasa.gov/.../planets/mercury/mercury.html - show in clusters
Sources: Looksmart 3

5. **Interactive Solar System** [new window] [frame] [preview]
Welcome to the Planets by NASA's JPL Liftoff Home **Explore** the **Solar System** Our **solar system** consists of the **sun**, the nine planets and their moons, asteroids and comets. The applet below shows the...
URL: liftoff.msfc.nasa.gov/.../solarsystem/solarsystemjava.html - show in clusters
Sources: Lycos 12, MSN 12, Wisenut 13

6. **Solar System** [new window] [frame] [preview]
Summaries and extracts from a book on the **solar system** and its **exploration**.
URL: www.thesolarsystem.org - show in clusters
Sources: Open Directory 18, MSN 91

7. **The Soviet Exploration of Venus** [new window] [frame] [preview]
Extensive detail on the soviet missions between 1961 to 1984. Includes many images that have been digitally enhanced.
URL: www.mentallandscape.com/V_Venus.htm - show in clusters
Sources: Open Directory 4

Fig. 4 *Exploration* cluster of the *solar system* query



Solar System for K-12 Educators [new window] [frame] [preview]
Educational guide for teachers offers ways to get students excited about the **solar system** through a list of resourceful links.
URL: falcon.jmu.edu/~ramseyil/solar.htm - show in clusters
Sources: Looksmart 2

2. **SSE: Education** [new window] [frame] [preview]
... A fleet of more than a dozen international spacecraft are making headlines from across the **solar system**. These back-to-back historic events can provide unique, teachable moments. Follow the action ...
URL: sseforum.jpl.nasa.gov - show in clusters
Sources: Wisenut 19

3. **Mission to Geospace** [new window] [frame] [preview]
Educational site with lots of links, articles, pictures, movies.
URL: www-istp.gsfc.nasa.gov/istp/outreach - show in clusters
Sources: Open Directory 21

4. **Solar System - Leicester University Educational Guide** [new window] [frame] [preview]
Concise text and quality graphics of the planets includes comparisons of their size and orbits.
URL: www.star.le.ac.uk/edu/planets/index.html - show in clusters
Sources: Looksmart 28

5. **Solar System Exploration Education and Public Outreach Forum** [new window] [frame] [preview]
We're the official NASA destination for **educational** activities and materials related to **solar system** exploration.
URL: sseforum.jpl.nasa.gov/index.cfm - show in clusters
Sources: Open Directory 35

6. **NASA Genesis Mission Outreach** [new window] [frame] [preview]
An abundance of standards-based science **education** materials allows teachers, students, and the public to study the **sun and solar system** origins as a NASA mission unfolds.
URL: www.genesismission.org - show in clusters
Sources: Open Directory 49

7. **The Solar System - Starchild** [new window] [frame] [preview]
Review this **educational** guide for kids about the dynamics of the **solar system**. Provides definitions of more technical terms.
URL: starchild.gsfc.nasa.gov/.../solar_system_level1/sola... - show in clusters
Sources: Looksmart 32

Fig. 5 *Educational* cluster of the *solar system* query

Clustering algorithms as well as information retrieval methods are generally based on a vector space model (see Sec.2.1.1.5) where documents are represented in the bag-of-word fashion. Nothing prohibits to use more relevant, i.e. more readable, features, such as the one used in [Moschitti and Zanzotto, 2002]. Here, features like terms and simple relations (verb-object and verb-subject pairs) are used to represent document content.

It is worth noticing that Vivisimo © is not the only commercial information retrieval engine with clustering capabilities. Among other similar tools, see RealTerm (http://www.infonetware.com/) and e-Knowledge Portal™ (http://www.knowledgestones.com). As it is a very active area in information retrieval research [Wu et al., 2003], several products have been produced as a follow-up.

3.2 "Active" Information Access System

The IR/clusterer "core" can be enhanced with additional capabilities, such the implementation of an "active" system.

The "active" system appears to be ambitious as it aims to follow the conversation as it flows in a project session and to "extract" an *implicit information need*. As depicted in Fig. 3, once explicit, information needs will be used to query an enhanced information retrieval system, as the one described in the previous section.

To be able to "follow/understand" a conversation requires the system to have deep linguistic abilities. The implementation of these abilities is neither easy nor ready on the market. Firstly, the speech recognition module has to be so robust to transcribe speech coming from the different actors of the conversation. Secondly, a computational model for conversation still remains a very complex research matter: in fact, NIST[3] challenges still not cover this kind of topics. Moreover, such a model should be able to recognise and model again the different actors of the conversation, trying to determine how the overall perspective on the problem(s) changes during the dialogue. The dialogue model is complex as people have different backgrounds (as pointed out in the description of the ESA CDF). While linguistic models have a higher possibility of being successful only if the domain where they operate is narrow and specific; conversely, a model that has to follow a conversation of a space mission design has to take into account the different backgrounds of the different conversation players.

In any case still remains the problem of defining a model of what an "implicit" information need is. That is, what can be considered as a valuable question that, if answered, can speed up the communication and the comprehension? Again, this is not a clear issue: what is useful to be shown? A criterion may be to investigate and give information on things and ideas where the communication fails, i.e. a concept that is not understood by two or more people in the same way. A model able to point out such kind of inconsistency would be a valuable means to reduce the time to reach a "clear" and useful agreement on specific topics. This results to be a very complex task as what should be investigated is the intended meaning of what is stated in order to understand if two people have a different opinion on the same term. Both cognitive and computational problems are still open

In this perspective, both tasks and related modules seem to be under defined and not clearly implemented. However, the way CDF is conceived can give the possibility of figuring out some sort of solutions. The basic idea is not to produce a complex information need extractor but a simple model taking advantages out from stable technologies.

It is a matter of fact that each dialogue player has a dedicated workstation in the CDF. This enables the use of a speech recognition module that may be trained on the specific person working at the workstation. Such trained systems are the only ones having the higher performance. Speech may then be reported with an high degree of confidence. This speech recognition systems may have their first use in the automation of the process of building the minute of the meeting: a first draft may then be corrected by the dedicated person.

The "automatically" generated or the manually compiled meeting minute can serve the purpose to feed a system able to "understand" the conversation matters and extract these kind of implicit information needs. This activity can be performed with models working at different levels. Even the bag-of-word abstraction classically underlying information extraction systems can serve the purpose of extracting terms that need to be made clearer through documents retrieved in the WWW or in the internal document repositories.

[3] The National Institute for Science and Technology of the United States of America (http://www.nist.org) organise challenges as the ones in TREC (http://www.trec.org) to push the development of what is perceived as a possible technology.

Repeated terms (i.e. linguistic representations of domain concepts and not indexing terms in the IR *sub-language*) may suggest that a disagreement exists as the underlying concept is not shared. This may be an easy way to decide a sort of list of candidates to be searched. Such extracted information needs can be afterwards validated by the participants to the conversation as relevant points to be analysed via a bibliographic search.

The "dialogue understanding" able to extract explicit information needs is very difficult while not an impossible module as it fits the CDF and it may be defined at different levels of complexity. The final solution may encompass all the possible "linguistic models" previously described, ranging from the less resource demanding vector space model based on the bag-of-model to the more demanding models doing syntactic and semantic analysis of the language. In any case such an approach could be very expansive, require a specific training, and reach not high performances.

3.3 Integrating semantic models: generic linguistic knowledge and specific domain knowledge

Natural language is rich of information and, as a consequence, very ambiguous. Words may convey very different meaning while different words may be used to express the same concept. Natural language understanding systems often try to tackle with this problem using some sort of background knowledge. This is often organised in so-called semantic dictionaries that explicitly deal with the problem of defining equivalences among different linguistic items (i.e. words or word sequences). These resources often take the form of networks of words (like in WordNet [Miller, 1995] and EuroWordNet [Vossen, 1998]) stating a strict equivalence among different words with the notion of synonymy sets (called *synsets*). These synsets are then organised in *is-a* hierarchies. These resources may be coupled with a graded activation of these relationships among words, that often take the form of probabilities [Resnik, 1995]. Such kind of measures may be used to adapt the resource to a particular domain, as it is suggested in [Basili et al., 2004] and [Zanzotto and Stellato, 2004].

For example, in WordNet, words as *planet*, *sun*, *celestial orbit* are covered even with the sense used in the space domain, and the spatial sense is the first sense. However, this is not true for all the space related words such as the word *space* itself.

General and domain specific linguistic semantic resources can be very relevant to improve the models over which the information need extraction system and the document clustering are based (see Fig. 3 (b)). In case both types of resources exist, there is a total agreement on their content by the involved experts (and it is not a default) still remains the problem of their interaction in a unique application. In fact their different knowledge models and structures may injure the semantic disambiguation process.

3.4 Learning the "mission ontology"

The *mission ontology* is a very relevant piece of specific knowledge that may be used to browse documents produced during the mission design. Glossaries attached to books are very clear example of the usefulness of such a resource.

There is a large research community trying to attach the problem of creating ontologies (see Sec. 4.3) in the field of ontology learning and in the field of terminology extraction. Some possible models are already available with satisfactory level of performance. For instance, it is possible to extract a domain specific terminology using a simple but effective method that ranks terms according to their domain relevance, which is computed as the frequency of each term in the documents (as outlined in [Daille, 1994]). As depicted in Fig. 3(c), this activity can not be completely automatic as a supervision is necessary: domain experts should be able, for example, to validate the terminology automatically extracted. As a result, a unique shared validated semantic resource could be accessible by the system to support its automatic process. With such a resource, performances of retrieving process could be improved.

At RTV a structured architecture has been defined and implemented to support the ontology engineering process (see Fig.6).

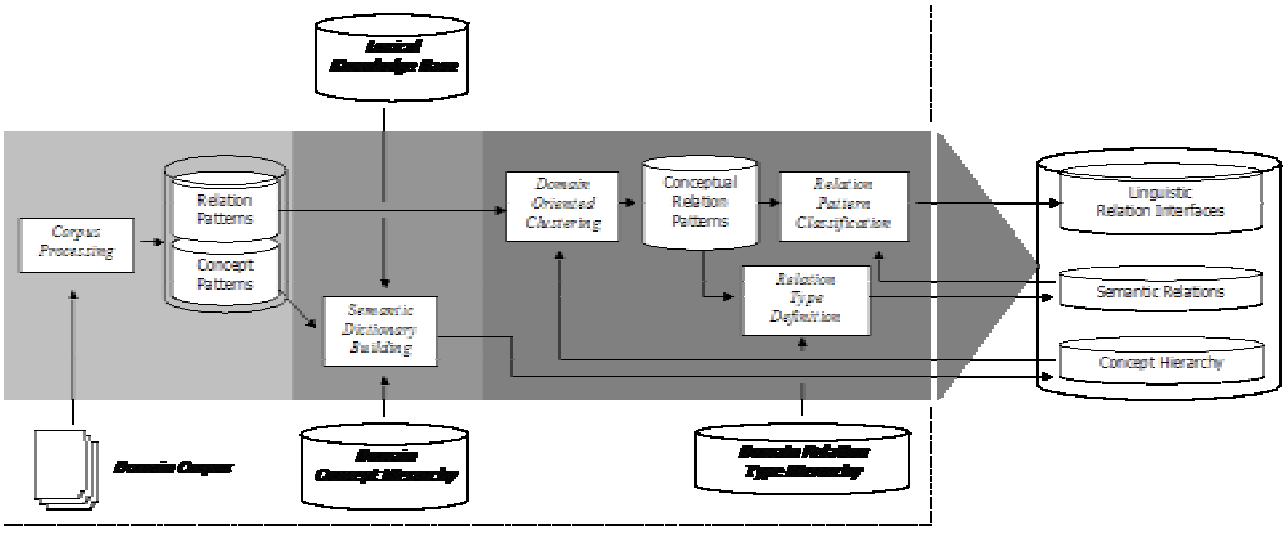


Fig. 6 Language-centre ontology learning architecture developed at RTV

Within such a framework it is possible to support knowledge base systems with a learning process enabling the system to be customized to the application through a domain related ontology.

3.4.1 Prototypical SHUMI Mission Ontology

Learning a specific domain ontology (or Domain Concept Hierarchy, DCH) is not an easy task. We propose a novel methodology for ontology learning, which makes use of both general semantic resources (Wordnet [Miller, 1995]) and a specific domain corpus (a set of documents regarding *spacecraft design*). The RTV methodology for ontology learning will be extensively described in Sec.4.3. An overall description of the approach followed for SHUMI follows.

With regard to the spacecraft design task defined in the *Shumi* project, a primary objective is to learn a *mission ontology*, that is, an ontology structuring the specific knowledge related to spacecraft mission through a hierarchy of concepts and relations among them.

In such a framework the priority step of the learning process is the extraction of the domain terminology (that is, an organized list of terms) from the corpus texts (see Fig. 7). A *term* can be roughly defined to be a surface (linguistic) representation of a domain key concept. For example, in the *spacecraft design* domain a key concept could be “the orbit of a spacecraft”, while a possible related linguistic forms expressing such a concept could be either “spacecraft_orbit” or “orbital_trace_of_the_spacecraft” terms.

The ontology creation process could be thus supported by the use of a related terminology.

In a first step candidate terms are extracted from the corpus through the use on NLP tools. Secondly the extracted terminology can be validate by a human expert, that is, only actual terms will be retained from the list of candidates. Finally, the validated terminology can be used to build or to feed the DCH: using either manual or semi-automatic techniques terms can be in fact used to identify concepts that can be possibly inserted into the mission ontology. The RTV approach thus aims to both create and incrementally improve the DCH, through the semantic analysis of domain documents, such mission reports and meeting minutes in the framework of spacecraft design.

As a second step the ontology can be enriched with a further form of domain knowledge: the *relational concepts*. Roughly, relational concepts can be considered to be semantic relations among domain concepts. They semantically link concepts thus representing a first structured representation of *events/facts*. Relational concepts get expressed in texts mainly through surface linguistic forms (from now on *surface forms*), which are particular forms of verb phrases, that is, semantically generalized lexical fragments of text governed by a

verb. While terms play the role of syntactic realizations of domain concepts, in the same way, surface forms represent the syntactic expressions of relational concepts. In this framework, relational concepts thus express *domain events*: action or facts regarding entities (concepts) of the domain. A typical relational concept for the *spacecraft design* domain could be "satellite reaching orbit", that represents events in which a domain entity ("satellite") is related to another domain entity ("orbit") by the action/relation of "reaching". A surface linguistic form that expresses this relational concept could be: "The satellite gets close to the orbit".

As concepts can be represented by many terms, relational concepts can have more that one surface form (in the above example, another surface form could be "Satellite approaches orbit"). As previously mentioned, a surface form of the event is a generalized form of lexical knowledge: in facts, it represents a sort of normalization of one or more actual textual *sentences*. For instances the surface form "satellite approaches orbit" can be the generalized forms of two different fragments of corpus text, as "the first satellite approached its lunar orbit in 1965" and "the new satellite will reach the programmed orbit in 10 minutes".

Surface forms, as terms, are automatically extracted from the domain corpus and validated by a human expert, in order to create a specific knowledge resource of domain fact and events. In the RTV approach, surface forms, once validated, can be integrated into the DCH with their associated sentences and referring relation concepts, together with terms and concepts, using semi-automatic or manual techniques.

In order to demonstrate the feasibility of such a novel approach, the integration of a *Terminology Extraction* tool (for the extraction of terms) and a *Relation Extraction* tool (for the extraction of surface forms), specifically dedicated to spacecraft design has been set by RTV, together with a terminology validation interface supported by a dedicated Database. The terminology extraction process has been thus carried out over a spacecraft domain corpus provided by ESA, followed by the manual validation phase carried by ESA experts. Then, using as supporting knowledge the set of extracted terms, the ESA corpus has been again automatically analyzed to extract surface forms, to be validated by the ESA experts.

In the following sections the *Terminology Extractor* and the term validation interface will be described in detail; it will be then presented an analysis of the terminology extracted and of the successive validation. Furthermore follows a description of the *Relation Extractor* and an analysis of the surface forms obtained with the relative validation phase.

3.4.1.1 Terminology Extractor

The terminology extraction strategy adopted at RTV, that will be fully described in Sec.4.3.1 , relies on a deep syntactic understanding and analysis of corpus documents, carried out by in-house technologies, based also on statistical methods applied to lexical forms and widely assessed in the NLP community: they have been specifically tuned to the task.

The aim of *Terminology Extractor* is to provide a simple batch interface, from where it is possible to lunch and supervise the ontology creation software modules related to terminology. In our vision the architecture should be intended as the first tool to be executed in order to support the ontology design in a simplified and easy-to-use way.

The architecture is able to supervise the terminology creation process from the starting point (that is the acquisition of input corpus about spacecraft design) to the final stage of relevant ontological information extraction.

3.4.1.1.1 Architecture

The architecture of the Terminology Extractor has been structured around a *batch* layer, from which it is possible to lunch the different modules of the system, implemented in the *JAVA*, *C*, and *Prolog* languages, depending on the already available resources and the data formats. The modules are (see Fig. 7):

- pre-processing modules
- parsing module
- terminology extraction module
- terminology sorting module

These modules have been integrated, with further existing NLP technologies previously developed at RTV. The architecture foresees, in fact, further tools in order to carry out the terminology extraction: the syntactic parser CHAOS (see Sec. 4.2.1.3) and an XML parser among others.

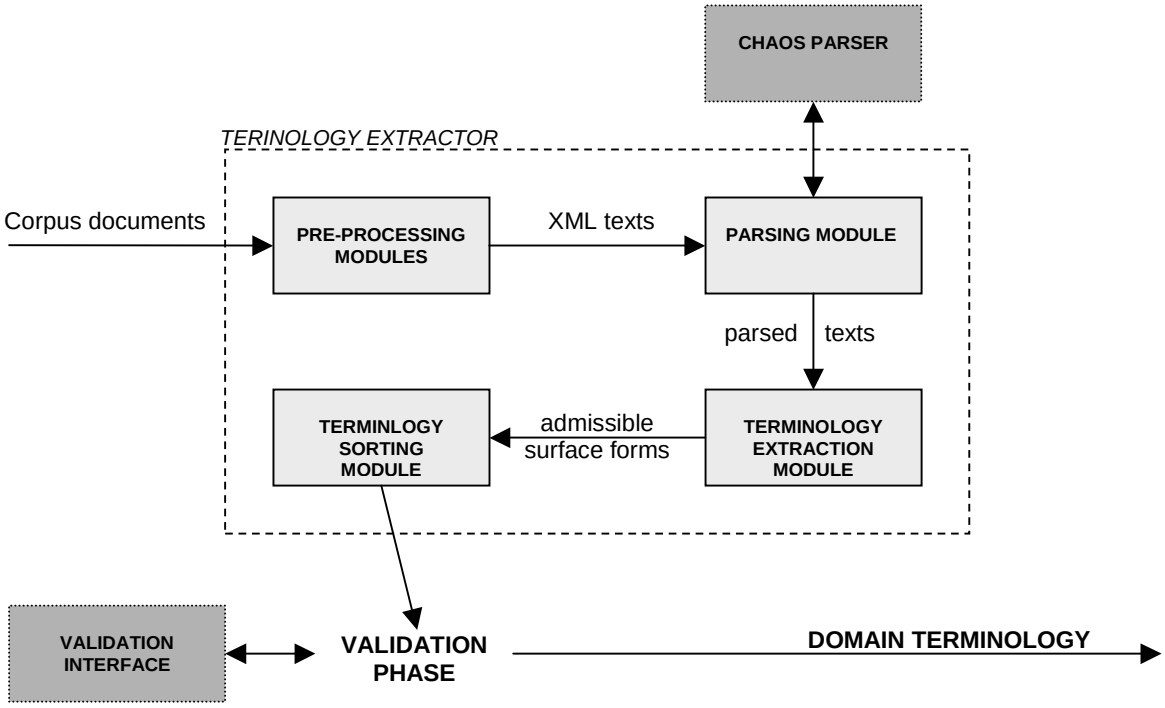


Fig. 7 A schema of the *Terminology Extractor* architecture

The **pre-processing modules** gets as input the corpus documents in ASCII textual format, splitting them into smaller files (for technical reasons) and converting them into XML file readable by the syntactic parser. The acquisition of input documents and the preparation of a readable corpus for the successive modules, is a critical step. The input documents (in form of ASCII text) are in fact usually the result of a previous format conversion phase (from PDF or HTML sources). Corpus related modules take care of checking such documents looking for possible corrections, conversions and adaptations. The pre-processing modules are also responsible for a first corpus syntactic analysis, whose aim is to identify paragraphs and other syntactic entities in the documents. This stage is needed to carry out the successive phases (terminology extraction and sorting). At the end of the corpus pre-processing phase, the input documents are thus made available to the successive modules in form of well-formatted XML documents.

The **parsing module** calls the Chaos parser sub-modules needed for the terminology extraction. Chaos is a robust and modular parser architecture developed at the RTV for carrying out syntactic analysis (currently it runs either English or Italian language texts). Thanks to the modular approach of Chaos, have been used only strictly required functionalities for the specific task of terminology extraction. These modules are: tokenizer, morpho-analyzer, yellow-pager, POS-tagger,NE-recognizer, chunker, VSA, SSA. For a description of these modules see Sec. 4.2.1.3.3. A deep understanding of the syntactic and morphological features of the documents content is, in fact, an unavoidable step for the identification of terminological expressions relevant for a specific domain.

The **terminology extraction module** extracts *admissible surface forms* from the previously parsed text; by identifying sequences of words with specific syntactic role, all the lexical expressions candidate to represent terms are identified in the text. Specific syntactic rules are used for the selection of the candidates. The syntactic roles are expressed through the well known *Brown Corpus Tag-Set* ([Green and Rubin, 1981]). For instance syntactic superficial sequences like *JJ NN* (an adjective followed by a singular common noun) and *NN NNS* (singular common noun followed by a plural common noun) are retained as useful possible surface forms of a term. As an example of the *JJ NN* form a candidate term could be *"lunar mission"*, while for the *NN NNS* form a candidate could be *"spacecraft projects"*.

Finally, the **terminology sorting module** sorts by relevance the list of previously produced admissible surface forms. Relevance is evaluated as the frequency with which each form has been met in the corpus: frequency can be considered as the best approximated measure to express term relevance, as underlined in [Daille, 1994]. The list of forms produced is the candidate terminology, i.e., the set of candidate terms still needs a final manual validation by a human expert, thus adding a semantic "related to the domain" relevance to each term.

For an in-deep description of the methodologies for terminology extraction adopted see Sec. 4.3.1.

3.4.1.1.2 Validation procedure

In order to be used as a support in the construction of the *mission ontology*, the candidate terms extracted by the Terminology Extractor still have to be validated by a human expert, since the automatic extractor can produce spurious and not "genuine" terms. The workflow developed at RTV thus foresees a successive phase: a *validation interface* has been developed to speed up the human validation procedure while helping the expert in understanding the semantic meaning driven by each term.

The task of the human expert is, in fact, to examine each candidate term, and then to accept/reject it as a valid term for the domain under analysis. The Interface supports the expert, by jointly providing him the term itself, together with all its contexts found in the corpus. That is, for each term all the document paragraphs in which the term appears are shown. In that way the expert can better understand, thanks to the provided contextual information, the current meaning of the term.

Candidate terms are shown to the experts in their syntactical-semantic form. Each term can be in fact a simple sequence of words (e.g. "spacecraft_mission") or a semantically generalized form. In the latter case the candidate term is formed by words and *Named Entities*. As defined see Sec.2.1.2.3 NE are generalizations for names of people, places, organizations, dates and amount of money, usually representing important entitites of the specific domain under analysis. An example of candidate term with a NE is "entity#ne#_mission" which indicates a mission of a generic *entity*, that is an organization, a person or a specific object. In such cases the task of validating the term appears to be more difficult than in the case of simple terms: that is when contextual information assumes a primary role in helping the expert to understand the meaning of the term. In the following table used Named Entities are reported, together with related entities.

NAMED ENTITY TYPE	EXAMPLES
Entity#ne#	
Organization#ne#	ESA, European Commission
Company#ne#	IBM, Boeing
Person#ne#	Yuri Gagaring, Neil Armstrong
Location#ne#	Frascati, Italy
Date#ne#	15/06/2003, 15th of May 2004
Money#ne#	340$, 23Euro

Nevertheless, while introduction of general classes as NE could make complex annotator task, it is an important generalization step in the overall extraction process. In fact NE subsumes a wider list of related term instances and represents one of the system's learning step over the automatic process.

The validation methodology is shown in Fig. 8, where the validation interface is implemented to help domain expert in their task[4].

[4] The left side of the interface shows to the domain expert the list of candidate terms to be validated, together with their frequency of appearance in the documents. Once a term has been selected, the right side will exhibit the documents paragraphs in which the term appears, in order to give to the expert further contextual information on the term. The

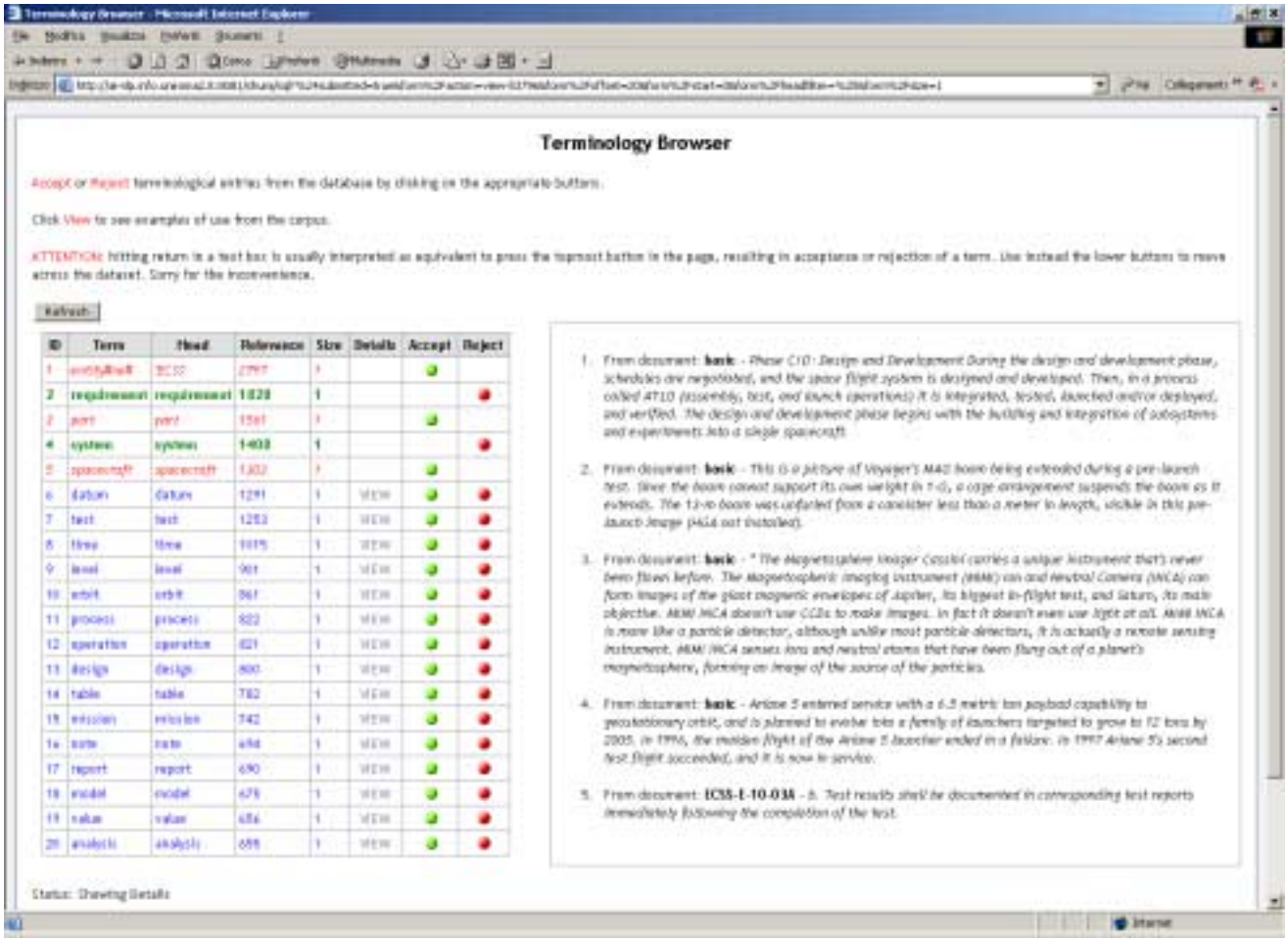


Fig. 8 A web-based terminology validation interface.

3.4.1.1.3 Document Collection structuring

In order to extract terminology related to Spacecraft Design, an ad hoc document collection has been provided by ESA, in form of:

- *study reports* of past space missions;
- general *tutorials* about space and spacecrafts;
- *glossaries;*
- *other space related material*.

expert can thus validate the terminology by simply clicking on the *Accept* or *Reject* box. The validation interface permits to easily browse the terminology in different ways, ordering terms by relevance (i.e. frequency in the corpus), by *syntactic head* (i.e. the governing word of the term, for example in "spacecraft_mission" the syntactic head is "mission"), by number *k* of words forming the term (*k-words term*) or in alphabetical order. The candidate terminology is stored in a SQL Database located at RTV. The validation procedure thus insists over the database, where rejection and acceptance actions are stored. In order to guarantee the expert to carry out the validation procedure from anywhere, the validation interface has been implemented in the OXF technology [OXF], which is able to manage a database interactive web-page. The Validation Interface thus allows the expert to remotely interact with the Database at RTV for the purposes of browsing or validating the automatically extracted terminology as well as for browsing samples from source documents. Such methodology also allows a multi-expert validation: insisting on a unique Database, that is automatically updated at each accept/reject action, different experts can always have an up-to-date view of the validation task.

The collection is thus intended to be a representative information repository of textual knowledge regarding the domain of interest, that can be used for the extraction of domain important concepts (lexicalised in terms).

The whole collection comprehends 32 textual sources, among formatted documents, reports and web pages: the problem of the heterogeneity of the data formats has been overcame thanks to the adaptability of the architecture. Most part of the sources are taken from ECSS [5] (European Cooperation for Space Standardization) Standards Series: the series documents describe standards that are intended to be applied for the management, engineering and product assurance in space projects and applications. Among other sources there are also two glossaries, one referring to space engineering, another to space in general: the integration of specific glossaries in document corpus is extremely useful for terminological extraction.

The data of the collection consists in 4,2 MB of rough plain text, that is, about 673.000 words and 3.500.000 characters. This amount of data can be considered enough for carrying out a preliminary and explorative terminology extraction task.

A list of the document collection follows:

- A CDF assessment study report of the World Space Observatory.
- Detailed space engineering glossary-like summary about spacecraft systems in HTML.
- The online glossary of the Science and Technology Homepage of ESA, in HTML.
- Basics of Space Flight, NASA, in HTML
- Basics of Space Flight Glossary, NASA, in HTML
- Rocket and Space Technology introduction, in HTML .
- Small Satellites Home Page, in HTML.
- Affordable Spacecraft: Design and Launch Alternatives, Princeton University.
- Solar Orbiter Technical Documentation, ESA, in HTML
- NASA Fact sheets – Satellites, NASA, in HTML
- Satellite Communication Tutorial, NASA, in HTML
- 21 ECSS Standards Series reports, ESA, in PDF

It must be noticed that the conversion process of some of the documents into an Extractor readable format has caused the injection of a few layout and superficial errors, that anyway didn't invalidate the whole terminology extraction procedure.

3.4.1.1.4 Comparative term analysis

The described methodology for extracting terms from corpus documents has been tested over the ESA document collection, in order to test the feasibility and to assess the soundness of our approach. The

[5] ECSS is a cooperative effort of the European Space Agency, national space agencies and European industry associations for the purpose of developing and maintaining common standards in space projects and applications.

extracted terminology has been then manually validated by ESA domain experts, in order to further verify the correctness of the extraction process and the quality of the term list.

The extracted terminology consist of 58.267 candidate terms (that is, terms that still have to be validate by a human expert). Different frequency threshold can be defined in order to cut the terms list between interesting and no-interesting concepts: 4820 candidate terms appear inside the corpus five or more times, 2520 ten or more times. At the end of the validation process, over the 58.267 candidate terms extracted, 7821 have been retained as useful: thus, about 14% of candidate terms have been accepted. In case of fixing a threshold frequency of 5 (that is, considering only candidate terms that appears in corpus 5 or more times), the accepted terms are 38% (with a coverage of 1814 terms over 4821).

Part of the rejected candidate terms are lexical entities that express concepts broadly used in the corpus, but that don't specifically refer to the spacecraft domain. Indeed, expression like *part, table, report, procedure* that are extracted as candidate terms refers to important concepts for the spacecraft domain, but are not specific enough to be considered as terms by validators.

In next Table the list of the first 20 more relevant candidate terms extracted is shown (the first columns indicates the term, the second its frequency inside the corpus):

TERM	*FREQ*
'ECSS'	2797
requirement	1828
part	1561
system	1408
spacecraft	1303
datum	1291
test	1253
time	1015
level	901
orbit	861
process	822
operation	821
design	800
table	782
mission	742
note	694
report	690
model	675
value	656
analysis	655

As it could be expected, the first 20 terms are all *1-word* terms, as it is more likely that such simple terms occurs in documents more than complex ones. The ESA expert validated positively 14 out of these 20

candidate terms. The validation of the first 20 extracted candidate terms has been reported in the following table:

TERM	*FREQ*	*validation*
'ECSS'	2797	NO
requirement	1828	YES
part	1561	NO
system	1408	YES
spacecraft	1303	YES
datum	1291	YES
test	1253	YES
time	1015	YES
level	901	NO
orbit	861	YES
process	822	YES
operation	821	YES
design	800	YES
table	782	NO
mission	742	YES
note	694	NO
report	690	NO
model	675	YES
value	656	YES
analysis	655	YES

Only *ECCS, part ,level, table, note* and *report* have been discarded by the expert. *ECCS* has been extracted by the Term Extractor as the most important term in the corpus due to the fact several documents are ECCS reports: then, the word “ECCS” appears very often, even in non informative parts of the documents (such references, subtitles, copyright), increasing, as a consequence, its relevance. The other rejected terms usually refer to words that while referring to important domain concepts (e.g. *report, part*) have a too generic sense or too many senses to be considered terms in all the effects.

Due to the possibility to automatically extract compound terms, a specific analysis has been carried on the length of terms. Extracted candidate terms have been organized in different sub-lists, according to the number of words that constitute the term itself: *k-word* terms (i.e. terms formed by *k* words) can be thus analysed together, in order to verify in general how interesting are terms formed by a certain number of words. It appears that the most interesting terms are those formed by 2 or 3 words, while the 1-word are too generic: in facts, over the 7821 validated terms, 1742 are 1-word, while 5786 are 2-3 words terms.

The following table summarizes the number of terms extracted for each *k-words* sub list:

k-TERM list	*Number of terms*
1-word list	12.142
2-words list	28.683
3-words list	11.926
4-words list	3.999
5-and-more-words list	1.517

Let us consider the list of the first 20 *2-words* and *3-words* candidate terms, as they usually carry a rich semantic information: for several of them a first generalization step has been carried on.

2-words TERM	*FREQ*
entity#ne#_system	126
application_datum	122
entity#ne#_packet	122
entity#ne#_requirement	118
entity#ne#_engineering	113
entity#ne#_state	105
magnetic_field	104
entity#ne#_model	104
solar_wind	101
entity#ne#_spacecraft	98
entity#ne#_datum	91
entity#ne#_test	90
technical_requirement	83
entity#ne#_design	81
entity#ne#_analysis	79
entity#ne#_mission	79
entity#ne#_orbit	77
entity#ne#_station	70
test_level	69
source_packets	61

3-words TERM	*FREQ*
entity#ne#_product_assurance	63
entity#ne#_source_packet	53
entity#ne#_'s_atmosphere	39
edition_of_the_normative_document	37
telemetry_source_packets	36
entity#ne#_project_management	25
entity#ne#_control_plan	25
entity#ne#_'s_surface	24
entity#ne#_revue_report	23
content_at_entity#ne#	22
entity#ne#_'s_orbit	21
www_location#ne#_note	20
entity#ne#_milestone_report	20
assurance_in_space_project	19
www_location#ne#_footnote	19
entity#ne#_problem_report	18
entity#ne#_'s_law	17
software_engineering_process	17
gray_segment_element	16
product_assurance_report	16

Word count in terms has been performed using a *stop list* , i.e. a list of words that shouldn't be included in the counting due to the poor semantic information they carry (for example articles and prepositions).

In the above lists *entity#ne#* is a Named Entity.

Let us consider, for example terms as *"solar wind"* and *"magnetic field"* represent in fact important concepts for a mission ontology, where spacecraft design related knowledge must be represented. Such terms are both a useful hint to identify concepts to insert into the ontology and to model the ontology itself.

Furthermore, the Extractor found terms related not only to a space mission domain in general, but also, more specifically, to the process of mission and spacecraft design. Terms like *technical requirement, software engineering process, product assurance report* emerge as powerful indicators of the design specific domain, while not directly related to space and space missions.

From the previously shown list of the 20 most frequent *2-words* candidate terms extracted, all the terms have been retained by the human expert, thus supporting our hypothesis of higher role of 2-3 words terms.

The relevance of our Extractor as an application tool is related to its ability to extract complex terms, that is, those that include semantic generalization through the use of NE. Such generalizations could be further exploited to automatically construct local hierarchies among the terms themselves, thus creating a structured terminology that could be directly integrated into the mission ontology. For example the generic term "entity#ne#_orbit" could be structured in a hierarchy, being a generalization of the more specific terms "planet#ne#_orbit" and "spacecraft#ne#_orbit". These kinds of terminological hierarchies could be developed by applying semantic measures (as the Conceptual Density, [Basili et al., 2004]) applied to semantic resources like Wordnet.

3.4.1.2 Relation Extractor

Once has been syntactically and semantically analysed the domain corpus, the *Relation Extraction* tool will extract surface forms by using as background knowledge the terms extracted by the Term Extractor: in fact in the process of identifying specific concepts to be inserted in the *Mission Ontology*, surface forms represent relations among terms. The strategy adopted at RTV to implement the tool is described in deep in Sec.4.3.1.2. In the next section the architecture of the tool will be described: it has been implemented and integrated into the batch interface already set-up for the Terminology Extractor, to speed-up a successive process for DCH building from corpus knowledge.

3.4.1.2.1 Architecture

The architecture of the Relation Extraction tool consider a framework similar to the Terminology Extractor: in fact both need a syntactic analysis of the corpus to extract forms of interest. The extraction and sorting modules are, on the contrary, quite different in both adopted techniques and statistical measures.

The Relational Extractor has been mainly implemented in JAVA, relying, when necessary, on C and Prolog sub-modules to carry out specific operations. Main modules are:

- parsing module
- surface forms extraction module
- surface forms sorting module

As in the case of terms extraction the parsing module strongly relies on the syntactic analysis produced by the CHAOS parser.

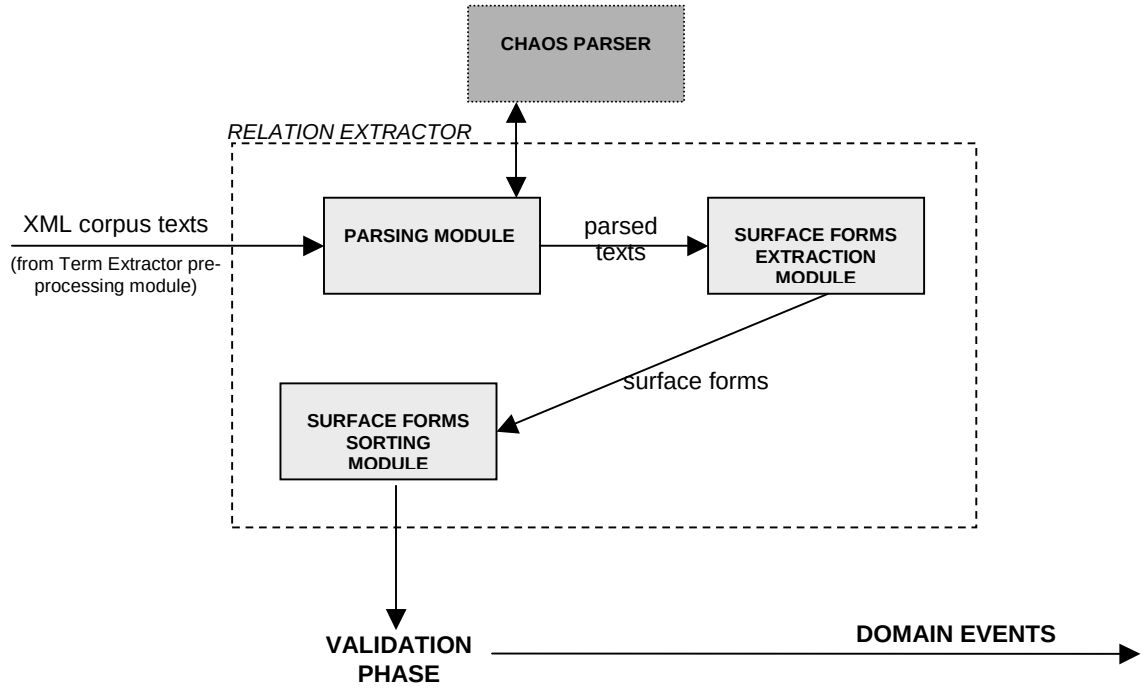


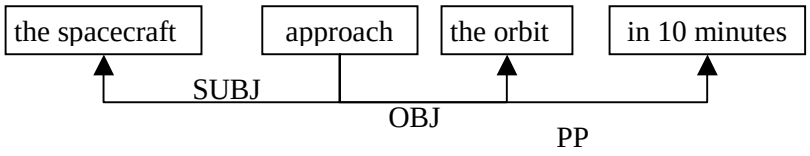
Fig. 9 A schema of the *Relation Extractor* architecture

Analogously the **parsing module** calls the Chaos sub-modules needed for the syntactic analysis of the text: tokenizer, morpho-analyzer, yellow-pager, POS-tagger, NE-recognizer, chunker, VSA, SSA. In the case of verb phrase extraction the VSA (Verb Shallow Analyzer) assumes a primary role, since it allows to identify the syntactic arguments of the verbs, thus enabling the extraction of sentences to be generalized into surface forms. The module takes as input the corpus documents in form of XML formatted files, obtained from the pre-processing modules of the Term Extractor.

The **surface forms extraction module** analyses the parsed text produced by the parsing modules and extracts from this text all the verb phrases that will be successively used to create the surface forms. A list of sentences is thus produced, each of which is represented by the governing verb and its arguments. For each argument it is indicated its lexical form and its syntactic role in the phrase. At this stage of processing, for example, the *sentence* could be represented as in the following:

approach((SUBJ, the spacecraft), (OBJ,the orbit),(IN,ten minutes))

The syntactic relations of the sentence can be better expressed through a graphical formalism, called *XDG* (*Extendend Dependencu Graph*) [Basili et al., 2001], that is also used as a common framework for the modules of the parser CHAOS. For instance, for the above sentence the XDG is:



The graph indicates the syntactic links between the verbs and its arguments (SOG=subject, OBJ=object, PP=generic argument).

The complete list of sentences extracted from the corpus is then passed to the successive module.

The task of the **surface forms sorting module** (taking as input the corpus sentences) is to generalize them into surface forms, and to successively rank them. The complex strategy adopted for the generalization step will be later on described in section 4.3.1.2. Once surface forms are produced, they are ranked accordingly to their frequency. The frequency of a surface form is calculated as the sum of the frequency of appearance of its corresponding sentences in the corpus. The ranked list of surface forms can be thus intended as *candidate surface forms* of relational concepts (a sort of prototypes), that still needs to be validated by a human expert. An example of extracted surface form, related to the above sentence, could be:

approach((SUBJ, spacecraft), (OBJ,orbit))

It must be noticed that, as in the case of terms, also in the extraction of surface forms Named Entity representations have been adopted, thus producing the following representation:

approach((SUBJ, mission#ne#), (OBJ,orbit))

where *mission#ne#* represent the entity class of the spacecraft missions. The surface form thus generalizes all the sentences which have “approach” as verb, “orbit” as object and any spacecraft mission as subject (i.e. “Mariner”, “Voyager” etc.).

3.4.1.2.2 Validation procedure

The validation procedure of the surface forms (carried out in cooperation with ESTEC experts) has implemented the same procedure of that used for term validation: the list of forms is provided to the domain expert, whose task is to accept or reject each instance.

The corpus used is the ESA corpus already analysed for terminology extraction; the previously validated terminology knowledge base is used in this step of processing to support the form extraction task.

3.4.1.2.3 Comparative Surface Form Analysis

From the whole ESA corpus 110.688 distinct surface forms have been automatically extracted. These forms have thus been supplied to a team of ESA experts for the validation procedure. Due the huge amount of time needed by humans to validate verbal phrases, only a part of them have been, at the moment, taken into consideration for validation (specifically groups of 5000 phrases ordered alphabetically by verb). Thus, out of the original 110.688 forms, currently only 45.000 have been validated and then analysed.

In order to give a deeper syntactic meaning to the surface forms (whose need will be clarified in the following lines), during the extraction procedure the set of Named Entities has been enriched with new, specific to the domain, Named Entities:

NAMED ENTITY TYPE	EXAMPLES
Mission#ne#	Apollo13, Voyager
Space_Organization#ne#	ESA, NASA, ASI
Space_Program#ne#	Spaceguard, NEO, JSGA
Celestial_corp#ne#	Sun, Moon, galaxies….

The list of Named Entities has been extracted from dedicated dictionaries available on-line and from specific documents regarding astronomy and spacecraft.

The ESA experts have retained as useful for the spacecraft design domain 9.299 out of the 45.000 taken into consideration. The percentage of correct forms, 21%, seems to be quite good considering that the procedure of surface form extraction is affected by the problem of *overgeneration*, that is, each verb sentence met in the corpus creates several related surface forms, some of which can be sometimes too general to be considered interesting. It is very common to get 4 or 5 surface forms from only one sentence. There are also extraordinary sentences that overgenerate much more. For example the sentence:

provide((SUBJ,innovation),(DIROBJ,view),(FOR,study),(OF,corona),(WITH,,mission#ne#))

produces the following surface forms:

provide((with,'mission#ne#'))
provide((subj,null),(with,'mission#ne#'))
provide((of,null),(with,'mission#ne#'))
provide((dirobj,null),(with,'mission#ne#'))
provide((dirobj,null),(subj,null),(with,'mission#ne#'))
provide((dirobj,null),(of,null),(with,'mission#ne#'))
provide((dirobj,view),(with,'mission#ne#'))
provide((dirobj,view),(subj,null),(with,'mission#ne#'))
provide((dirobj,view),(subj,innovation),(with,'mission#
provide((dirobj,view),(of,null),(with,'mission#ne#'))
provide((dirobj,view),(of,corona),(with,'mission#ne#'))

among which some are too generic and thus meaningless, as:

provide((WITH,,mission#ne#))

Moreover, we have to consider another problem: low score is also related to the inner difficulty of the validation procedure. In facts, contrary to what happens for term validation, surface forms are sophisticated and complex syntactic and semantic structures: the exact comprehension of the actual meaning under lied by a form is thus not an easy task. Then, it must not be excluded that many of the discarded surface forms actually express important generalized information for the domain, which could not be captured by the expert analysis.

The following table shows the first 20 more relevant (that is, more frequent) surface forms validated as meaningful by the ESA experts, while other relevant forms, suggested by the system, have been discarded by experts (see later on in the section for details):

Validated Surface Form	*FREQ*
perform((subj,test))	75
conform((to,requirement))	56
meet((dirobj,requirement))	50
conform((subj,null),(to,requirement))	42
do((subj,service))	31
conduct((subj,test))	30
conform((dirobj,null),(to,'space_organization#ne#'))	30
conform((to,'space_organization#ne#'))	30
conform((dirobj,null),(dirobj2,null),(to,'space_organization#ne#'))	28
perform((subj,analysis))	26
launch((dirobj,vehicle))	24
conform((dirobj,null),(subj,null),(to,'space_organization#ne#'))	24
launch((dirobj,spacecraft))	24
conform((subj,'entity#ne#'))	24
conform((subj,null),(to,'space_organization#ne#'))	24
conform((subj,'entity#ne#'),(to,null))	23
conform((dirobj,null),(dirobj2,null),(subj,null),(to,'space_organization#ne#'))	23
consider((subj,'entity#ne#'))	23
launch((dirobj,satellite))	23
perform((dirobj,function))	23

It must be firstly noticed that the frequency with which the surface forms appear in the corpus is not comparable to the frequency of the terms, as these are much more specific lexical and syntactic expression than the latter. For example the most frequent retained term, “requirement”, appears in the corpus 1828 times, while the most frequent surface form, “perform((subj,test))”, only 75.

As it can be inferred from previous table, most of the surface forms retained by the experts are governed by verbs whose driven semantic *meaning in phrases* usually directly refers to events regarding planning and design, such as *conform, perform, meet* . That is, these verbs, used in specific context (i.e. spacecraft design) assume a particular meaning. For example, the verb “meet”, that in general can assume many senses and semantic values (“meet a friend”, “meet death”, “meet point of conjunction”: according to *The Concise Oxford Dictionary* “meet” has 10 senses), in a spacecraft design context it assumes a specific semantic value, that corresponds to the definition:

“satisfy, give valid answer to”

This “sense restriction” (from 10 possible senses to only 1) has two important implications in the overall automatic process. From one side it is a confirmation of the importance of surface forms in order to build a correct DCH: in fact from the surface forms it emerges how verbs behave either semantically or syntactically

in specific domains. That is, for each verb it is possible to study how it is used in the domain, which is its preferred syntactic structure, and with which arguments it usually occurs. From the previous example, for instance, it emerges that "meet" in the spacecraft design domain is usually accompanied simply by a direct object whose lexical form is "requirement". Again, the verb "conform" is particularly common with an "of" preposition to which is attached "requirement" or "space_organization#ne#". In such a way, it would be possible to create actual *domain syntactic-semantic subcategorization frame*s for each verb: that is, to insert into the DCH the surface forms associated to each verb that better express the behaviour of the verb in that specific domain.

Another important implication is that restricting the verb senses a sort of *verb sense disambiguation* is automatically carried out. This is an important achievement, in an automatic process to be able to suggest precise semantic value to polisemic verbs (thus excluding uncommon sense for the domain in question) is in fact of great help in automatic semantic analysis of textual corpus.

In order to better understand the flavour of the surface forms extraction and validation task, we propose in the following a brief and deeper analysis of a few forms. For example, consider the surface form:

point((TO,'celestial_corp#ne#'))

This form has frequency 4. In fact, there are in the ESA corpus the following four sentences which refers to it:

- *"A spacecraft's attitude must be stabilized and controlled so that its **high-gain antenna may be accurately pointed to Earth**"*
- *"The whole **spacecraft** had to be maneuvered **to point the HGA to Earth for communications**."*
- *"A spacecraft's attitude, its orientation in space, must be stabilized and controlled so that **its high-gain antenna may be accurately pointed to Earth**"*
- *"The reference frame used in the visualisation is a **rotating frame**, which **has its X-axis pointing to the Sun** and its Z-axis pointing to the ecliptic north pole"*

As it can be seen the surface form is a good syntactic-semantic generalization for all the four sentences. For example in the first sentence the verb "point" is associated with the argument "to Earth". In that case the surface form is both a syntactic and semantic generalization. From a syntactic point of view in fact, the sentence has more arguments than those expressed in the surface form (the subject "antenna"), and it is thus a specialization of the surface form. The semantic generalization consists in the named entity "celestial_corp#ne#" which is in fact a general expression for planets and other space objects, then also for "Earth". It is thus interesting to notice in previous example how "celestial_corp#ne#" gets specified in the corpus with the two entities "Earth" and "Sun".

As a further example of overgeneration lets consider the following corpus sentence:

"Voyager 2 leaves Earth at about 36 km/s relative to the Sun"

This sentence produces six surface forms:

leave((dirobj,'celestial_corp#ne#'),(subj,'mission#ne#'))

leave((dirobj,'celestial_corp#ne#'))

leave((dirobj,'celestial_corp#ne#'),(subj,null))

leave((dirobj,'celestial_corp#ne#'),(dirobj2,null))

leave((dirobj,'celestial_corp#ne#'),(dirobj2,null),(subj,null))

leave((dirobj,'celestial_corp#ne#'),(dirobj2,null),(subj,'mission#ne#'))

As it can be seen not all the forms could be retained as useful for the single example, but it must be taken into consideration that each of the six forms is a good candidate for generalizing many other sentences of the same corpus. So, even if overgeneration can produce some apparently meaningless forms, it is necessary to consider all the forms in order to have an effective extraction process.

Among the previous six forms only the sixth has been retained as useful by the human expert , while the third has been discarded, maybe due to its excessive value of generality (it does not specify the semantic type of the subject, which is simply indicated by "null", that means "any word").

As stated above, due to the inherent difficulty in validating the forms, some of these that could be retained as useful has been discarded by the human expert, maybe related to an understandable lack of comprehension of the surface form construction. For example the following surface forms could have been retained as useful, while they have been discarded by the validators:

provide((dirobj,capability))

provide((dirobj,service))

apply((subj,requirement))

match((dirobj,requirement),(dirobj2,circumstance))

For example, the third form corresponds, among the other, to the following sentence:

"Random vibration tests

a. For random vibration acceptance test operations, the ***same requirements as for qualification tests shall apply*** *(see 5.3.3.5.3) with the exception that the power spectral density shall be adjusted to the acceptance spectrum as minus 6 dB shall be applied)."*

which could express an information related to the spacecraft design domain.

Furthermore, the first form (*provide((dirobj,capability))*) also subsumes interesting domain specific sentences, as:

"the Advanced Launch System, the Advanced MannedLaunch System, and NASP-derived vehicles. Of these, only ***National Aero-Space Plane (NASP)- derived vehicles (NDVs) are intended to provide a survivable capability for wartime launch."***

"Thrust modulation. ***The thruster shall provide the capability of being modulated in high- and low-frequency modes*** *if required by AOCS."*

*"****The space link shall provide a time correlation capability*** *that enables the time maintained on the spacecraft, the onboard time, to be correlated with the time maintained on the ground".*

*"****The space network shall provide a time distribution capability*** *that enables a reference time maintained onboard the spacecraft to be distributed throughout the space network."*

These difficulties in capturing the meaning under lied by some of the surface forms should be thus overtaken in order to obtain a better validation, and then a general correct result in building the DCH. Indeed, a fully satisfactory DCH can be obtained only if all the information enclosed in the corpus (in forms of validated and meaningful terms and surface forms) are provided. The process could be improved by supplying the human experts with examples of corpus sentences related to each surface forms (similarly to what has been done for terms using the validation interface); in that way the expert could better understand the meaning of the form. The implementation of such a validation interface is not trivial, for technical reasons, but it could be implemented in a full working final architecture.

A final observation about surface forms relates to an interesting aspect of verb behaviour in specific domain. As already stated, it could be possible to identify a sort of *domain syntactic-semantic subcategorization frame*s for the verbs extracted from the corpus. To enhance such an intuition it could be useful to rely not only on forms which express their arguments through simple lexical forms (for example *(OBJ,requirement)*) or through Named Entities (for example *(OBJ,mission#ne#)*). In fact, examining the surface forms related to a single verb, it would be possible to obtain new forms, which even better generalize the semantic behaviour of the verb in the specific domain. Lets take for instance the verb “absorb”. Some of its surface forms (retained by the ESA experts) are:

absorb,((subj,'x-rays'))
absorb,((subj,dust))
absorb,((subj,gas))
absorb,((subj,radiation))
absorb,((subj,substrate))
absorb,((subj,atmosphere))

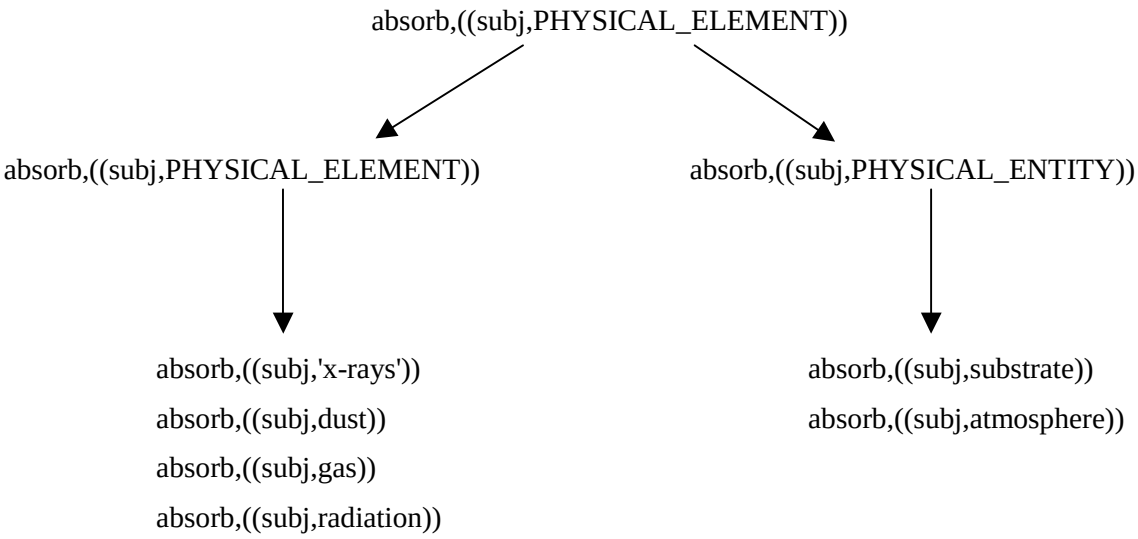
As it can be seen, in order to capture the semantic behaviour of the verb in a spacecraft context, it could be useful to aggregate the above forms into two groups:

absorb,((subj,'x-rays'))	absorb,((subj,substrate))
absorb,((subj,dust))	absorb,((subj,atmosphere))
absorb,((subj,gas))	
absorb,((subj,radiation))	

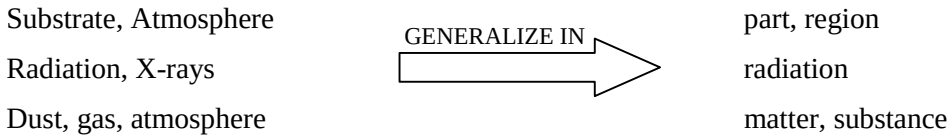
These two groups express two distinct semantic behaviour of “absorb”. The first group aggregates subjects of the verb which all refer to a *physical element*, while the second subject of physical macroscopic entities. Thus, in this sense, the verb reveals to drive two different deep meaning, which can be collapsed in two new general forms:

absorb,((subj,PHYSICAL_ELEMENT))
absorb,((subj,PHYSICAL_ENTITY))

Such a deeper analysis could enrich a DCH with a useful *surface form hierarchy* like:



To carry out automatically the process of hierarchy building, semantic resources must be used in order to find name generalization: WordNet could be one of such resources. For the above example, using name clustering techniques based on WordNet, such as the *Conceptual Density* [Basili et al., 2004], available at RTV, the subjects of the forms could be clustered as:



3.4.1.2.4 Surface Forms and related sentences

In this paragraph we will show groups of surface forms with some of their associated sentences extracted from the corpus, in order to further clarify the meaning of the extraction procedure.

Surface Forms:

lex(provide,1,[(for,study),(subj,'celestial_corp#ne#')]).
lex(provide,1,[(for,study),(of,null),(subj,'celestial_corp#ne#')]).
lex(provide,1,[(for,study),(of,magnetism),(subj,'celestial_corp#ne#')]).
lex(provide,1,[(for,null),(subj,'celestial_corp#ne#')]).
lex(provide,1,[(for,null),(of,null),(subj,'celestial_corp#ne#')]).

Textual fragments :

Scientific Rationale

2.2.1 Photospheric magnetic flux elements

While our basic understanding of the equilibrium properties of the Sun has been validated by helioseismology, most notably by the impressive results from SOHO, our understanding of the non-equilibrium properties u primarily associated with magnetic fields u remains poor. Magnetic fields play a crucial role throughout astrophysics, ranging from the formation of stars to the extraction of energy from supermassive black holes in galactic nuclei. ***The Sun provides a natural laboratory for the study of cosmic magnetism under conditions not accessible on Earth and on scales not resolvable in distant astronomical objects****.*

****

Surface Forms:

lex(provide,1,[(dirobj2,null),(of,'celestial_corp#ne#')]).
lex(provide,1,[(dirobj2,null),(of,'celestial_corp#ne#'),(subj,null)]).
lex(provide,1,[(dirobj2,null),(of,'celestial_corp#ne#'),(subj,'entity#ne#')]).
lex(provide,1,[(dirobj2,'entity#ne#'),(of,'celestial_corp#ne#')]).
lex(provide,1,[(dirobj2,'entity#ne#'),(of,'celestial_corp#ne#'),(subj,null)]).
lex(provide,1,[(dirobj2,'entity#ne#'),(of,'celestial_corp#ne#'),(subj,'entity#ne#')]).

Textual fragments :

1.

The FDT consists of a refractor with a focal length which varies between 233 mm and 888 mm. ***Thetelescope provides a full-disc image of the Sun of constant diameter on the detector during all orbitalphases between 0.22 and 0.9 AU****. The zoom lens consists of a classical afocal PNP triplet followed by afixed aperture stop and a camera lens which provides a focus at a fixed position. All lenses will besinglets as chromatic aberration is not a concern with the narrow-band filtergraph.*

2.

This is where an EUV Imager (EUI) on the Solar Orbiter can fully exploit the unique capabilities of the mission. EUI will provide EUV images with an angular resolution increased by one order of magnitude in order to reveal the fine-scale structure of coronal features. The closeness of the Sun makes this task easier. ***EUI will also provide full disc EUV images of the Sun in order to reveal the global structure of inaccessible regions such as the "far side" of the Sun and the polar regions.***

****

Surface Forms:

lex(provide,1,[(dirobj,opportunity),(to,'celestial_corp#ne#')]).
lex(provide,1,[(dirobj,opportunity),(subj,null),(to,'celestial_corp#ne#')]).
lex(provide,1,[(dirobj,opportunity),(subj,'mission#ne#'),(to,'celestial_corp#ne#')]).

Textual fragments :

***The Solar Orbiter will provide the first opportunity of going closer to the Sun and into the inner heliosphere to 0.21 AU**, but unlike Helios and Ulysses, the Solar Orbiter will also carry powerful, high-resolution optical instruments together with the in-situ instruments.*

****

Surface Forms:

lex(leave,1,[(dirobj,'celestial_corp#ne#'),(subj,'mission#ne#')]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,s)]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,s),(subj,null)]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,s),(subj,'mission#ne#')]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,null)]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,null),(subj,null)]).
lex(leave,1,[(dirobj,'celestial_corp#ne#'),(dirobj2,null),(subj,'mission#ne#')]).

Textual fragments :

Voyager 2 Gravity Assist Velocity Changes

***Voyager 2 leaves Earth at about 36 km/s relative to the sun.** Climbing out, it loses much of the initial velocity the launch vehicle provided. Nearing Jupiter, its speed is increased by the planet's gravity, and the spacecraft' velocity exceeds solar system escape velocity. Voyager departs Jupiter with more sun-relative velocity than it had on arrival. The same is seen at Saturn and Uranu.*

****

Surface Forms:

lex(provide,2,[(with,'mission#ne#')]).
lex(provide,2,[(subj,null),(with,'mission#ne#')]).
lex(provide,2,[(of,null),(with,'mission#ne#')]).
lex(provide,2,[(dirobj,null),(with,'mission#ne#')]).
lex(provide,2,[(dirobj,null),(subj,null),(with,'mission#ne#')]).
lex(provide,2,[(dirobj,null),(of,null),(with,'mission#ne#')]).
lex(provide,1,[(dirobj,view),(with,'mission#ne#')]).
lex(provide,1,[(dirobj,view),(subj,null),(with,'mission#ne#')]).
lex(provide,1,[(dirobj,view),(subj,innovation),(with,'mission#ne#')]).

lex(provide,1,[(dirobj,view),(of,null),(with,'mission#ne#')]).
lex(provide,1,[(dirobj,view),(of,corona),(with,'mission#ne#')]).

Textual fragments :

Scientific Rationale elements, hydrogen and helium. In particular, ***this innovation will provide new and better views of the solar corona than possible with SOHO presently and STEREO in the future****. In particular, the Solar Orbiter will be able to provide the first UV images of the full corona for the two most abundant elements, the first global maps of the solar wind outflow, the first images of the He II coronal emission.*

****

Surface Forms:

lex(provide,1,[(dirobj,observation),(of,region),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,corona),(of,region),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,null),(of,region),(subj,'mission#ne#')]).

Textual fragments :

Through a novel orbital design, the Solar Orbiter will explore in-situ the innermost heliosphere and will scrutinise, through high-resolution remote-sensing, the Sun and its atmosphere, while flying closer to the Sun than any other spacecraft and out of the ecliptic to higher latitudes. ***From there, the Solar Orbiter will provide the first observations of the polar regions of the Sun and the whole equatorial corona****.*

****

Surface Forms:

lex(provide,1,[(dirobj,observation),(of,corona),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,expansion),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,expansion),(of,null),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,expansion),(of,corona),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,null),(of,corona),(subj,'mission#ne#')]).

Textual fragments :

Even with new dedicated missions such as STEREO, it is impossible to determine the mass distribution of large-scale structures such as streamers and the true longitudinal extent of Coronal Mass Ejections (CMEs).

The Solar Orbiter will provide the first observations of the complete equatorial corona and its expansion in the equatorial plane*. Furthermore, it will provide the third dimension of CMEs.*

****

Surface Forms:

lex(provide,1,[(dirobj,observation),(dirobj2,corona),(subj,'mission#ne#')]).
lex(provide,1,[(dirobj,observation),(dirobj2,corona),(of,null),(subj,'mission#ne#')]).

Textual fragments :

Through a novel orbital design, the Solar Orbiter will explore in-situ the innermost heliosphere and will scrutinise, through high-resolution remote-sensing, the Sun and its atmosphere, while flying closer to the Sun than any other spacecraft and out of the ecliptic to higher latitudes. ***From there, the Solar Orbiter will provide the first observations of the polar regions of the Sun and the whole equatorial corona.***

****

Surface Forms:

lex(propose,1,[(dirobj,'mission#ne#')]).
lex(propose,1,[(dirobj,'mission#ne#'),(subj,team)]).
lex(propose,1,[(dirobj,'mission#ne#'),(subj,null)]).

Textual fragments :

Solar Orbiter

A High-Resolution Mission to the Sun and Inner Heliosphere

Assessment Study Report

July 2000

SCI(2000)6

FOREWORD

The Solar Orbiter Mission was proposed to ESA by an international team of scientists.

****

Surface Forms:

lex(perform,1,[(dirobj,measurement),(subj,'mission#ne#')]).
lex(perform,1,[(dirobj,measurement),(dirobj2,null),(subj,'mission#ne#')]).
lex(perform,1,[(dirobj,measurement),(dirobj2,image),(subj,'mission#ne#')]).

Textual fragments :

NOTE of two contributions: the sound power transmitted to the space vehicle via airborne transmission path (see subclause 5.1.20.2.2) and via the structure borne transmission path (see subclause 5.1.20.2.3)

5.1.20.2.2 Equipment airborne sound pressure measurement a. ***The equipment sound power measurement should be performed in accordance with the ISO 3740.***

****

Surface Forms:

lex(perform,1,[(dirobj,manoeuvres),(subj,'mission#ne#')]).
lex(perform,1,[(dirobj,manoeuvres),(dirobj2,null),(subj,'mission#ne#')]).
lex(perform,1,[(dirobj,manoeuvres),(dirobj2,increase),(subj,'mission#ne#')]).

Textual fragments :

Using solar electric propulsion (SEP) in conjunction with multiple planetary swing-by manoeuvres, it will take the Solar Orbiter only two years to reach a perihelion of 45 solar radii at an orbital period of 149 days. Within the nominal 5 year mission phase, ***the Solar Orbiter will perform several swing-by manoeuvres at Venus, in order to increase the inclination of the orbital plane to 30o with respect to the solar equator****. During an extended mission phase of about two years the inclination will be further increased to 38.*

****

Surface Forms:

lex(allow,1,[(dirobj,measurement),(subj,'mission#ne#')]).
lex(allow,1,[(dirobj,measurement),(dirobj2,null),(subj,'mission#ne#')]).
lex(allow,1,[(dirobj,measurement),(dirobj2,irradiance),(subj,'mission#ne#')]).

The influence of faculae and sunspots on the value of the solar constant value can be eliminated only by observing the Sun above the poles, where these features are not expected to occur. Therefore, the Solar Orbiter, at its highest inclination with respect to the ecliptic, will allow a measurement of the "true solar irradiance" (TSI), as well as a quantification of the effects of solar activity through a comparison of these

high-latitude measurements with the traditional irradiance measurements simultaneously made in the ecliptic.

Surface Forms:

lex(point,4,[(to,'celestial_corp#ne#')]).

Textual fragments :

1.

A spacecraft's attitude must be stabilized and controlled so that its high-gain antenna may be accurately pointed to Earth*, so that onboard experiments may accomplish precise pointing for accurate collection and subsequent interpretation of data, so that the heating and cooling effects of sunlight and shadow may be used intelligently for thermal control, and also for guidance: short propulsive maneuvers must be executed in the right direction.*

2.

HGAs may be either steerable or fixed to the spacecraft bus. The Magellan HGA, which also served as a radar antenna for mapping and as a drogue for aerobraking, was not articulated; ***the whole spacecraft had to be maneuvered to point the HGA to Earth for communications.*** *HGAs can also serve as a fine sunshade. Magellan&'s and Cassini's are good examples. Mission ops people routinely pointed it to the sun in order to provide some needed shade for the rest of the spacecraft.*

3.

Attitude and Articulation Control. ***A spacecraft's attitude, its orientation in space, must be stabilized and***

controlled so that its high-gain antenna may be accurately pointed to Earth, *so that onboard experiments may accomplish precise pointing for accurate collection and subsequent interpretation of data, so that the heating and cooling effects of sunlight and shadow may be used intelligently for thermal control, and so that propulsive maneuvers may be executed in the right direction.</P></PP>*

4.

21.1.1 Visualisation of the WSO mission

The PTB is capable of simulating the WSO mission and visualising the mission using a 3Ddisplay. The following items are available in the 3D visualisation:

** Satellite model as imported from Configuration*

** Solar arrays deployment mechanism*

** Telescope baffle and cover mechanisms*

** Block DM upperstage including thrust cone*

** Telescope field of view cone*

** Satellite trajectory trail*

** Environment: Earth, Moon, Sun and stars*

** Earth shadow zone*

The reference frame used in the visualisation is a rotating frame, which has its X-axis pointing to the Sun and ***its Z-axis pointing to the ecliptic north pole.*** *In this frame, the L2 orbit is best visualised, as compared to an inertial frame (e.g. equatorial J2000), where the position of the spacecraft would evolve around the Earth over time.</P></PP>*

Surface Forms:

lex(provide,27,[(dirobj,capability)]).

Textual fragments :

1.

"the Advanced Launch System, the Advanced MannedLaunch System, and NASP-derived vehicles. Of these, only ***National Aero-Space Plane (NASP)- derived vehicles (NDVs) are intended to provide a survivable capability for wartime launch."***

2.

"Thrust modulation. ***The thruster shall provide the capability of being modulated in high- and low-frequency modes*** *if required by AOCS."*

3.

*"****The space link shall provide a time correlation capability*** *that enables the time maintained on the spacecraft, the onboard time, to be correlated with the time maintained on the ground".*

4.

*"****The space network shall provide a time distribution capability*** *that enables a reference time maintained onboard the spacecraft to be distributed throughout the space network."*

4. Enabling Methodologies

As outlined in previous sections, the main goal of an *Automatic Assistant* architecture in a CDF environment is to implement techniques able to query a large document collection, in order to retrieve interesting documents related to the on going project, hopefully clustering them and eventually extracting specific information.

Different approaches can be followed to implement a robust and efficient system for querying document collections, based on simple but effective statistical model, or on complex syntactic-semantic methodologies. In this section we discuss in deep the approaches we would follow at RTV in order to build the envisioned architecture.

4.1 Bag-of-Word Abstraction

As already described in Sec. 2.1.1.5, in order to find information related to a query in a generic document collection, the simplest way is to look at each document using a Vector Space model on a Bag of Word Abstraction. In this approach each document is represented by a set of terms gathered with a related weight, and the union of the terms extracted from all the document of the collection is called *document space* [Salton, 1983]. Terms are thus independent entities listed in the vector space.

Heavy effort has been devoted to the definition of term weighting measures. Roughly speaking, the weight should express "an estimation of the usefulness of the given term as a descriptor of the given document" [Greengrass, 2000], that is, the estimation of its usefulness in distinguishing the given document from the others in the collection. The most widely used measure is *tf***idf* (term frequency * inverse document frequency). This measure assigns a higher weight to those terms that appear frequently in the given document (*tf*) and rarely in the rest of the collection (*idf*). In turn, different measure have been proposed for term frequency (*tf*), ranging from the simple count of occurrences of the term in the document normalized to the most counted term in the same document, to more complicated normalized measures. *Idf* is usually calculated as:

$$idf = \ln\left(\frac{N}{n}\right)$$

where N is the number of documents in the collection and n the number of documents that contain the term.

For each document a vector is thus calculated; its length corresponds to the dimension of the document space (i.e. the number of terms in the collection) , and its elements correspond to the weight assigned for the document to the term in the different dimension of the space. A similar vector is calculated for the query. *Similarity* between the query and each document is then evaluated , and documents will be ranked accordingly. The most common way to measure similarity is the *inner product* [Salton, 1989] between the query vector and the document vector:

$$\sum_{i=1}^{N} QT_i * DT_i$$

Where *QTi* is the weight assigned for the query to term *i* of the collection, and where *DTi* is the weight assigned for the document to term *i* of the collection. If weights have been previously normalized using a

cosine measure , the inner product is called *Cosine Similarity*: if two vectors are identical in the document space similarity is one, if they form a 90 degrees angle they don't have any term in common (null similarity).

Other measures for similarity are *distance metrics*, such the *Euclidean distance*, as well as measures that don't consider term frequency, as the *Dice Coefficient* [Rijsbergen, 1979]:

$$Dice = \frac{2w}{n1+n2}$$

Where *w* is the number of common terms in two vectors, *n1* and *n2* the number of non-zero terms in each vector.

At present, many implemented systems are able to carry out document querying using the Vector Space model on the Bag of Words approach, some of which will be later described in Sec.5.1.

In order to gain a deeper understanding of the information contained in the document collection, the Vector Space Model can be applied with different approaches (more sophisticated and rich than Bag of Words), based on syntactic and semantic analysis: see following sections.

4.2 Intermediate Syntactic-Semantic Language Interpretation

A second and more sophisticated approach to implement the proposed system is to rely on a robust syntactic and semantic analysis of the documents. The information content is foreseen to emerge in a more complex and rich form, enabling a deeper understanding of texts.

In Sec. 4.2.1 we introduce the concept of robust syntactic analysis (*parsing*) and we present the RTV approach in such a framework. In Sec. 4.2.2 robust semantic approaches to text analysis are discussed, underlying our point of view.

4.2.1 Robust parsing

4.2.1.1 Introduction

The ability of dealing with odd (i.e. ill-formed or simply partial) sentences is largely shown by humans. The human interpretation device is tolerant to phenomena like lack of the lexical information (e.g. foreign words), unknown words (e.g. proper nouns never met before), and odd grammatical constructions (e.g. gender disagreement, badly transcribed coordination structures or gaps in the information stream as in remote/telephonic dialogue). The above form of tolerance is what has been recently called *robustness* in NLP. The modelling of such phenomena within computational devices (as introduced in [Menzel, 1995]) is thus more than a relevant research area either for a better linguistic investigation as well as for design of large-scale NLP systems.

Robustness has been traditionally stressed as a general desirable property of any computational model and system. In the software engineering practice, *robustness* is (somewhat informally) defined as *the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions* [IEEE, 1990]. Although such a definition is satisfactorily used for any information system it requires more specification when used within a linguistic computational model or in NLP applications. In fact, it is needed a systematic definition of what kind of invalid input is here intended. For example, in syntax a formal definition of ungrammaticality is required. As this notion requires a complete definition of grammaticality, the circularity and moreover the criticality of the latter notion prevent from a systematic analysis. Furthermore, the notion of *stressful environmental conditions* has to be interpreted in the linguistic analysis. The stress comes from two major sources:

- external/exogenous stress that is incompleteness in the context, i.e. missing information in the source sentences or lack of competence in a wider context (e.g. inter-phrasal context or discourse).

- internal/inner/endogenous stress that is wrong/odd/misleading information in the sentence. This can relate to *legal* information, e.g. high levels of syntactic and/or semantic ambiguity, or to *illegal* input evidence. As the latter has been above considered as a case of noisy or invalid input, with the stressful information we will refer to the former case where higher ambiguity levels are considered.

Stressful environmental conditions (endogenous and exogenous) are also critical to be formally defined since a comprehensive model at different linguistic levels (even pragmatic) would be required.

As robustness in humans is a typical empirical phenomenon a different definition is required. It should thus be tighter to linguistic observation and consequently not expressed formally, at least in a full way.

4.2.1.1.1 Robustness in NL Parsing

When looking only to NL parsing activity, robustness is possibly more specifically defined. Robustness in parsing is tight both to invalid input and endogenous stress. As ill-formedness is a characteristic form of invalid input, ungrammaticality captures only this first aspect. Endogenous stress is mainly related to sentence complexity where lack of information (e.g. in the lexicon) may be the source of the major failures. When the parser is faced with very complex structures, the result is a lower accuracy in the recognition. Robustness should be achieved by preserving most of the information thus resulting in the so-called *graceful* degradation of performance [Menzel, 1995].

The above (somewhat informal) definition of *robustness* in NL parsing has thus to take into account some notion of *performance*. Syntactic parsers indeed fail to systematically propose the *correct* interpretations, mainly when exposed to large amounts of textual material. This is strictly true in real application environments. Performance criteria may largely vary among applications, where precision and coverage (two of the mostly used measures) may assume quite different relevance. A coherent view on robustness for NL parsing is thus tightly related to a relative notion of performance and any attempt to study it should take this into account.

Previous approaches to NL parsing robustness have proposed extensions of the grammatical system (i.e. more rules) or changes in the representation (e.g. probabilistic layers around grammatical frameworks, as in PCFG). One of the more systematic approaches [Menzel, 1995] relies just on a layered representation and on a modified processing (i.e. preference based reasoning) to enforce autonomy and support expectation driven disambiguation in NL parsing. It is to be also noticed that changes in the representation (especially of the parser output) usually characterized the so-called robust parsers where partial interpretations (e.g. NP chunks) are produced.

Robust parsers are also based on the notion of *under-specification.* Parsers able to expose partial results are inherently more robust in the sense defined before with respect to "monolithic" parsers. Under-specification is obtained as, for example, some information (e.g. PP dependences) is left unanalysed and a not fully connected graph is output (e.g. [Abney, 1996] [Basili et al., 2000a]).

Finally, robust parsers are usually based on complex parsing architectures where pools of modules are cooperatively applied to the source sentences. They add information (e.g. syntactic labels after POS tagging or syntactic dependencies after lexicalised analysis as in [Grinberg et al., 1996]) or, in other cases, they prune and disambiguate over redundant representations (e.g. PP attachment disambiguation over parse forests).

The cooperation among modules usually relies on strategies like "*disambiguate as late as possible*", so that ultimate choices are made only when useful information (syntactic and semantic) is available.

When under-specified representations are adopted it is easier to allow different components to add evidence *incrementally* until disambiguation can be triggered.

The results of such strategy are *modular approaches to parsing*. The parsing process is decomposed in subtasks organized in pools (i.e. cascades or pipelines of individual/independent components). Each module tries to maximally confine part of the overall ambiguity: an example relates to the NP boundaries that in chunking (e.g. [Abney, 1996]) are detected first. Whenever the modules are able to limit all sources of a given type of ambiguity as soon as possible, all the later phases inherit a simpler representation where more constraints can be applied to reduce complexity.

The adoption of modular approaches to parsing raises the problem of flexible parsing architectures as no specific (deterministic) architecture is good for all cases and domains. A modular architecture is useful if it can be configured according to a specific notion of robustness, well suited for the performance required in the target domain.

Modular parsing architectures require the definition of possibly reusable modules. More robust parser can be obtained by re-configurable architectures/systems. The result is that also design methodologies are important for robustness. A methodology for building re-configurable parsers is more useful if it allows controlling the degree of robustness since the design phases.

The above observations about robustness (sources, limits and their influence on parsing) emphasized the need for a more operational notion of it able to also influence its direct measurement.

4.2.1.1.2 Robustness in NL Parsing: the attempt of an empirical definition[6]

All the above observations lead us to the following main assumptions:

- Robustness can be hardly defined in a fully formal way, as it requires linguistic and deviant phenomena to be modelled.
- Robustness has characteristics related to endogenous and exogenous stress, or invalidity in the source input.
- Robustness has to deal with performance, and this latter is tightly related to a corpus/domain and to the application/task.

Any serious attempt to deeply analyse and model robustness cannot neglect from all the above assumptions. Although other approaches to robustness have relied upon psycholinguistic analysis of its counterpart in human parsers, we will attempt a more data-driven analysis based on empirical evidence and measures.

It should be in fact noticed that robustness is an important issue when large-scale analysis is undertaken. Any large corpus exhibits a "noise" (i.e. deviation from linguistic principles and also non linguistic phenomena) that determines lack in robustness in the underlying parsing system. It is evident how corpus phenomena are difficult to be captured by a given theory. They in fact are irremediably changing throughout different corpora and sub-languages. Although the source theory is by itself very robust (as it is modelled on a subset of human language phenomena that are its final scope), corpora tend to significantly disclose from it. This tends to replicate whenever new corpora are approached.

In order to determine a more usable notion of robustness we can thus try a corpus-centred notion of it. A theory T' is thus *more robust* than a theory T, iff small changes in the corpus , i.e. $\Delta(C)$, implies small changes in the results, i.e. $T'(C)$ or $T(C)$, i.e.

$$\frac{\Delta T'(C)}{\Delta C} < \frac{\Delta T(C)}{\Delta C} \qquad (1)$$

where ΔC roughly represents changes from a corpus C to a corpus C'. Notice how the above definition tries to capture the notion of graceful degradation. $T(C)$ here implies some notion of performance of the theory with respect to a data set. Although performance is **strictly** related to the target task, as different applications may require optimisation of different phenomena, we will leave this issue not specified at the moment. It does not prevent our analysis from drawing further (and useful) consequences.

However, it is evident that $T(C)$ is measurable in large only if a NLP system S is employed: S embodies T in its lexicons, grammars and control rules. When we make reference to $T(C)$, we are dealing indeed with a different function, $S(T,C)$, expressing a system S that, according to the theory T, is applied to the corpus C. This has consequences on the definition (1). Robustness of the process $S(T,C)$ can be now rewritten as:

$$\frac{\Delta S(T',C)}{\Delta C} < \frac{\Delta S(T,C)}{\Delta C} \qquad (2)$$

where $S(T,C)$ models the performance allowed by S in the application of a given T to C.

Equation (2) is useful as it allows decoupling the theory from its application to the corpus. This emphasizes the role of T and S independently. The performance $S(T,C)$ strictly depends on:

[6] For more details, see [Basili and Zanzotto, 2002]

- The linguistic knowledge embedded in *T* that is its lexicons and rules, e.g. grammars
- The assumptions that *T* makes about the input and output representation. For example the output of a parser can range from a single (i.e. the best) tree to a parse forest or a redundant syntactic graph.
- The algorithmic assumptions implied by *S*. First of all, *S* can (or not) support a specific decomposition of the process in several linguistic levels. Second, it is very sensible to the adopted representation, where either single data structures (like charts) may serve all the process or independent representations are used by different subtasks.

In view of measuring and thus assessing a more precise notion of robustness we can now rely on the above three aspects: when a computational framework is available to design a system *S* able to include aspects of one (or more) linguistic theory(ies) *T* and to support large scale performance evaluation over different corpora, a (possibly) relative notion of robustness can be measured and exploited in view of target applications. Several systems *S* can be obtained via organizations of different architectures (e.g. cascades of different parsing modules). Different theories can be tested via tuning and adaptation of lexicons and grammars. Finally large-scale evaluation should be made available with respect to changes in the corpora or against some of their separate and independent subsets related to different syntactic aspects or built according to different complexity.

Other purpose is to define :

- A *parsing framework* for design of systems *S* that support the application of different theories *T* (without major revisions). Notice that this does not reduce to defining a general formalism (or generalizing existing ones). Existing formalisms (e.g. feature structures as in HPSG) have been often criticized as they can be weak with respect to robustness: in [Menzel, 1995] the tight integration among syntactic and semantic language levels in HPSG is seen as a potential source of complexity for robustness. The required full constraint satisfaction (at syntactic and semantic level) can even prevent a suitable management of problematic situations where more flexibility is mandatory. Moreover, formalisms are often divergent and a single unifying formalism is not available. Trends in several NLP application areas (e.g. IE as in [MUC 1995] [Pazienza, 1997] suggest that heterogeneous architectures can be often successfully defined.

  The definition of such a framework is instead mostly related to the design phase of the target NLP system *S*, where software infrastructures play a major role.
- A *unifying representation of grammatical information* for the target systems *S* able to transparently support the intermediate parsing phases.
- A suitable notion of *performance* by which *S*(*T*,*C*) can be modelled. This notion will allow the systematic assessment of robustness in which applications will act as *Turing-like* tests.

The following sections will present, first, the principles underlying the required framework (section 4.2.1.2.1), then some parsing architectures reflecting the framework, i.e. modular and lexicalised parsers (section 4.2.1.3.1).

4.2.1.2 A modular, possibly pipelined, and lexicalised architecture for robust natural language parsing

The parsing design methodology should allow the production of systems that can be easily configured in order to achieve the desired degree of robustness. The proposed design methodology is based on two principles: the one inherited from the engineering practice, i.e. the *modularisation*, and the other more proper of the AI field, i.e. the availability of *"self"-adaptable components*. Modularisation imposes a clear separation between the activities performed by each module with evident benefits on reusability (modules are loosely coupled among them). Furthermore, the attention to "self"-adaptable components goes in the same direction. Knowledge-based approaches require an intensive work for tuning the general-purpose tool to the particular application environment (modules are loosely coupled with the knowledge domain).

4.2.1.2.1 Modular approaches: robust redundant voting policies vs. computationally attractive cascades

In the software engineering practice, *modularisation* is suggested as a method for the production of easy-to-reuse pieces of systems, i.e. the modules. In order to be re-usable, these modules have to be characterized by

high internal *cohesion* and loose *coupling*. Modularisation speeding up the initial system construction allows concentrating the efforts in fine-tuning the system to the particular application scenario.

Once the modularisation is accepted as an added value of the design approach, the next step is deciding which is the desirable composition of the modules for the tasks that the overall systems are designed for. In syntactic parsing system study, different approaches have been proposed for combining modules together: parallel vs. pipelined combining methods have been adopted.

Again from the engineering practice, *redundancy* is a well-known method against system failures. Hence, an increase on the "degree" of system robustness can be obtained duplicating the modules devoted to a particular task. This principle has been applied also to syntactic parsing in [Worm and Rupp, 1998]. In [Worm and Rupp, 1998], syntactic processors implementing different theories/models independently produce competing interpretations of the sentence. A chart based uniform representation is envisaged and a voting mechanism is applied to decide either which interpretation should be chosen or to combine different partial analysis. The "competing" parsers differ from the point of view of the information they produce over the input sentence: they range from *deep* parsers based on HPSG formalisms to shallow parsers based on finite-state cascades or HMM rules. A "prefer-the-deeper-analysis-whenever-available" is adopted and sentence interpretations are obtained mixing together partial interpretations. *The combined parser is "robust" in the sense that a reasonable (eventually degraded) response is (generally) produced.*

Pipelined approaches (i.e. module cascades) have the disadvantage/advantage to be more deterministic. Processing redundancies are avoided and finite-state-automaton cascades generally adopted [Hobbs et al., 1996] and [At-Mokhtar and Chanod, 1997]. In the perspective of real world applications where time constraints are important, they result to be more appealing since their computation time is inherently lower with respect to redundant approaches. Moreover, the integration of different approaches in a cascade-fashion is postulated in [Abney, 1996], [Collins, 1996]. In [Collins, 1996], a stochastic approach is applied over symbolically processed textual material: an intermediate level of phrase interpretation is adopted (i.e. the NP kernels). This work suggests the possibility to integrate symbolic and sub-symbolic approaches. The same mixture exists in [Carroll and Briscoe, 1998] where sub-categorization frames and statistical parsing approach has been positively integrated.

The computational appealing and the suggested possibility of plugging modules inspired by different theories in the processing chain are nice features of pipeline approaches that can be capitalized in our robust methodology for building up re-configurable syntactic parsers.

The modularisation we want to push here is fine-grained: the components are responsible of the detection of some syntactic phenomenon and are interested on a syntactic representation of the sentence that disburdens their analysis. The observations are translated in requirements for the formalism that has to transfer the syntactic analysis among modules. The unifying formalism must exhibit the possibility of *data encapsulation* and *partial analysis storage*.

4.2.1.2.2 Grammars, Lexicons and "self"-adaptable components

Modularisation design principles (high cohesion and loose coupling) by themselves do not guarantee that the "linguistic" modules are conceived to be reusable in a given operational environment (i.e. sub-language/domain). It is also a wide shared perception that some shallower syntactic material can be produced with rules independent from the domain (for instance the NP-chunking). These latter approaches can lead to the definition of modules that have a low degradation of the performances when exposed to the new working conditions. However, it is a well-known limitation that, in order to obtain accurate syntactic parsers, current methodologies propose domain dependent approaches. The domain dependent resultant parsers are generally based on a wide knowledge of the domain. The challenge in this field is to propose approaches able to learn selective rules with the minimal supervision. This is undoubtedly a positive aspect in the perspective of speeding up the tuning to an operational scenario.

The knowledge based approaches need information on the given application domain in the form of distributional frequencies of linguistic phenomena [Collins, 1996] or lexicalised rules [Pollard and Sag, 1994]. Generally statistical approaches are supervised: prediction rules are estimated on syntactically annotated corpora (the Penn Treebank [Marcus et al., 1993], the Susanne corpus [Sampson, 1993], etc.). These latter are expensive extensional representations of the grammatical intuitions of the annotators over a large amount of textual material. On the other hand, lexicalised approaches are based on precise intuitions of the grammar writers inspired by "real" corpus textual material. Both the approaches provide high

performance products. However, the tuning effort is high since, from the one side, portions of the corpora have to by annotated and, on the other side, grammatical rules have to be hand-written.

Knowledge based approaches are applicable in this framework if the required information can be learnt automatically with a *low* level of human supervision. Therefore, processors based on simple syntactical lexicalised sub-categorization frames (e.g. the verb lemma and the prepositions of the arguments) result to be applicable. In fact, this kind of "unpretentious" information is learnable with unsupervised algorithms [Brent, 1993] [Basili et al., 1997].

In the framework we propose that modules like:

- shallow analysers that take decisions over simple and domain independent phenomena
- lexicalised analysers based on syntactic sub-categorization frames and coupled with a weakly supervised learning modules

are more attractive since loosely coupled with the domain. They speed up the production of the system and the satisfaction of the performance (robustness) criteria.

4.2.1.3 Parsing Engineering in the practice: CHAOS, a pool of syntactic processors

The robust methodology for producing syntactic parsers proposed in the previous section foresees:

- the decomposition of the parsing process in (possibly) pipelined activities characterized by *high cohesion* and *low coupling*
- the definition of a uniform formalism supporting data exchange in the fine-grained decomposition
- the setting up of modules characterized by a low coupling with the application domain

In this section, we propose a case study where the above principles are applied in the production of CHAOS, a pool of syntactic parsing modules, which has been used in real applications (as text classification in TREVI [Basili et al., 1998c] and hyper-textual linking in NAMIC [Basili et al., 2001]) throughout different domains (finance, sport, medicine, etc.) and different languages (English and Italian).

4.2.1.3.1 Decomposition principles in CHAOS

The decomposition of a syntactic parsing process into different modules has to be motivated by the effective possibility of identifying sub-components with high degree of *internal cohesion* and a loose degree of *coupling*. The wide shared assumption that *verbs* control the semantics of the sentence and, thus, their syntactic projections constrain the overall syntactic interpretation can be an interesting inspiring principle for the modularisation. For instance, in the sentence extracted from an economical newspaper article:

The executives and the employees say the Acme company, whose revenues plunged to $783 million for the quarter ended Dec. 31, 2000 from $1.67 billion for the comparable period in 1999, is furiously trying to cut costs.

the role of the verb *plunge* is central in the sub-sentence. If the sub-categorization frame

```
(plunge, (Subj) (PP:from) (PP:to) )
```
[7]

related to the particular realisation were available, interpretations connecting together for example *for the quarter ended Dec. 31, 2000 from $1.67 billion* in a single prepositional phrase are obviously inadmissible. Since verbs play a key role in producing the correct interpretation of the sentence, a module devoted to this kind of phenomena is very appreciated. In fact, during the design activity it allows controlling the performances and thus the satisfaction of the constraints.

If this processor is available, the *loose coupling* principle imposes a first decomposition between processors devoted to the detection of phenomena influenced by the verb syntactic projections and those that are not. Then, since a pipeline is gracefully imposed, it should be decided what should be usefully done before the verb attachment detection and what should be done after. An interesting intermediate level

[7]The represented grammatical realization of *plunge* expects a subject, **(Subj)**, and two prepositional phrases, one with the preposition *from*, **(PP:from)**, and the other with the preposition *to*, **(PP:to)**.

between the words and the sentences is the notion of *chunk* [Abney, 1996]. Chunks are both psycho-linguistically motivated and computationally attractive. These are generally *phrase kernels*, as NPs [Collins, 1996], whose boundaries can be detected via finite state automata. Furthermore, meaningful portions of these chunks, i.e. their syntactic heads and their potential governors, are emphasized. In our case, chunks are also the sentence fragments for which the spans are not influenced by any verbal syntactic projection. The activity of chunk detection should be done before the verb argument detection, since the knowledge gathered in the chunking phase obviously disburdens the verb argument detection. As already stated, the notion of phrase kernels has been used also in stochastic approach to parsing as in [Collins, 1996. This does not limit the *integrability* between symbolic and sub-symbolic modules. Moreover, the same consideration applies for the verb sub-categorization based module since, as argued in [Carroll and Briscoe, 1998], it does not prevent the effective integration of statistical processors. The un-retrieved verbal argument as well as the NP-modifier detection will have a clear benefit if done after the verb argument detection. The search spaces of the later processors are constrained by relations drawn by the verb argument matcher.

The inspiration principle for the design of the module pool is then the concern of using in the best way the disambiguating power of the verb sub-categorization frames. The module competences are partitioned accordingly and their positions in the pipeline chain are then derived.

4.2.1.3.2 An unifying formalism: XDG

The proposed fine-grained modularisation of syntactic parsing requires a uniform formalism able either to represent partial analysis flowing between the modules or to show to the (eventually pipelined) modules only the information relevant for the single steps. In fact, processors dealing with the verb argument detection as well as the pp-attachment problem are interested to be exposed to the input as a chain of *VP*, *NP*, and *PP*-kernels where relevant features as the phrase heads and the prepositions of the *PP*s are highlighted. This nice property is owned by constituency-based syntactic representation scheme as the one inspiring the *charts* underlying the VIT formalism [Worm and Rupp, 1998]. In the software engineering, this *information-hiding* attitude is referred as *data encapsulation*.

However, the constituency-based approach has a limitation: the traditional notion of constituent as a subsequence of words in the analysed sentence. This limits its application in a fine-grained modularised framework. For instance, a pp-attachment resolution module should be free to draw the conclusion that a *PP*-kernel is the *VP*-kernel without postulating the structure of the rests of *NPs*/*PPs* between the two. A dependency-based annotation scheme [Tesniere, 1959], [Grinberg et al., 1996] is more indicated to cope with this kind of problem, but it is not well-suited for information hiding: the nodes of the graph are always words, no encapsulation of the information is foreseen. As an instance, the pp-attachment module has to navigate the structure in order to extract the key information to perform its choices (the preposition and the noun head of the PP-kernel as required by the pp-attachment resolution algorithm presented in [Brill and Resnik, 1994][Retnaparkhi and Roukos, 1994]).

The formalism we have defined is a mixture inheriting the positive aspects of the two (apparently diverging) approaches: the *data encapsulation* and the *partial analysis storage attitude*. The proposed annotation scheme is an extended dependency graph (XDG). It is a dependency graph whose nodes *C* are *constituents* and whose edges *D* are the *grammatical relations* among the constituents, i.e.

$$XDG=(C,D)$$

The *XDG* set is completely defined when the node tags, Γ, and the edge tags, Δ, are fully specified, i.e. it will be denoted by $XDG_{\Gamma\Delta}$. The Γ and Δ tag sets depend upon the level of the syntactic analysis (and the underlying grammatical theory).

The XDG formalism efficiently models the syntactic ambiguity. In general, alternative interpretations for dependencies are represented by alternative $d \in D$. A useful property can be imposed on *xdg*s to select a single (partial) syntactic interpretation. A *planar xdg* is a single (although possibly partial) syntactic reading. *Planarity* [Grinberg et al., 1996] interdicts *crossing links*, thus is can be used to select unambiguous sentence fragments. An unambiguous partial interpretation is any planar sub-graph of an *xdg*.

4.2.1.3.3 The module pool

A module *P* of the modular syntactic parser is a processor that, using a specific set of rules *R*, adds syntactic information to the representation of the sentence, i.e.

$$P : R \times XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma'\Delta'} \qquad (3)$$

so that $P(r,xdg)=xdg'$, where *xdg* and *xdg'* are the input and the enhanced graph, respectively. This implies that syntactic processors *SP*s are modelled as functions over *XDG*s, and their nature is reflected by properties of those functions. As any P_i module foresees the use of its own rule bases (elements in R_i), the first argument of a function P_i can be omitted for sake of synthesis, so that hereafter equation *3* will be written as

$$P_i : XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma'\Delta'}$$

With $P_i(xdg)= P(xdg; r_i)=xdg'$.

Actions that a module *P* perform on the *XDG* can be *monotonic* or *non-monotonic*. *Monotonic modules* preserve all the choices (i.e. nodes and arcs, as constituents and dependencies already recognized) expressed by the input graph.

Furthermore, with respect to the input *XDG*, the ability of a module *P* refer to:

- *constituent gathering*, for processors grouping set of words into larger constituents;
- *dependency gathering*, where nodes are left untouched and only dependencies are added.

Finally, a further distinction can be done with respect to the parameter R_i of each processor P_i. P_i is a *lexicon-driven* processor when R_i is lexicalised (e.g. a verb sub-categorization lexicon, r_i). P_j is a *grammar-driven* processor when R_j does not include any lexicalised form of syntactic information (e.g. categorical or PSG rules).

According to the above definition several processors can be defined. An overall modular parser *MP* is thus defined as a cascade of processing modules $(P_1 ,\ldots\ldots,P_n)$ via composition of processors:

$$MP : XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma'\Delta'}$$

with

$$MP(xdg) = P_n \circ P_{n-1} \circ \ldots\ldots P_2 \circ P_1(xdg)$$

The modules actually used in CHAOS are described in the next sections.

4.2.1.3.3.1 Grammar-driven components

In order to be loosely coupled with the special language, the grammar-driven components should have general (and possibly under-specified) rules. Decisions will be taken by modules having high-expectations on the behaviour of the words (i.e. the lexicalised components). The two grammar-driven components adopted in the CHAOS pool are: (1) a chunker [Abney, 1996] and (2) a shallow syntactic analyser [Basili et al., 1992].

The chunker is the component that has to pack ambiguity independent from verb valence information. It, thus, provides a set of (possibly) complex sentence fragments as kernels of nominal phrases NPK (e.g. *[The executives] and [the employees] say* ...) or prepositional modifiers PPK (e.g. ... *plunged [to $783 million] [for the quarter]* ...). Basic information related to a chunk is a syntactic category (e.g. NPK, PPK, etc.), a potential governor and a grammatical handler possibly different from the governor. It recalls quite closely the notion of instance of morpho-syntactic *template* in most dependency based parser. In terms of the *XDG* notion introduced above, a chunking process matching grammatical rules (the *chunk prototypes*) over a part-of-speech tagged sentence (Γ'={Verb, Noun, Preposition, Adjective ...}) and produces an *xdg* whose nodes are chunks, characterized by a governor, and the syntactic category (Γ={VPK, NPK, PPK, ...}), i.e.:

$$Chunker: XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma'\Delta'}$$

It is a machine computationally complex as a finite-state automaton since it is possible to express the chunk prototypes as regular expressions. The coupling with the domain is small since it postulates and uses only prototypical descriptions of simple structures.

The shallow syntactic analyser aims to draw relations among the chunks without using deep information (sub-categorization lexicons). The grammatical recognition is based on a shallow parsing strategy presented in [Basili et al., 1992]. A discontinuous logic grammar formalism is here used to model matching of non-adjacent (i.e. expressed by gaps) modifiers and *specifiers*. Logical patterns as feature structures are used to express legal realizations of constituents with gaps: *skip* rules are used to express sentence fragments among head and dependents. Such fragments are simply skipped by the parser and left unanalysed although logical constraints (via unification) are imposed to their feature description. The result of the analysis is an *XDG* enriched with potentially ambiguous grammatical relations. The ambiguity is modelled via a *plausibility* score. In term of the formalism introduced, the shallow dependency parser is:

$$SSA: XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma\Delta}$$

where $XDG_{\Gamma\Delta}$ have chunks as nodes (Γ and Δ are Γ={VPK, NPK, PPK, ...} and Δ={SUBJ, DIROBJ, PPMOD, ...}).

4.2.1.3.3.2 Self-adaptable components

The precision of the whole syntactic parsing can be controlled if high expectations on the word behaviour are postulated. This is generally obtained by using lexicalised rules, i.e. rules activated by particular lexical items. Many of these rules depend tightly on the domain since they capture word meanings. This is particularly true for verbs. For instance, the verb *operate*, in the medical sub-language, can have the meaning of *"perform a surgery on"* and, thus, has the sub-categorization structure `(operate, (SUBJ,PP:on))`. In the finance sub-language it is likely to express the meaning of *"operate in a market sector"* and, consequently, the preferred reading is provided by the frame `(operate, (SUBJ,PP:in))`. This difference can result in very high performance variation. Furthermore, it is important to activate the subpart of the lexicon that can provide improvements in the particular domain. Modules based on sub-categorization lexicons are valuable in this framework if underlying lexicons are re-configurable and tuneable to the particular sub-language. In [Basili et al., 1997], [Basili et al., 1999], the possibility of acquiring this form of knowledge has been demonstrated to be effective in a shallow parsing environment.

Therefore, a specific processor, the Verb Argument Syntactic Matcher,

$$VASM: XDG_{\Gamma\Delta} \rightarrow XDG_{\Gamma\Delta}$$

is adopted in the pool. It matches verb argument structures and organizes the detected phrase fragments into a hierarchy of clauses. *VASM* is a lexicalised processor able to work at different levels of lexicalisation that processes $XDG_{\Gamma\Delta}$ whose nodes are chunks (Γ and Δ are Γ={VPK, NPK, PPK, ...} and Δ={SUBJ, DIROBJ, PPMOD, ...}). Successful matches add to the target *xdg* dependency arcs also called *icd*s, i.e. *inter-chunk dependencies*. An original feature is the specific combination of the argument matching with the clause recognition. As sentences have more than one verb defining different sentence clauses, the matching of argumental *icd*s also determines the set of detected clause boundaries. In this perspective, coordination and subordination between clauses are approached on the basis of verb argument recognition. The recognition of the complete hierarchy of the sentence clauses is refined incrementally along with the matching of argumental *icd*s for the different verbs ([Basili et al., 1998a] for technical details). In *VASM*, the role of lexical information is not only to fill slots of lexical entries, but also to control, via planarity constraints, the matching for other verbs and the activity of the grammar-driven modules. The kind of suggested analysis has been also adopted for Italian where the relatively free order of arguments in sentence often require control rules to judge among competing slot fillers.

The use of sub-categorization frames introduces a *graceful* correlation among the syntax and the semantics analysis as sub-categorization frames *shallowly* convey the semantics of the words (verbs, in this case). The integration of the two level of analysis allows the propagation of semantic constraints in the later phases of the process of syntactic analysis.

4.2.2 Robust semantic analysis

Semantic Web (SW) [Lee, 2001] tends to push the successful information access paradigm where a *slowly changing* knowledge, i.e. a conceptual level, is used to organise a target world. Final users will expose and gather information at the instance level seen with the glasses of the slowly changing (and shared) conceptual level they can contribute to build. The conceptual level is generally used to organise the knowledge and it is hardly queried as represents static information.

The viability of this approach is suggested by the pervasive presence of Data Base Management Systems as a successful solution for accessing information. They are based on the same principle: a conceptual level is used to organise an instance level that is hardly queried. Moreover, the fact that the crucial decision of the partition between what stays in the conceptual level and what in the instance level has not limited their applicability in large as partial solutions may be established within the final application domain.

Some NLP applications as Information Extraction offer facilities to retrieve information stored in a document using a formal conceptual language there called *template* (see Sec.2.1.2). The equivalence between Information Extraction and the above approaches is so direct that Information Extraction systems are often explained through the metaphor of being systems able to feed database with unstructured documents written in natural language, i.e. documents where no structure has been imposed to the actual writer. *Templates* are used to model relationships among typical instances of the target domain. These instances are generally referred to as *Named Entities* (e.g. *Companies*, *People*, *Dates*, etc.). For example, the target of the exercise in the MUC-7 competition [MUC, 1997] was to satisfy the *managing succession* information need. This lead to the definition of a template like *People left/joined Company in Date* where *Person*, *Company*, and *Date* should be instantiated according to the specific information gathered by the analysed document. Moreover, also *factoid* questions used to test open domain Question-Answering systems generally ask specific relationships among what can be perceived as an instance, e.g. *What country is know as the "Land of the Rising Sun"?* , *What company manufactures Sinemet?* The definition of what is the conceptual level is here not necessarily foreseen.

Assuming the existence of a defined conceptual level that gives a formal language for the knowledge in the application domain, Semantic Web has the problem of bridging the gap between instances as they are represented *in nature* and the formal language. It is generally the case that final users express themselves through a "form" of natural language. For instance, in a university domain the sentence *Prof. Brown teaches the database course* may represent in a document the intended state-of-affairs. The same information could be conveyed by the relationship `teacherOf(Faculty_Member,Course)` written in a (semi-)formal language stating that a `Faculty_Member` may be `teacherOf` a `Course`. In case it is the formal expression for the relationship, in the SW vision the document expressing this information should be exposed together with the related *metadata*, i.e. `teacherOf(Faculty_Member(Brown),Course(database))`[8] .

The automatic mapping between natural language sentences and an internal (ontological) language is the target of semantic processing models for natural language. Information Extraction systems are shallow examples of these models where this activity is often referred as *template filling*. Templates are often very specialised and in term of a SW ontology they may represented in one or two relationships among concepts. Then, in order to be useful for the "knowledge scale" envisioned in the Semantic Web, IE systems have to demonstrate to be scalable taking profit of the ontologies produced in the SW collaborative work. These "interconnected" ontologies may represent domain specific knowledge within these natural language interpretation systems as WordNet [Miller, 1995] is a de-facto standard for representing the general linguistic knowledge of an English speaker. It is possible to imagine that the quality of these resources, whenever real needs will drive their production, will be high. In spite of the freedom allowed in the production of new conceptualisations, it is reasonable to expect that a first knowledge representation jungle will leave room to a more orderly place where only the more appreciated conceptualisations will survive in the practice of representing web documents.

[8]The simplified version here used does not take into account the `hasTopic` and `hasName` relationships.

Methods for semantically analysing natural language may be then very useful in the SW and, conversely, SW may help in the production of such kind of models. Notice that the capabilities required to these semantic models have a limited extent as it is assumed that, at the processing time, the ontology remains unchanged. Concepts and relationships are not added. This behaves mostly as an "empty shell" whose instance level is fed by analysed documents (e.g. the sample pages in Fig. 10) or natural language users' interactions (an example is given in Fig. 11).

Database Theory and Practice, *II semester 2004*

Prof. John Brown

Teacher's lecture notes

- First lecture: Introduction to Database
- Second lecture: … …

Fig. 10 A sample of the Web Page Natural Language

(a) *Who teaches the database course?*
(a') System Answer: *John Brown*
(b) *Where do I find the teacher's lecture notes?*
(b') System Answer: *In his home page*
(c) *Is he an employee of this university?*
(c') System Answer: *Yes*

Fig. 11 A sample dialogue in the restricted interaction

Interpreting natural language sentences under these restricted conditions seems to be feasible as MUC conferences' results demonstrate [MUC, 1997]. The interpretation under these conditions asks for specific capabilities as:

- recognising what is perceived as instances of ontological concepts, i.e. *named entities* that are textual elements having a clear referent in the world (e.g. *John Brown* and *database course*)
- recognising nominal phrases in the document that are referring to specific entities in the world and correctly resolving anaphorical links. For instance, the natural language expression *teacher* in both examples is referring to a specific entity in the world, i.e. *John Brown*.
- populating the ontology with instances coming from the two previous analysis
- recognising active ontological relationships among specific instances (e.g. the relationship `teacherOf(Faculty,Course)` active both in Figs. Fig. 10 and Fig. 11)

In the rest of this section, we will analyse how SW ontologies can be used to support a language interpretation model working under the above conditions. We will firstly describe the limits of an ontology thought in the formalisms as OWL will have whenever used to drive linguistic analysis. Secondly, we will focus on a model for carrying out semantic processing of natural language. This will help us in describing the difficulties that can arise in using domain concept hierarchies in NLP. As a result of this analysis the need of a mapping between domain concept hierarchies (DCHs) and general purpose linguistic knowledge bases (LKBs) will become clearer.

4.2.2.1 Towards "Linguistic Interfaces" to Domain Ontologies

Ontologies for the Semantic Web are generally written in formal languages (OWL, DAML+OIL, SHOE) that are generalisations/restrictions of Description Logics [Baader et al., 2003]. Concepts and relationships are described in TBox assertions. As the collaborative work induced by the SW initiative will force the construction of ontologies in these languages, their expressivity is a crucial aspect.

For a concept a typical entry models its *identifier* (a *label* representing a natural language expression describing the "meaning" of the concept) and a link indicating the fact that the described concept is a specialisation of an other concept in the network (i.e. *Subclassof*). For example, the concept of *course* is represented in a university domain as:

ID:	Course
Course:	Course
Subclassof:	Work

As it emerges from the example, the label has the only purpose of highlighting the concept for human readers. The formalism does not force to indicate the alternative linguistic expressions such as *class* in the example. The usefulness of this information will emerge in the next section when we will describe the model for semantic analysis of natural language.

Such a limit on the expressivity is even more obvious when considering relationships. Relationships are typically binary and are described in structures containing an identifier, a label, the two participants called *domain* and *range*. An example is:

ID:	TeacherOf
Label:	Teaches
Domain:	#Faculty
Range:	#Course

that describe the relation between teachers and courses. In this case, the information contained in the label and in the identifier is fairly useful to interpret natural language expressions like: *Prof. Brown teaches the database course* whenever *Prof. Brown* and *database course* are respectively recognised as (maybe indirect) instances of the classes `#Faculty` and the `#Course`. However, this kind of lexical information is not sufficient for correctly interpreting alternative sentences as *Prof. Brown delivers the course on linguistics*, *Prof. Brown gives the course on linguistics*, or *Brown is the professor of the linguistics course*).

Such one-to-many correspondences between ontological concepts and relationships on the one side and linguistic forms is what can be called the "linguistic interface" of an ontology.

Ontology producers are not forced to indicate this "linguistic interface" and labels alone are not enough for this purpose. Synonymy may be seen as a irrelevant and dangerous phenomenon for ontological representations. It may be considered a possible generator of unnecessary concept name clashes, i.e. ambiguities for concept names. Nevertheless, this "linguistic interface" to domain ontologies constitutes the basis where to build the semantic model of the natural language processing system on.

4.2.2.2 Semantic interpretation through "Linguistic Interfaces"

Some systematic attempts to model the relationship between the domain (expressed with an ontological representation) and natural language have been carried out in the restricted scope of information extraction systems where knowledge-based models have been implemented. These result in shallow semantic interpretation models for natural languages. A very interesting example is LaSIE [Humphreys et al., 1998]. In the following we will use this system (i.e. its knowledge representation language [Gaizauskas and Humphreys, 1996] and its processing steps) to clarify the activity of a semantic interpreter in the "restricted" environment of interpreting "grounded" natural language expressions. Unlike more shallow IE systems (e.g. [Hobbs et al., 1996]), a knowledge-based model make an explicit use of declarative description of the world, often called *World Model*. The world model (WM), an other way to refer to an *ontology*, is a formal representation of concepts and relationships, here called *objects* and *events*, for a particular domain or set of domains. Notice that there is not necessarily a one-to-one relation between *events* and *domain relationships* as the first are mainly used to organise lexical knowledge for capturing relevant relationships among concepts in the domain.

As in any ontological model, concepts are classes used to organise "referable" entities that, in this case, are *individuals* in the target domain. Generally they are called *named entities* to indicate that these are individuals in the world having a linguistic expression able to denote them. Classically (i.e. in [MUC]) the more relevant named entity classes are *People, Places, Organizations* but these may also be better organised for more specific domains. For instance, a possible excerpt of concept hierarchy in a university domain is represented in Fig. 12 under the node `object`. Events are generally represented in separated portion of the hierarchy. This is a clear difference with the organisation of the relationships in OWL. Events are used to organise linguistic expressions denoting some specific domain relationship. Their organisation in a hierarchy is mainly thought for this purpose. This will be clearer in the followings.

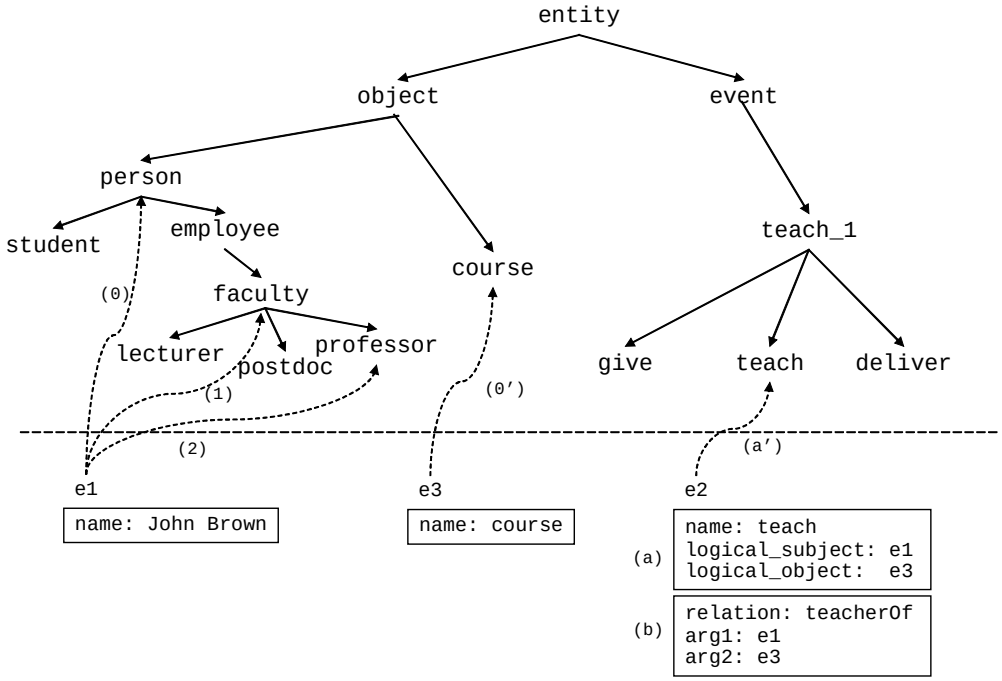


Fig. 12 An example of language interpretation in the university domain

The basic idea underlying the definition of a world model to support reasoning with texts is that both concept and relationship instances are represented through word (or governing word for the events) instances in the interpreted text. This mainly follows the intuition, in line with linguistic theories such as [Montague, 1974], that words are divided into passive arguments when they behave as concepts and active functors when they behave as relational concepts. Often, the first role is played by nouns while the second by verbs or *deverbal* nouns.

The "linguistic interface" can be then realised in two ways. On the one hand, actual denotators for concepts and events model a first linguistic interface. Concept names are used to indicate words that, within texts, may play the role of anaphoric expressions, i.e. *teacher* in Fig. 10 and Fig. 11. Event names may be used to indicate words possibly governing relationships. On the other hand, syntactic-semantic mapping rules complete the description of the linguistic interface for the events. Event names are used to activate a possible relationship as for instance *teach* in case of the relation `teacherOf(Faculty,Course)` but the real presence of that relation depends on the availability of the two participants in the surroundings. Typically, the relation is completely active if the verb *teach* has an instance of the concept `Faculty` (as logical subject) and an instance of `Course` (as logical object). The description of this level of the linguistic interface has to be done by means of active *properties* as these have to be used during the text interpretation process. The event related to the word *teach* will be transformed in an instance of the relationship only if all the constraints on the sentence are satisfied. These constraints indicate how words together with their selectional preferences may activate the related portion of an internal language. The language for expressing the selectional preferences of words have to allow the expression of grammatical and conceptual constraints. The former are restrictions over the possible grammatical structures that have to be found in texts. The latter are constraints

imposed to the conceptual nature of the grammatical arguments. In the knowledge representation language adopted in LaSIE (an extension of XI [Gaizauskas and Humphreys, 1996]) these latter mapping rules are represented through two special properties, i.e. *presupposition* and *consequence*. These two active properties are used at a different level of the text interpretation process. A typical syntactic-semantic mapping rule may be represented by the following sample that expresses the constraints for the *teach* event. First of all, `faculty` and `course` can be assumed to populate the *object* hierarchy. A specific "*event*" type `teach_1()` will express one of the linguistic expressions required to model the `teacherOf()` association in a university domain via the following `consequence`:

```
props(teach\_1(E), [
     (consequence([relation(E,teacherOf),arg1(E,X),arg2(E,Y)]) :-
                    X instanceOf faculty(\_),
                    logical\_subject(E,X),
                    Y instanceOf course(\_),
                   logical\_object(E,Y)
   ]).
```

The concept `teach_1()` is a specialization of `event()`. Among its properties (referred to as `props`) the above consequence states that whenever all the constraints are satisfied an instance of the relationship *teacherOf* is found. Constraints stated in the right hand aside of the (Prolog-like) rule ask that `X` and `Y` are respectively instances of the concepts `faculty` and `course`. Each of them is requested to stay in a specific grammatical relation with the verb *teach*. If all these constraints are satisfied the relational properties `relation(E,teacherOf)`, `arg1(E,X)`, and `arg2(E,Y)` can be inferred[9] . This rule is associated with the event `teach_1` having as possible governing word the verbs *teach*, *give*, and *deliver* (see Fig. 12 under the node `event`).

In this model, semantic interpretation is seen as the process of changing the world model according to the analysed text. As we assumed that the target language was limited, i.e. we were interested only in text representing instances of known classes, during the interpretation the pre-existing view of the world (i.e. the world model) does not change. New classes are not added. The interpretation is carried out at the instance level. Then, each word will be considered as a possible instance of one of the existing concepts. Given an empty discourse model *DM=∅*, each sentence *s* in the text *t* will activate the following steps of the interpretation process:

1. add concept and event instances suggested by *s* and by the use of the linguistic interface
2. draw all the presuppositions/consequences implied by the current state of the WM
3. look for co-referring elements
4. draw all the consequences implied by the current state of the WM

As an example, let us consider the analysis of the sentence:

Professor John Brown teaches the database course.

and let us assume that the syntactic analysis and the named entity tagging has been already carried out, i.e. *John Brown* is a nominal phrase representing `person` modified by the name *professor*, *the database course* is a nominal phrase governed by the word *course*, and *teaches* is a verb having as logical subject *John Brown* and as logical object *course*. In the first step of the interpretation, relevant words (i.e. nouns and verbs) of the sentence are seen as instances of internal concepts. These instances will be named with a unique identifier (e.g. the *John Brown* instance is represented by `e1`, *teach* by `e2`, and *course* by `e3`). These instances are linked to the related concepts using the first level of the linguistic interface, i.e. the concept names, and the named entity classes. In the example, the *John Brown* instance is attached to the node `person` using its named entity class, the verb *teach* instance to the `teach` concept node using the concept name, and the *course* instance to the node `course` for the same reason, respectively link `(0)`, link `(a')`, and link `(0')` in

[9]They will be added to the other known information about the concept (`E`) to be used later in the template-filling phase. In this phase only the *role* of each argument is important: it is here given by an index among the argument set (see the `arg/2` predicate).

Fig. 12. Additional properties may be added to `e2` as the verb instance establishes specific syntactic relations with `e1` and `e3` (see box `(a)` in Fig. 12). In the second step, some presuppositions or consequences can be derived. Active properties as rule can be used to make the hypothesis that `e1` may be specialised in `faculty` as that is the only constraint violated making the rule firing. Link `(1)` can be activated, see Fig. 12, and link `(1)` erased. Moreover, as `e1` has *professor* as modifier, a similar rule can be applied to make more specific the intuition on `e1`. Link `(2)` can be activated. Finally, as the third step may not be applied in this example, final consequences may be drawn in the last step of the analysis. This means that the active properties may generate as stable all the inferable properties. In the example, property box `(b)` may be finally added. The knowledge about the referred entities has been then changed as a consequence of the reasoning applied using the "active" properties in the world model and the sentence may be interpreted according to the internal language, i.e. relation `teacherOf` between two entities is active at the end of the process.

4.3 Learning Domain Ontologies and “Linguistic Interfaces”

A further refinement of the system is the implementation of the mission ontology, as discussed in Sec.3.4 .In this section we describe our approach and methodologies to create domain ontologies (such the one concerning Spacecraft Design) using a textual corpus (Sec. 4.3.1). In Sec. 4.3.2, we propose a brief survey of other approaches developed in the AI community.

As discussed above, one of the important aspect to push the semantic web vision forward is to demonstrate an ability in helping the writers of the www sites in expressing the information they want to expose in the formalised ontological language if necessary. Moreover, as the information in a www site may not be modelled by the already expressed conceptualisation, a very relevant activity is the automatic production of a possible conceptualisation starting from the www site or some related texts.

Using the definition of the notion of the ontology given before, we may determine the different directions in which we can work. Given the www site (i.e a very specific collection of documents), the “learnable” objects are:

- **The interface level**. This may be learnt for two reasons: the first, to learn information extraction rules able to subsequently automatically fill the instance level; the second, to learn different ways in for expressing the same question
- **The conceptual level**. This may be learnt for adding concepts and relationships among concepts that are not foreseen in the previous conceptualisation

Learning the interface level may help in maintaining the consistency of a knowledge node and in including in the grid homogeneous sites. Learning the conceptual level will give the possibility of including heterogeneous knowledge nodes in the grid.

As already discussed, the information gathered in the annotation phase will be a very important test-bed over which the learning techniques can be explored. In the following we will describe some ideas on which we will work for addressing these issues.

4.3.1 Ontology Extraction from Plain Texts

We want here to propose an acquisition method for the derivation of the linguistic interface of an ontology able to suggest linguistic patterns for known concepts and relations as well as to propose new concepts and new relation types. Our perspective is thus slightly different from other works (e.g. [Riloff, 1996] [Yangarber et al., 2000]). Plain texts are the starting point of our analysis. Texts here are assumed to drive the discovery of new domain knowledge. Fig. 13 presents an overview of the overall process where a cascade of activities (in the horizontal arrow) is defined to produce the final ontology.

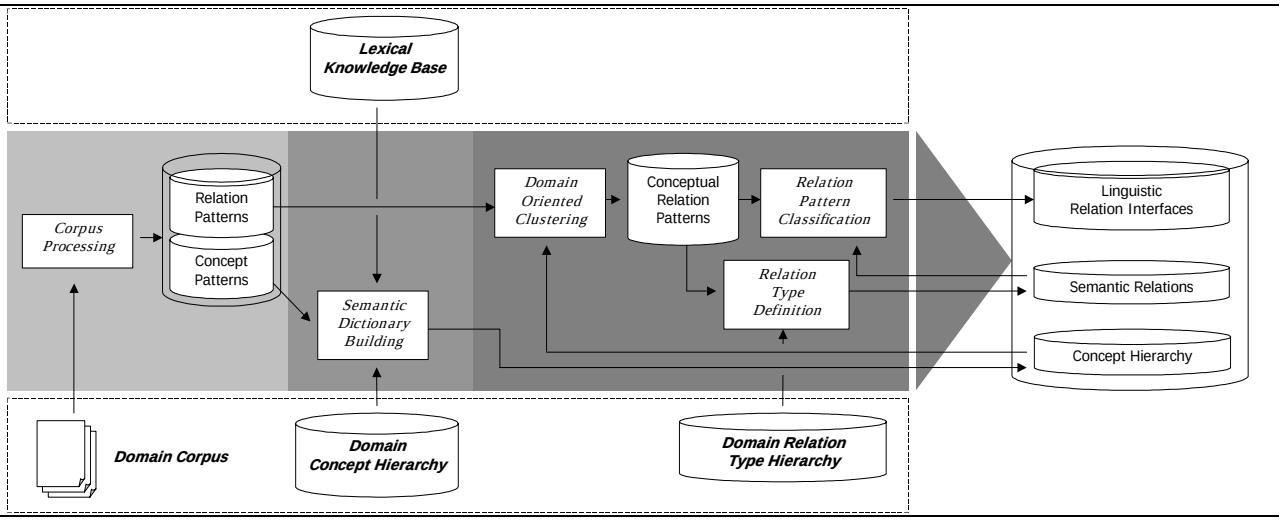


Fig. 13 Learning ontologies from free text: the architecture

The acquisition process makes use of previously existing domain knowledge. This foresees at least a domain concept hierarchy (DCH) and a relation type system (RTS). General-purpose linguistic knowledge is also required. It includes at least morphological and grammatical models as well as lexical semantic knowledge (such as the Wordnet lexical hierarchy). The overall process aims to harmonise the two above sources by exploiting a large-scale corpus. The result is an augmented version of the source concept hierarchy that include a variety of linguistic knowledge. The process should in fact:

- determine the relevant concepts that should be used in the target concept hierarchy DCH, possibly extending it;
- propose relevant relationship prototypes that are linguistic explanations for the relation prototypes postulated in the domain hierarchy RTS or that constitute newer prototypes;
- determine the linguistic forms in which the domain concepts are realized in texts;
- propose textual denotations of the relation prototypes, that is, the linguistic interface of domain relations/associations.

For the enterprise above, a terminological perspective is helpful. The main objective of research in terminology has been the extraction of synthetic representations for domain knowledge out from available material [Pearson, 1998]. Thesauri and technical vocabularies, i.e. the *explicit domain models*, are in fact built using domain text collections considered as *implicit domain models*. These simple assumptions characterize several approaches in the computational terminology area [Computerm, 1998]. At the beginning of the process only the text collection and a possibly pre-existing term list is made available. A very general definition of the notion of *term* is then exploited, i.e. a *surface form of a relevant domain concept* [Jacquemin, 1997]. The different approaches to Terminology Extraction (TE) tend to give an "operational" definition of term describing (see [Zanzotto, 2002]) the following aspects:

- the *admissible surface forms*. Admissible surface forms are usually described as prototypes over a valid natural language interpretation level (i.e. morphological, syntactical or semantic level).
- the *domain relevance*. Domain relevance is used as a decision function and it has been generally implemented via statistical measures over text collections: the frequency of the term surface forms or the mutual information are examples of such functions. The simple frequency in the corpus has been suggested as the more effective decision function among surface representations including the same number of content words ([Daille, 2004]).

The above model has been generally exploited for the description of the concepts in a domain while an attempt to use it for the detection of domain relationships has been done in [Basili et al, 2002]. It is worth noticing that in the model underlying TE systems, it is the domain corpus and not the information needs, which drives the extraction of domain knowledge.

In the learning of an ontology for Information Extraction, the principles of TE practice can be usefully adopted and in particular its notions of *admissible surface forms* and *domain relevance*. The first helps to characterize promising candidates for concept and relationship denotations. The notion of domain relevance optimises the work of ontology engineers: in fact, only the most relevant linguistic material (domain concepts as well as domain relations and their linguistic interface) will be shown. These are patterns sorted according to the domain relevance. Due to their inductive nature, unexplained patterns are potential candidates for new relations and are helpful even in the design of concepts and relations. Moreover, since linguistic constraints are used to express concepts and relationships they can be also used to cluster data. Sparse phenomena with a common semantic interpretation can thus be grouped: the higher is semantic agreement the higher will be the ranking of the underlying phenomena according to the domain relevance function. The result is that the target content of the linguistically principled ontology will grow faster as it integrates meaningful information emerging from the domain text collections.

The overall learning process (Fig. 13) is organized as follows. Firstly admissible surface forms are extracted from the corpus and promising concept and relation candidate are stored as patterns. This activity is referred to as *Corpus processing*. Then, an analysis devoted to determine a concept hierarchy is applied to the more relevant concepts extracted in the previous phase. It makes use also of the pre-existing domain concept hierarchy (DCH). This activity generalizes the available evidence across the general-purpose lexical knowledge base and is called hereafter as *Semantic Dictionary Building*. Its aim is mainly to map domain concepts into the general lexical database. The resulting concept hierarchy can in fact be used in the analysis and interpretation of relational patterns in the domain texts. This generalization allows to conceptually cluster the surface forms observed throughout the corpus. The derived generalizations can undergo the statistical processing during the *Domain Oriented Clustering* phase. Their distributional figures are derived and the resulting generalized patterns can be organised according to their domain relevance score. The *Relation Type Definition* and the *Relation Pattern Classification* are then manual phases. The first activity is targeted to the production of a set of *Semantic Relations* (*SR*) determining a system of domain specific relationship names, i.e. labels helpful in the interpretation/explanation of prototypical concept associations in the domain. The semantic relations type system determined in the previous phase is then used to classify the specific linguistic patterns clustered and ranked in the previous phases. The result of this last activity is the set of linguistic rules for the matching and prediction of relations in *SR*. We will call hereafter such rules as *Linguistic Relation Interfaces*.

4.3.1.1 Mapping lexical knowledge bases and domain concept hierarchies

The process of building a domain semantic dictionary aims to detect a suitable subset of semantic primitives able to represent promising and effective generalizations of linguistic expressions in the domain. The need for linguistically consistent knowledge requires the availability of language oriented "is_a" hierarchies to model and explain textual phenomena. Such a process seeks a suitable mapping between domain specific resources, i.e. a *domain concept hierarchy* (**DCH**), and domain-independent lexical knowledge base (**LKB**). The role of DCH can be played by topic taxonomies while an example of LKB could be the hyponomy/hyperonimy taxonomy in WordNet [Miller 1995]. Notice how this mapping deals with a general many-to-many correspondence between the DCH ontological primitives and the LKB word senses. For example, the word "*space*" has eight senses in WordNet while it could correspond to just one topic category in a spacecraft ontology.

As in the rest of the section we need to discuss concepts (in DCH), word senses (in LKB) and several significant implications, we introduce more formally a set of useful definitions. Any concept *C* in the domain hierarchy (DCH) is characterized by its linguistic label hereafter noted as t_C. This label *t* corresponds either to a singleton word or to a multiword expression. This information can be used as a reference within the lexical knowledge base *LKB*. We will denote LKB entries by means of Greek letters, e.g. α. Those LKB senses that correspond to possible linguistic meanings of label *t* will be denoted as α_t. Sometimes α_t may not

exist for technical concepts as they are not present in the domain independent LKB [10] . In general, a label *t* will correspond to more than one sense.

As DCH and LKB have both an internal structure some other useful properties can be introduced. First of all we will call *linguistic extension* of a DCH concept *C*, denoting it as *ext*(*C*) the set of the labels for *C* or for one of its descendants *C*' as follows:

$$ext(C) = \{t(C') | C subsumes C' according to DCH\} \quad (1)$$

For example the linguistic extension of *Spacecraft* in a mission ontology could include words and terms like "*rocket*", "*lunar spacecraft*".

Given its extension, a DCH concept *C* can be interpreted in LKB via its *linguistic generalization set*, that is the set of generalizations, α_t, in LKB for the labels $t \in ext(C)$. It will be denoted by *lgen*(*C*) that is defined as

$$\lg en(C) = \{\alpha \in LKB | \exists t \in ext(C) \wedge \alpha_t is subsumed by \alpha in LKB\} \quad (2)$$

Due to language ambiguity the generalization set *lgen*(*C*) includes more senses in LKB than those needed for represent *C* as each sense α_t for a given $t \in ext(C)$ is retained. In the next two sections the model to constraint generalizations in LKB by means of DCH information will be defined aiming to reduce the overall ambiguity and detect the correct LKB sense assignments(s) to DCH elements.

4.3.1.1.1 Inspiring principles

One of the aims of the proposed integration is to constraint the search for word sense assignment (i.e. navigation in the LKB) through information provided by the domain resource. Vice versa the LKB structure will be used to bias the search of DCH meanings, i.e. explain linguistically the nature of the DCH primitives.

Cross-corresponding concepts between a DCH and a lexical model LKB can be detected by exploiting in combination both constraints. We will rely on the following two principles:

(**P1**) (*Extensional Nature of DCH*). Given a domain concept hierarchy DCH, whatever the nature of its basic unit is, *subsumption throughout the hierarchy has always an extensional interpretation*, i.e. for each couple of concepts *C*' and *C*" subsumed by a common ancestor *C* in DCH, there is always a linguistically consistent concept $\alpha \in LKB$ such that the linguistic expressions $t'=t_{C'}$ and $t''=t_{C''}$ have senses $\alpha_{t'}$ and $\alpha_{t''}$ both subsumed by α in LKB .

(**P2**) (*Intentional strength in LKB*). A set of linguistic denotations $W=\{w_i\}$ [11] whose senses are all subsumed by a given $\alpha \in LKB$ has an *intentional strength* for *W* that is a function of the senses of w_i and of their distribution in the LKB sub-hierarchy dominated by α. α represents the trade-off between the generalization required to represent all the denotations w_i and their specialization, i.e. the capability of separating the individual different senses of the w_i's. Any monotonic non-decreasing function of such a trade-off is a valid measure of the intentional strength of α with respect to words w_i .

[10] Notice that in this case we could relax the search of the multiword expression, e.g. *Common Hepatic Duct*, and try to match senses for sub-expressions obtained by neglecting some modifier, e.g. *Hepatic Duct*. The longest expressions corresponding to one LKB entry would be retained as a possible linguistic interpretation.

[11] The use of w_i here emphasizes the difference with respect to the previously adopted notion of t_i. w_i are linguistic symbols that independently from any domain are referential in the world. t_i are terminological labels of DCH concepts and their semantics is NOT exhaustively determined on a linguistic ground. Principle **P1** focuses on the interpretation of domain symbols t_i by means of the DCH hierarchy. As **P2** focuses on purely linguistic information determined by LKB, a different notation is required.

Notice that DCH nodes *C* are in a many-to-many mapping to LKB senses. As a consequence sets $W=ext(C)$ may not receive a unique $\alpha \in LKB$ but are usually covered by more than one generalization (i.e. there is not α that is a common ancestor for all $t \in W$, implying that the intentional strength is 0). In this case an alternative can be found by partitioning *W* in more coherent (and possibly overlapping) subsets W_i. These will give independently rise to common generalizations, α_i: each one is a trade-off as the higher in the hierarchy is α_i, the larger is the size of the corresponding W_i.

4.3.1.1.2 Mapping domain concepts to lexical senses

The semantic dictionary that will result from the harmonization of DCH and LKB is a lexical hierarchy extended with the domain concepts (see Fig. 14), as metafeatures. In fact, each useful sense α in LKB will be augmented with references (e.g. the labels *t*) to domain concepts *C* linguistically interpretable as α. In terms of the properties **P1** and **P2** domain concepts are mapped to word senses in the following way. A domain concept *C* receives the minimal set of word senses in *lgen*(*C*) with the *maximal intentional strength* as subsumers of non-trivial subsets of the linguistic extension of *C*, i.e. *ext*(*C*). Usually specific entries in DCH (e.g. *Tissues*) are mapped into one ore more LKB senses (e.g. *'body_part'* and *'epithelium'* in WordNet). Vice versa one sense may be tagged by several DCH primitives (e.g. *'body_part'* as *'Digestive System'*, *'Cardiovascular System'*, *'Tissues'*, ...).

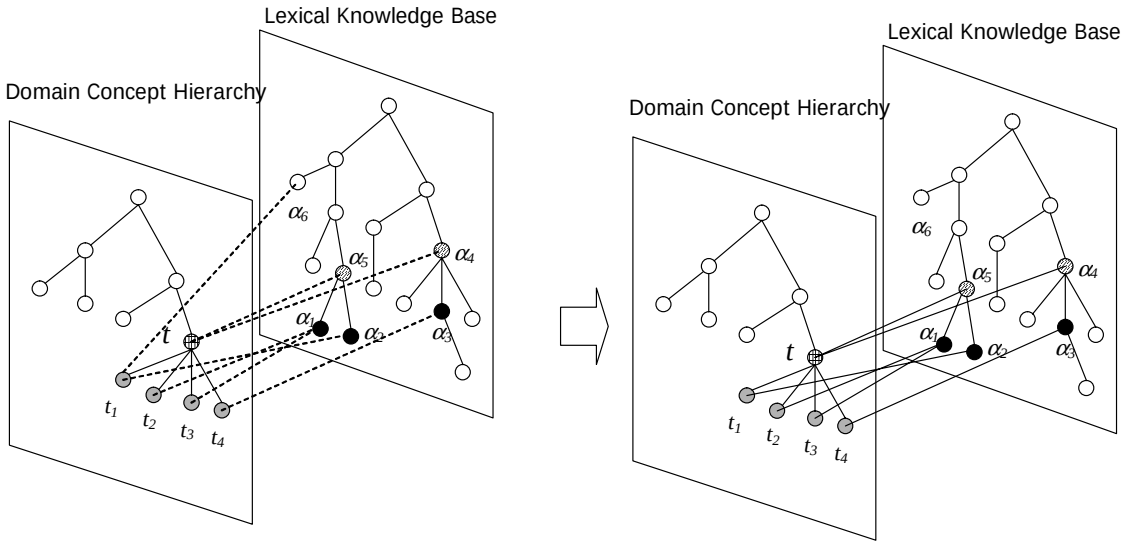


Fig. 14 Integration of domain and generic knowledge: WordNet and MeSH

In our model the notion of *conceptual density* (*cd*), as introduced by [Agirre and Rigau, 1996], is used as a measure of intentional strength (principle **P2**). The conceptual density aims to state why and how much a set of words may be considered similar according to a reference lexical hierarchy, LKB[12] . Given a set *W* of words (eventually with multiple senses) and a specific node α in the lexical hierarchy dominating at least one sense for each $w \in W$, the conceptual density $cd(W,\alpha)$ is a real value associated to the common ancestor α: it is proportional to the number of covered senses of $w \in W$ and inversely proportional to the size of sub-tree rooted at α. Therefore, the smaller the sub-tree (i.e. the more specific is α as a generalization of the senses of *w*'s), the higher is the *cd* value. Although its application in the ontology engineering framework proposed in this paper is new, this measure has been widely applied to word sense disambiguation problems: technical details are also discussed in [Agirre and Rigau, 1996].

The model we propose here requires that a triggering set *T* of DCH concepts *C* (with category labels t_C) has been previously selected. This set will drive the application of the principle **P1** and **P2** over the DCH. Then, for each concept $C \in T$, the corresponding set of linguistic expressions (*ext*(*C*) in Eq. (1)) is determined

[12]WordNet has been used as the underlying reference taxonomy for the definitions and experiments related to the conceptual density, [Basili et al., 2004].

by DCH. *ext*(*C*) and the conceptual density are then used to derive an *optimal* set of LKB senses within the linguistic generalizations of *C* (i.e. *lgen*(*C*) in Eq. (2)): this set is optimal as it is made of the intentionally strongest senses α_i that generalize all the expressions of $t \in ext(C)$. By means of a greedy technique, the generalizations α_i of non-trivial subsets $W(i) \subset ext(C)$ are selected according to decreasing values of conceptual density until the entire set is not completely covered. In this way, each $C \in DCH$ is mapped to an $\alpha(C) \in lgen(C)$ characterized by the highest intentional strength (i.e. *cd*()).

The algorithm that maps the *DCH* into the *LKB* is triggered by the subset of concepts *T* and is sketched in the following.

merge(*DCH*,*LKB*,*T*)
$\forall C \in T$
Step 1 Determine the linguistic extensions *lgen(C)*
in DCH made of all descendants of *C*
Step 2 Compute the optimal mapping $G(C) \subseteq lgen(C)$,
by a greedy selection maximizing
the **conceptual density**
Step 3 Attach *tC* to senses in *G(C)*
Step 4 $\forall t \in ext(C)$
Attach *t* to $\alpha \in$ LKB iff:
α is a sense for *t* in LKB
and
$\exists \beta \in G(C) \mid \beta$ **subsumes** α in LKB

The resulting of the above process is a *Concept Hierarchy* that integrates denotations of domain concepts with their linguistic counterparts: the former will support disambiguation in language processing, while the latter will favour linguistically consistent generalizations of general (i.e. non domain-specific) surface forms.

As previously noted, some labels in the *DCH* are not represented in the LKB: they are possibly too much specific and are uncovered by LKB. Partial forms of terms $t \in ext(C)$ are also used in the above mapping algorithm as they bring useful information for determining suitable interpretations (α_C) and their conceptual density. As the term head is the semantic carrier of multiword expressions, terms not covered in LKB are processed by backing off to the sub-terms obtained via incremental elicitation of modifiers: e.g. $w_1 w_2 ... w_n$ is reduced to the longest covered sub-term $w_i ... w_n$ that has a sense in the LKB. Unfortunately this approximation may introduce noise in the mapping process as sub-terms are usually more polysemic than complete terms.

4.3.1.2 Acquiring relational concepts from texts

In order to acquire domain knowledge relying only on text collections, we shall process the corpus to extract relevant linguistic *forms*. We expect that the linguistic forms of *relevant* relational concepts could emerge from a possibly domain independent corpus analysis process. For what concerns this analysis, we assume that a relational concept is represented in verb phrases $r=(rv,(ra_1,ra_2,\ldots, ra_n,)$ as (*boost*, ((*subj*, `Spacecraft`), (*obj*,`Satellite`), (*pp*(*into*), orbit)). Therefore, we here present an algorithm that, after the detection of *admissible surface forms* (i.e. linguistic "prototypes" written at a syntactic interpretation level), produces a ranking according to their domain relevance (i.e. their frequency).

In the following sections, we will first define the equivalence among admissible surface forms while estimating the size of the search space of the ranking algorithm. Secondly, an efficient algorithm for the estimation of the importance function based on the frequency of the relations in the target corpus is presented in Sec. 4.3.1.2.2

4.3.1.2.1 Admissible surface forms: size of the problem

A relational concept may appear in a number of different contexts where verbs have some additional arguments. If the corpus *C* may be seen as a collection of verb contexts $c=(v,(a_1,a_2,\ldots, a_n,)$ where *v* is the governing verb and each argument a_i is a couple (g_i,c_i) representing its grammatical role g_i (e.g. subject, object, pp(for), pp(to), etc.) and the concept c_i semantically governing it, the problem is reduced to understand which are the more stable relationships established by each verb. Note that a context $c \in C$ is a positive example of the target relation $r \in R$ if $rv=v$ and *r* partially cover *c*, i.e. the arguments of *r* should then appear in any order in the context *c*.

An algorithm evaluating the relevance of all the possible relations $(rv,(ra_1,ra_2,\ldots, ra_n,)$ works on huge search space. The number of different relations are obtained by partitioning the corpus *C* according to the verb governing the contexts. For each verb *v*, a subset of the corpus is then defined as

$$C(v)=\{(a_1,\ldots,a_n,)|\ (v,(a_1,\ldots,a_n,)) \in C\}$$

Defining $A_\Lambda(v)$ and $A_\Sigma(v)$ respectively as the possible lexicalised arguments and the possible syntactic arguments of a relation $r(v) \in R(v)$:

$$A_\Lambda(v) = \{a | \exists (a_1,...,a_n) \in C(v) \wedge \exists i.a_i = a\} \quad (1)$$

$$A_\Sigma(v) = \{(s, object) | \exists i.g_1 = s \wedge \exists ((g_1,c_1),...,(g_n,c_n)) \in C(v)\} \quad (2)$$

the set $R(v)$ of the possible relations for the named *v* is the following $R(v) = \bigcup_{i=1...MC(v)} R_i(v)$ where $R_i(v)$ is the collection of all possible combination without repetition of *i* objects extracted from the set. The distinction between lexicalised and syntactic arguments is useful to take into account the fact that some relations may have a recurrent argument whose surface concept is not recurrent. In these cases, a generalisation of the argument concept, i.e. *object*, is retained.

If $R(v)$ is the set of all the relations for the investigated verb *v*, the domain importance of each $r(v) \in R(v)$ should be assessed. Therefore, the evaluation of the frequency of the relation $r(v)$ over the corpus $C(v)$ will be used.

Given the defined sets, the size of the $R(v)$ set is, in the worst case, the following:

$$|R(v)| = \sum_{i=1...MC(v)} \left(\frac{|A(v)| + i - 1}{i} \right) \quad (3)$$

where $MC(v)$ is the maximum context size for the verb *v* in $C(v)$. It is worth noticing that $|R(v)|$ values lie in a very large range, due to the size of $A(v)$. In the next section we will focus on a measure of relevance (for the target domain) that allows to systematically reduce the size of the space where pattern selection is applied for each verb *v*.

4.3.1.2.2 Estimating relational concept relavance

In order to tackle the inherent complexity due to the argument order freedom neglected in [Yangarber, 2003], we defined an informed exploration strategy relying on these observations: (1) the target of the analysis is to emphasize the more important relations arising from the domain corpus; (2) the frequency of a specific relation strictly depends on the frequency of a more general relation (hierarchy of relations). A very simple but effective domain relevance estimator is the frequency of the relation over the corpus. As a consequence

the complexity of the search algorithm if only promising relations are explored, i.e. patterns whose generalisations are over a frequency threshold.

The idea is then to drive the analysis by using pattern generalisation that may be obtained projecting the patterns on their "syntactic" counterpart. The projection $\Sigma(r)$ of the relation r over the syntactic space Σ is defined as follows:

$$\hat{\Sigma}(r) = \left(\hat{\Sigma}(ra_1),...,\hat{\Sigma}(ra_m)\right)$$

where $\hat{\Sigma}(ra_i) = ra_i$ if ra_i is a "syntactic" argument ($ra_i \in A_\Sigma(v)$) or $\hat{\Sigma}(ra_i) = (s_i, object)$ if $ra_i = (g_i, c_i)$ is a lexicalised argument ($ra_i \in A_\Lambda(v)$). The resulting search space $R_\Sigma(v) = \left\{\hat{\Sigma}(r) \middle| r \in R(v)\right\}$ is greatly smaller than $R_\Sigma(v)$ since $|A_\Lambda(v)| >> |A_\Sigma(v)| = \#\, preposition + 2$. This search space can be used for the extraction of the more promising generalised relations. This subset $\overline{R}_\Sigma$ can be used for narrowing the search space of the following step. In fact, when the acceptance threshold is settled, resultant admissible relations are confined in the following set:

$$\overline{R}(v) = \left\{r \middle| \hat{\Sigma}(r) \in \overline{R}_\Sigma(r)\right\} \quad (4)$$

The overall domain importance estimation procedure may take also advantage from considering that the order of the relation arguments may be fixed after the analysis of the promising syntactic patterns. The final counting activity can be thus performed with a simple sorting algorithm of the $O(nlog(n))$ complexity.

4.3.1.3 Machine learning techniques for classifying linguistic forms

Retrieved relational concepts (*forms*) should then be organized in the ontological model, in order to support an efficient retrieval procedure in textual documents stored in the grid. Once the hierarchy of relational concepts is in place after the *concept formation phase*, the task of positioning the forms in the hierarchy may be seen as a classification process.

We will explore the possibility of a classification process carried out using ML techniques, applied to lexical and semantic information. The *feature-value vector* model underlying many ML algorithms suggests an observation space in which dimensions represent features of the object to be classified and dimension values are the values of the features as observed in the object. Each instance object is then a point in the feature space, i.e. if the feature space is $(F_1,\ldots\ldots, F_n)$ an instance I is:

$$I = (f_1,....., f_n) \quad (5)$$

where each f_i is the value of the feature F_i for I.

Classifying linguistic forms with ML algorithms implies their translation in observable object. As we want to investigate the use of general purpose lexical semantic information such as WordNet [Miller, 1995], we propose here the notion of *semantic fingerprint* to introduce a conceptual hierarchy in a feature-value model. Hierarchies in the feature values are somehow in contrast with their expected *flatness*. To use this information, these hierarchies should be somehow reduced to a flat set *SF* where the problem of the inherent structure is simply forgot.

A word w (a verb or a noun) will leave its fingerprint $SF(w)$ on the set *SF* that represents all the active senses with respect to the chosen semantic interpretation catalogue *SF*. The semantic fingerprint of word w is:

$$SF(w) = \{s \in SF | s generalises' and s' \in senses(w)\} \quad (6)$$

where *senses*(*w*) are all the senses activated by the word *w* in the considered semantic resource. It will be the task of the machine learning algorithm the selection of the sense (or the senses) more promising for representing the investigated relationship. The algorithm will therefore also work as verb/noun sense disambiguator if the semantic information and the way we use it demonstrates to be useful.

Integrating the semantic fingerprint in the feature vector model is straightforward. Given an S_i={*true, false*} for each element in *SF*, the subpart of the feature space related to the semantic fingerprint is $S_1 \times ... \times S_n$ where *n* is the cardinality of *SF*. Each instance *i* containing the word *w* will have the feature value s_j=*true* if $s_j \in SF(w)$ and s_j=*false* otherwise.

With the semantic fingerprint abstraction we investigated two "semantic" models against a "bag-of-word" model. These are originated from the assumption that verbs play a relevant role in the problem under analysis. Then, the proposed models are:

$$verb\text{-}gen: V \times W_1 \times ... \times W_n \times VS_1 \times ... \times VS_k \quad (7)$$

$$noun\text{-}gen: V \times W_1 \times ... \times W_n \times NS_1 \times ... \times NS_m \quad (8)$$

where *V* ranges over all the possible verbs, $W_1 \times ... \times W_n$ represents the "bag-of-word" approach collecting all the verb arguments, $VS_1 \times ... \times VS_k$ is the semantic fingerprint for the verbs, and, finally, $NS_1 \times ... \times NS_m$ is the semantic fingerprint for the nouns. The baseline model, that it is in itself a good model, is called *plain* and it collects verbs and the bag-of-word of the arguments (i.e. $V \times W_1 \times ... \times W_n$).

4.3.1.4 Experimental analysis

For both clarification and evaluation purposes hereafter we will refer to a specific domain scenario [13] (financial news) over which to analyse the performance of the knowledge modelling as well as the retrieval task. We firstly prepared a relevant test set in order to clarify the final classification task. The manual tagging procedure and the results are presented in Sec.4.3.1.4.1. Then, we have experimented our semantic-fingerprint-based models using well-assessed machine learning algorithms gathered in Weka [Witten and Frank, 1999]. It is worth noticing that the *cross-algorithm validation* can give hints on the relevance and the stability of the chosen feature spaces and on the correctness of the proposed model. The results of this investigation are reported in Sec.4.3.1.4.2.

4.3.1.4.1 Test set preparation

In the test set preparation, our aim has been to have two different sources of information in order to cross check the results of the experiment. Given a catalogue *C* of relational concepts, we have produced:

- *classified forms*: a set of one-to-many associations between the concepts in *C* and the linguistic normalised forms
- *classified sentences*: a set of one-to-many associations between the concepts in *C* and sentences in the analysed corpus somehow related to the analysed linguistic forms

[13] While Spacecraft Design is our reference domain, we will analyse results from another domain scenario (financial news) due to a couple of reasons: (1) SHUMI is still an on-going project on which, currently, final results must still be produced; (2) we considered more than one domain to verify the overall validity of our approach. The implemented methodology has been defined in a general framework and will be used for each domain. As it requires the support of a human expert for validation purposes, at the time of paper writing we could base on the competences related to economical domain. This forced the choice of final results to be shown.

For the experiments, we used a corpus consisting of financial news, a text collection of around 12,000 news items published from the Financial Times in the period Oct./Dec. 2000. As described in Sec.4.3.1.2, we, firstly, run a *corpus processing phase* selecting around 44,000 forms appearing more that 5 times. Secondly, in the *concept formation phase* a domain expert inspecting the top ranked forms defined 12 target relational concepts (see Fig. 15).

Class		Forms	Sentences
1	*RELATIONSHIPS AMONGS COMPANIES*		
1-1	Acquisition/Selling	157	619
1-2	Cooperation/Splitting	96	471
2	*INDUSTRIAL ACTIVITIES*		
2-1	Funding/Capital	12	86
2-2	Company Assets (Financial Performances , Balances, Sheet Analysis)	166	1335
2-3	Staff Movement (e.g. Management Succession)	70	355
3	*GOVERNMENT ACTIVITIES*	*12*	*283*
3-1	Tax Reduction/Increase	3	40
3-2	Anti-Trust Control		19
4	*JOB MARKET - MASS EMPLOYMENT/UNEMPLOYMENT*	*7*	*50*
5	*COMPANY POSITIONING*	*4*	
5-1	Position vs Competitors	10	174
5-2	Market Sector	10	369
5-3	Market Strategies and plans	149	1512
6	*STOCK MARKET*	*2*	*3*
6-1	Share Trends	319	1197
6-2	Currency Trends	2	30

Fig. 15 The class (relational concepts) hierarchy of the financial domain, and corresponding forms and sentences distributions.

The *classification phase* has been performed by 2 human experts. They were given two separate set of normalised linguistic forms, two separate set of sentences extracted automatically from the corpus and a non-ambiguous definition of each class. The two experts were given respectively 3500 and 2200 forms to classify, taken from the first 6500 forms produced in the corpus processing phase. To evaluate the consistency between the classifications produced by the two experts, 300 of the given forms were in common, and over those forms the rater agreement was evaluated.

In case of doubt during the classification the expert could ask the system to show one or more sentences instance of the form, in order to gain enough information to classify the form itself. Annotators were also asked to classify all the shown sentences.

At the end of the phase, out of the normalised forms considered, 787 were retained as useful by the first expert, 298 by the second, i.e. the information carried in the words or in the named entity classes survived in the form has been considered sufficient to draw a conclusion on the classification. Moreover, the first expert classified 6609 sentences and the second 3550.

The two data sets, *classified forms* and *classified sentences*, have then been prepared. The first one consists of the 1091 forms obtained merging the two experts forms retained sets (for the 300 common forms, in case of disagreement the first expert class has been chosen). The second data set comprise the 6609 sentences classified by the first expert. The overall distribution of forms and sentences, for both the domain experts, is reported in Fig. 15. The inter-annotation agreement on the normalised forms is 90%, while the agreement on the sentences is 74%. These results show us a sufficient consistency over the data set, that can be thus considered a well defined gold standard for the experiments.

Method		*plain*	*verb-gen*	% inc/dec	*noun-gen*	% inc/dec
Trees	j48.J48	63,91%	63,68%	-0,23%	64,37%	+0,46%
	ID3	59,31%	59,31%	0	59,54%	+0,23%
	DecStump	26,44%	31,95%	+5,52%	26,44%	0%
Lazy	IB1	58,39%	63,22%	+4,83%	57,70%	-0,69%
	IBk	62,53%	65,98%	+3,45%	60,69%	-1,84%
Rules	j48.PART	59,77%	60,00%	+0,23%	63,22%	+3,45%
Bayes	NaiveBayes	53,33%	58,85%	+5,52%	40,23%	-13,10%
Misc	VFI	59,31%	57,24%	-2,07%	58,39%	-0,92%
	HyperPipes	60,92%	62,76%	+1,84%	62,07%	+1,15%

Fig. 16 Results on the set of *classified forms*, using a 5-fold cross-validation (baseline is 27%)

Method		*plain*	*verb-gen*	% inc/dec	*noun-gen*	% inc/dec
Trees	j48.J48	59,19%	64,80%	+5,61%	64,98%	+5,78%
Lazy	IBk	59,19%	54,72%	-4,47%	53,99%	-5,34%
Bayes	NaiveBayes	47,25%	54,03%	+6,78%	42,48%	-4,77%
Misc	VFI	43,81%	52,08%	+8,27%	51,84%	+8,03%
	HyperPipes	31,21%	42,56%	+11,35%	42,48%	+11,27%

Fig. 17 Results on *classified sentences*, using a 5-fold cross-validation (baseline is 40%)

4.3.1.4.2 Analysis of the results

The classification problem over the two different proposed data set has been therefore analysed with a pool of algorithms. We firstly analyse the results on the *classified forms* and then we check our intuitions on the *classified sentences.*

For the first set, the *classified forms*, results are reported in Fig. 16. The baseline of the classification is around 27%, corresponding to a naive classification of all the instances in the more probable class (i.e. 6-1). All the algorithms report both in the lexical and the two lexical-semantic spaces better results with respect to the baseline, showing that the chosen features convey the right information for our classification problem. Moreover, the use of the semantic information seems to be relevant, as it emerges in the performance improvement obtained with the majority of the investigated algorithms using the semantic prints on both verbs and nouns.

In particular, the verb semantic generalization features seem to be particularly useful: the best performance for the vast majority of the tested algorithms is in fact achieved using the lexical-semantic verb

space. Furthermore, the experiment overall best performance is obtained by the IBk algorithm working on this space.

In order to verify how the verb semantic information drives the classification, it can be interesting to examine the rules produced by a rule based algorithm, such j48.PART. This algorithm derives its rules from a pruned partial decision tree built using the C4.5 implementation [Quinlan, 1993]. One of these rules that involves semantic information, is the following: *price=no* ∧ *job=no* ∧ *hire=no* ∧ *succeed=yes* ∧ *entityNE=yes* 2-3. This rule indicates that every sentence containing a verb of *succession* (i.e., a troponym, in the Wordnet sense, of the verb *succeed*) together with an *entityNE* (that is, a company or a person) has to be classified in class 2-3 (*staff movement* events). This semantic generalised rule, according to the Wordnet hierarchy, therefore classifies verbs of *succession* like *enter, supplant, replace, substitute*. Such a general rule can not be captured in a simple lexical space.

Analysing the results of Fig. 16, the noun semantic generalization seems to be slightly less effective than the one on verbs. It is interesting to notice how in the tree obtained by j48 the noun semantic information is used. For instance, the presence in a form of a noun whose *base concept* (i.e. noun semantic generalization in EuroWordNet [Vossen, 1998]) is *financial_obligation* is used to capture *"government activities: tax-reduction/increase"* events (class 3-1). In this way forms that contain nouns like *debt, rate, tax* are all classified in class 3-1. This simple rule has been very effective on our data set, classifying positive instance with 100% precision.

For the experiment over the *classified sentences* (Fig. 17) we used a reduced pool of algorithm, representative of the different classification methodologies. In this case the baseline is around 40%. Similarly to the previous experiment, the results show a performance improvement using the verb and noun semantic information. In that case the improvement is even more sensible, thanks to the larger data set which emphasize the beneficial effect of the information carried by the used features. Looking at the decision trees produced by the j48 algorithm, it can be noticed that in the lexical space the verb lemmas are the most selective information, while in the lexical-semantic space the semantic verb generalisations and the noun generalizations and lemmas tend to discriminate over the data set more than the verb lemmas. Since the introduction of the semantic spaces improves the algorithm performance, it can be stressed again that this kind of information has an important discrimination power.

Summarizing: we introduced a knowledge based approach to improve development of the Semantic Grid conceptual layer, based on NLP and ML techniques and methodologies. Our approach is strongly based on the idea that an ontological organization of the knowledge and the use of terminological and semantic information automatically extracted from a domain corpus can support the development of a coherent and consistent Semantic Grid infrastructure. The explicit use we make of many-to-one mappings between linguistic forms and their corresponding meaning (i.e. relational concepts) is strengthened by its diffusion in other linguistic applications. Many researches are in fact devoted to propose methods for automatically building equivalence classes of patterns in fields such as Information Extraction [Yangarber, 2003],[Riloff, 1996], Question Answering [Ravichandran and Hovy, 2002], Terminology Structuring [Morin, 1999], or Paraphrasing [Barzilay and McKeown, 2001],[Kaji et al, 2002]. As for all the methods, the use of some previous specific knowledge seems mandatory, i.e. focused and structured templates plus examples in [Yangarber, 2003],[Riloff, 1996], definitions and examples of the target relationships in [Morin, 1999], [Ravichandran and Hovy, 2002], and parallel corpora for [Barzilay and McKeown, 2001], we tried to attack the problem from a different perspective. As in a Semantic Grid scenario previous specific knowledge is not always available.

Many issues are still open, firstly those related to the knowledge publishing (as described in [de Roure and Shadbolt, 2003b]) and the development of a related usable tool.

4.3.2 Methods for Ontology Learning from texts

In this section we introduce further approaches to learn ontologies from text to what previously described in Sec.4.3.1, proposed in the AI community. Acquiring domain knowledge for building ontologies is an expensive task that requires much time and many resources. Ontology learning can be defined as the set of methods and techniques used to build an ontology from scratch, or to enrich and adapt an existing ontology with semi or automatic tools integrating several different sources. Other terms are also used in literature to refer to the semi-automatic construction of ontologies like ontology generation, ontology mining, ontology extraction, etc. Several approaches exist for the partial automation of the knowledge acquisition process: all

of them require human interaction. We focus here on natural language analysis and machine learning techniques used for such automation. The following paragraphs have been partially excerpted from the excellent survey of ontology learning methods and techniques delivered by the OntoWeb project Consortium [Gomez and Manzano, 2003].

4.3.2.1 Approaches to Ontology Learning

We can identify different approaches to ontology learning with respect to the kind of knowledge considered in the learning process: ontology learning from text, from dictionary, from knowledge base, from semi-structured schemata and from relational schemata, as in [Maedche and Staab, 2001]

1. **Ontology learning methods from texts** consist of extracting ontologies by applying natural language techniques to texts analysis.
2. **Ontology learning from dictionary** bases on the use of a machine readable dictionary to identify relevant concepts as well as relations among them.
3. **Ontology learning from a knowledge base** aims to learn an ontology using as source previously defined knowledge bases.
4. **Ontology learning from semi-structured data** looks for eliciting an ontology from sources which have any predefined structure, such as XML schemas.
5. **Ontology learning from relation schemas** aims to learn an ontology extracting relevant concepts and relations from knowledge in databases.

Ontology learning from text processing is the area on which RTV moves; in fact it seems to us the most appealing as it is helpful in dynamically producing ontologies for real applications and for any application domain: we will focus on symbolic and statistics-based approaches: the trade-off between the two is that statistics-based approaches allow for better scaling, while symbolic approaches might turn up being more precise. A coarse-grain classification of the most well-known approaches in these groups can be summarized in the following:

- **Pattern-based extraction.** Pattern-based approaches [Morin, 1999] [Hearst, 1992] in general are heuristic methods using regular expressions that originally have been successfully applied in the area of information extraction. In this lexico-syntactic ontology learning approach the text is scanned for instances of distinguished lexico-syntactic patterns that indicate a relation of interest, e.g. the taxonomic relation. Thus, the underlying idea is to define a regular expressions that captures re-occurring expressions and map the results of the matching expression to a semantic structure, such as taxonomic relations between concepts. A relation is recognized when a sequence of words in the text matches a pattern. For instance, a pattern can establish that if a sequence of *n* names is detected, then the *n-1* first names are hyponyms of the *n*th
- **Association rules.** Initially defined on the database field as "*… a set of transactions, where each transaction is a set of literals (called items), an association rule is an expression of the form X implies Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database which contain X tend to contain Y*" [Agrawall et al., 1993], association rules are used on the data mining process to discover information stored on databases if we already have a rough idea of what we are looking for [Adriaans and Zantinge, 1996]. The association rules method for ontology learning has been originally described and evaluated in [Maedche and Staab, 2000] later on. The association rules have been used [Maedche and Staab, 2001] to discover non–taxonomic relations between concepts, using a concept hierarchy as background knowledge.
- **Conceptual clustering.** [Faure and Poibeau, 2000] have shown a cooperative machine learning system called ASIUM which is able to acquire taxonomic relations from syntactic parsing. Their system is based on a conceptual clustering algorithm, where basic clusters are formed on head words occurring in the same syntactic role (i.e. with the same verb after the same preposition). Aggregates of basic clusters form new concepts and the hierarchies of concepts form the ontology. Concepts are grouped according to

the semantic distance between each other to make up hierarchies. Several metrics can be used to calculate the semantic distance between two concepts based on different factors and must be provided in these methods.

- **Ontology pruning** [Kietz et al., 2000]. The objective of ontology pruning is to build a domain ontology based on different heterogeneous sources. It usually bases on a generic core ontology as a top level structure for the domain-specific ontology, and a dictionary which contains important domain terms described in natural language used to acquire domain concepts. These concepts are classified into the generic core ontology. Then, domain-specific and general corpora of texts are used to remove concepts that were not domain specific. Concept removal follows the heuristic that domain-specific concepts should be more frequent in a domain-specific corpus than in generic texts.
- **Concept learning** [Hahn and Schulz, 2000] introduced a methodology for the maintenance and refinement of domain specific taxonomies. An ontology is incrementally updated as new concepts are acquired from real-world texts. The acquisition process is centred around linguistic and conceptual "quality" of various forms of evidence underlying the generation and refinement of concept hypotheses. In particular they consider semantic conflicts and analogous semantic structures from the knowledge base into the ontology in order to determine the quality of a particular proposal. Thus, they extend an existing ontology with concepts and taxonomic relations between concepts.

4.3.2.2 Methods for Ontology Learning from texts

To clarify goals and scopes of the different learning processes, we introduce hereafter a short survey of some of the most representatives approaches together with a short description, the steps used to learn and the knowledge sources (if different from raw texts) used for learning, as well as the adopted techniques, the domain it has been used and tested on. The methods and approaches presented in this section (hereafter in alphabetical order) are: Agirre and colleagues' method, Gupta and colleagues' approach, Hwang's method, Missikoff and colleagues' method, Moldovan and Girju's method.

Finally, we briefly introduce some state of arts tools and architectures adopted for the task, including their main goals and techniques used during the learning process, where user interacts with the process, the kind of resources used, the software architecture, the possibility of interoperate with other tools, the import and export facilities that the tool provides, the interface facilities, and other references to bibliography and URLS. The tools presented in this section are: ASIUM, CORPORUM-Ontobuilder, LTG Text Processing Workbench, Mo'K Workbench, Ontolearn, SOAT, SubWordNet Engineering Process Tool, SVETLAN', Text-To-Onto.

4.3.2.2.1 Agirre and colleagues' approach

Agirre *et al.* [Agirre et al., 2000] aim to enrich the concepts in existing large ontologies using text retrieved from the Web. Main goal of this approach is to overcome two shortcomings of general purpose ontologies like WordNet: the lack of domain/topic information in concepts, and the proliferation of a multitude of different senses, often closely related or overlapping. The proposed method starts by collecting a set of documents containing surface realizations (i.e. terms) related to a given concept. For each sense of a concept in the ontology, and in order to construct lists of closely related words for each one, the words in the text that are most closely related to the concept are collected. The approach is based on the use of topic signatures, used in text summarization, that have been described in [Hovy and Lin, 1999] and [Lin and Hovy, 2000]. The strategy proposed to build such lists is as follows: at first, the information contained in the ontology is used to build the queries that retrieve relevant documents in respect to the given concept sense. Then retrieved texts are grouped for each word sense in a collection. For each collection, words and their frequencies are compared with data in other collections covering different senses of the same concept.

The method proposes four steps to enrich an existing ontology:

1. *Retrieve relevant documents for each concept.* The goal of this step is to retrieve documents related to an ontology concept from the web. Queries are set for each concept sense using the information stored in the ontology, such as synonyms of the concept, hyperonyms, attributes, etc. The documents that could belong to more than one sense are discarded, and documents related to the same concept sense are grouped together to form collections, one for each sense. This phase is supervised.

2. Build topic signatures. The documents in each collection, related to a specific concept sense, have to be processed in order to extract the words and their frequencies using a statistical approach. Then, the data from one collection is compared with the data in the other collections. The words that have a distinctive frequency for one of the collections are grouped in a list, which then constitutes the topic signature for each concept sense. This phase is unsupervised.

3. Clustering word senses. Given a word, the concepts that lexicalise its word sense are hierarchically clustered. To carry out this task different topic signatures are compared to discover shared words, in order to determine overlaps between the signatures. Various semantic distance metrics and clustering methods can be used for this purpose.

4. Evaluation. This is performed in the same way as a word sense disambiguation task. The topic signatures and hierarchical clusters are used to tag a given occurrence of a word in another corpus with the intended concept using different disambiguation algorithms.

The approach has been tested with WordNet and the benchmark corpus SemCor to perform the evaluation task.

4.3.2.2.2 Gupta and colleagues' approach

[Gupta et al., 2002] present an approach to acquire and maintain sublanguage WordNets from domain specific textual documents. The approach aims to enable rapid development of SubWordnets for NLP applications and proposes an iterative three-step lexicon engineering cycle for developing SubWordNets as in what follows:

1. *Discover Concept Elements*: goal of this step is to discover concept elements, including words, generated multi-word phrases, and potential relationships among these elements that occur in input sublanguage documents. For example, "Marine Mountain Warfare Training" and "Maritime Interception Operation Training" would be discovered as multi-word phrases in the Navy Lessons domain. An unnamed relation between them could be discovered and suggested to the user. Subsequently, a user could identify the relation as of meronym/holonym type. This typically uses a combination of shallow language and text processing along with learning, discovery, and extraction techniques.

2. *Identify Concepts*: The objective of this step to identify new concepts and relations from phrases and relations discovered in the previous step. Concept identification is supported by grouping phrases into concept nodes and establishing concordance with synsets in WordNet. The new concept nodes and relationships can be used to update the SubWordNet.

3. *Maintain Concepts (Update SubWordNet)*: This step allows controlled insertion, deletion, and updating of concepts and relations derived from the previous step in a SubWordNet while maintaining its integrity. Users can iterate through these steps with as many sublanguage documents as needed to develop SubWordNets and to maintain them on an ongoing basis.

4.3.2.2.3 Hahn and colleagues' approach

Hahn and colleagues present a method for the maintenance [Hahn and Schnattinger, 1998] and growth [Hahn and Markó, 2001] of domain-specific taxonomies based on natural language text understanding. A given taxonomy is incrementally updated as new concepts are acquired from real-world texts. The acquisition process is focused around the linguistic and conceptual "quality" of various forms of evidence underlying the generation and refinement of concept hypotheses. On the basis of the quality of evidence, concept hypothesis are ranked according to credibility and the most credible ones are selected for assimilation into the domain ontology. In this approach, learning is achieved by the refinement of multiple hypotheses about the concept membership of an instance. New concepts are acquired by taking into account two sources of evidence: background knowledge from the domain texts, and linguistic patterns in which unknown lexical items occur.

The model introduced for text knowledge elicitation can be summarized in the following general steps.

1. *Language processing.* It aims to determine structural dependency information from the grammatical constructions in which an unknown lexical item occurs in terms of the corresponding parse tree. The conceptual interpretation of parse trees involving unknown lexical items in the terminological knowledge base is used to derive concept hypotheses, which are further enriched by conceptual annotations reflection structural patterns of consistency, analogy, etc. This kind of initial evidence is represented by corresponding sets of linguistic and conceptual quality labels.

2. *Evaluation of the quality labels.* As it has been mentioned above, there are two kinds of quality labels to be evaluated. The first one is the *linguistic quality label* that reflects structural properties of phrasal patterns or discourse contexts in which unknown lexical items occur, and depending on the type of the syntactic construction, different hypothesis generation rules may fire. The second type is the *conceptual quality labels*, that results from comparing the representation structures of a concept hypothesis with those of alternative concept hypotheses or already existing representation structures in the underlying domain knowledge base from the viewpoint of structural similarity, compatibility, etc.

3. *Quality estimation.* The overall credibility of single concept hypotheses is estimated by taking the available set of quality labels for each hypothesis into account. Thus, the final computation of a preference order for the entire set of competing hypotheses. This output is a ranked list of concept hypotheses. Whenever new evidence for or against a concept hypothesis is brought, all concept hypothesis are re-evaluated.

4. *Evaluation.* An empirical evaluation of the text knowledge acquisition process is carried on by using different measures that evaluate the learning accuracy and the learning rate. The learning is achieved by the refinement of multiple hypotheses about the concept membership of an instance.

4.3.2.2.4 Hwang's approach

[Hwang, 1999] focuses on the problem of locating, evaluating, retrieving, and merging information in an environment in which new information sources are continuously added.

As part of the InfoSleuth project, an approach has been developed to represent and retrieve information from large textual databases. It is based on the use of dynamic ontologies that capture the semantics of information that resides the documents. The ontology is organized in simple taxonomies. Concepts from the taxonomy are then identified within the documents to enable the retrieval process. To carry out the process, NLP and machine learning techniques are used.

The procedure for generating the ontology follows these steps:

1. *Human experts provide* the system with a small number of *seed-words* that represent high-level concepts. Relevant documents are collected from the web automatically (with POS-tagged or otherwise unmarked text).

2. The system *processes the incoming documents*, extracts only those phrases that contain seed-words, generates corresponding concept terms and places them in the "right" place in the ontology, and alerts the human experts of the changes. This feature is named "discover-and-alert". At the same time, it also collects candidates for seed-words for the next round of processing. The iteration continues a predefined number of times. The method indexes documents according to the concepts identified within them for future retrieval and also the "context lines" in which the concept has been discovered to show how the concept was used in the text as well as frequency of co-occurrence inside each document.

3. Several kinds of *relations* are extracted. Examples of relations are: "is-a", "part-of", "manufactured-by", "owned-by", etc, which are extracted based on linguistic features. The "assoc-with" relation is used to define all relations that are not an "is-a" relation. The distinction between "is-a" and "assoc-with" relations is based on a linguistic property of noun compounds. The method only can discover some of the attributes associated with certain concepts based on linguistic characters.

4. In each iteration, a *human expert is consulted* to ascertain the correctness of the concepts. If necessary, the expert has the right to make the correction and reconstruct the ontology. While constructing the ontology, the method also allows the indexing of documents for future retrieval.

There are some *problems* for automatically generating ontologies with this approach such as: *syntactic structural ambiguity, recognising different phrases that refer to the same concept, word sense problems, etc.*

4.3.2.2.5 Kietz and colleagues' approach

This method [Kietz et al., 2000] is a generic method used to discover a domain ontology from given heterogeneous resources by the use of natural language analysis techniques. It is a semi-automatic process as the user takes part in the process. In their approach, authors have adopted the balanced cooperative modelling [Morik, 1993], where the work of building the ontology is distributed between several learning algorithms and the user. The method is based on the assumption that most concepts and conceptual structures of the domain to be included in an ontology as well as the terminology of a given domain are described in documents. The authors propose to learn the ontology having as a base a core ontology (it could be: SENSUS, WordNet, etc.)to be enriched with new specific domain concepts. New concepts are identified using NL analysis techniques over the resources previously identified by the user.

The resulting ontology is pruned and focused to a specific domain by the use of several approaches based on statistics. Finally, relations between concepts are learnt by applying learning methods. Such relations are added to the resulting ontology.

This method consists of the following steps: select sources, concept learning, domain focusing, relation learning, and evaluation of the resulting ontology. The process is cyclic in the sense that the resulting ontology can be refined by applying the method iteratively.

The acquisition process proposed by this method consists of the following steps.

1. *Select sources.* The process starts with the selection of a generic (top-level) ontology, which is used as a base for the learning process. This ontology should contain generic and domain concepts. The user will specify which documents should be used in following steps to refine and extend the previous ontology. By its nature, sources are heterogeneous in their formats and contents. Sources can be free text documents, semi-structured text, domain text, and generic text. Documents can deal with general or domain specific.

2. *Concept learning.* Its goal is to acquire new (both generic and specific) concepts to decide if the discovered concepts are specific enough to be included in the ontology. The method proposes to analyse the frequency of the terms. Those terms that are more frequent in a domain-specific corpus than in a generic corpora (and they are not contained in the given ontology) should be proposed to the user to decide whether they should be incorporated into the ontology. The selection of the tools depends on the language to be processed (Spanish, English, German, etc.).

3. *Domain focusing.* Its purpose is to prune the enriched core ontology by removing general concepts.

4. *Relation learning.* Frequency analysis can be used to learn ad hoc domain relations. This is founded over the underlying idea that frequent couplings of concepts in sentences can be considered as relevant relations between concepts in the ontology. This approach is used to find frequent correlations between concepts and it is based on the association rule's algorithm proposed in [Skrikant and Agrawal, 1995].

5. *Evaluation.* Its goal is to evaluate the resulting ontology and to decide whether it is necessary to repeat the process again.

This method is supported by the tool Text–To–Onto [Maedche and Volz, 2001].

4.3.2.2.6 Missikoff and colleagues' approach

OntoLearn [Missikoff et al., 2002] is a method for ontology building and enrichment using NL and machine learning techniques. The method proposes WordNet as a source of prior knowledge to build a core domain ontology, after pruning all of the unspecific domain concepts. The method follows both statistical approaches (to determine the relevance of one term for the domain) and semantic interpretation, based on machine learning techniques (to identify the right sense of terms and the semantic relations among them).

The method proposes three main steps to achieve its goals [Velardi et al., 2002]: terminology extraction, semantic interpretation, and creation of a specialized view of WordNet.

1. *Terminology extraction*. Terms and combinations of terms, such as "*last week*", are extracted from a parsed corpus by using NL techniques. Terms are considered as the surface appearance of relevant domain concepts. High frequency in a corpus is a property observable for terminological as well as non-terminological expressions. The method proposes to use a measure of the specificity of a terminology candidate with respect to the target domain via comparative analysis across different corpora. For this purpose, two different elements are defined to determine a threshold for the relevance of a domain terminology expression. The first element is the *domain relevance score,* which is a measure of the amount of information captured in the target corpus relative to the entire collection of corpora used for the learning process. The second element is the *domain consensus* which captures those terms that appear frequently across a given domain's documents.

2. *Semantic interpretation*. Main goals of this step are to determine the right concept sense for each component of a complex term, like a semantic disambiguation process, and then to identify the semantic relations holding among the concepts to build a more general concept. At the end of this step, a domain concept forest will be obtained, showing the taxonomic (as well as other) relationship among complex domain concepts represented by expressions. To carry out this step, it is necessary to use semantic and linguistic resources (the method has been tested with WordNet) to assist in the semantic interpretation of terms. This step consists of two main processes, the first of which is a *semantic disambiguation process*. The sense of each word is defined as a synset of synonyms (or the right synset in WordNet in which the word can be placed). The second process is extracting semantic relations that hold between the components of complex terms extracted in the previous step.

3. *Creating the domain ontology*. This step aims to integrate the taxonomy obtained in the previous step with a core domain ontology. In the case that an existing domain ontology is not available, the method proposes to create a new one from WordNet, pruning concepts that are not related to the domain, and extending it with the new domain concept trees under the appropriate nodes.

This method has been developed and tested inside the *Harmonise* and *Fetish* European projects, both in the tourism domain.

4.3.2.2.7 Moldovan and Girju's approach

It is a method for discovering domain-specific concepts and relationships trying to extend an existing ontology, like WordNet, with new knowledge acquired from parsed text. The source for discovering new knowledge is a non-specific domain corpus, augmented by using other lexical resources like domain specific and general dictionaries. The user provides a number of domain-specific concepts that are used to discover new concepts and relations from the source. The user performs the validation of the process and confirms the correctness of both concepts and relations learnt.

To enrich an existing ontology with new concepts and relations, the following five steps are carried on, as in [Moldovan and Girju, 2001]:

1. Select seed concepts. A few seed-concepts, that a user considers important for the target domain ontology, are selected. This set of seed-concepts is extended with each concept's corresponding synonyms to form a synset. The knowledge that is to be acquired must be related to one or more of these seed-concepts, and consists of new concepts not defined in the existing ontology as well as new relations. The new relations link the new concepts with other concepts, some of which may already be present in the existing ontology.

2. Discover new concepts. To discover new concepts from a general corpus, the method proposes the following phases [Moldovan and Girju, 2000]. Firstly, *documents that contain the seed-concepts are retrieved* and stored before they are processed. Only the nouns are considered as candidate concepts by the method. Secondly, for each document, *sentences that contain the seed concepts are extracted.* Only the noun phrases are considered. Thirdly, *each of the previous sentences are POS-tagged and parsed.* There are two possible types of sentences to be selected. One of these is when the seed is the head noun of the phrase (the phrase would take the form [word, word, word, ..., seed]). The other possibility is when the seed is not the head noun of the phrase (the sentence would take the form [word, word, ..., seed, ..., word, word]). Then, after parsing all sentences, *new concepts are extracted*. To carry out this process, three main points have been proposed by the authors [Moldovan and Girju, 2001]:

a. Search the identified noun-sentences in the corpus for concepts stored in the existing ontology. The purpose is to find words that co-occur, one of which must be a existing concept of the ontology. A dictionary may be used to find more of these compound concepts.
b. For each co-location the process takes the words that modify a noun (that is the adjectives). Three types of adjectives are considered: descriptive (express an attribute of the modified noun), participial (derived from the participle form of verbs), and relational (related semantically or morphologically to the modifying noun). The method proposes to consider as *new concepts* only those that are formed with relational and participial adjectives and discard the descriptives. Based on this, the method proposes to take out all the adjectives from the previous step, with the following exceptions: when the adjective is part of a concept determined from the existing ontology or from a dictionary; or when the adjective is a relational or participial adjective.
c. *User validation.* The user inspects the list of the remaining noun phrases and decides whether to accept or decline each proposed concept.

3. *Discover lexical-syntactic patterns*. The main aim of this step is to discover semantic relations between concepts (between two new concepts or between a new one and one present in the existing ontology). The method then proposes to create a new corpus, different than the corpus used in the previous step. New noun-sentences are extracted from this corpus. The objective is to search for lexico-syntactical patterns covering the concepts of interest, extracted in the previous step, inside the new group of sentences.

4. *Discover new relations between concepts.* To carry out this process three elements are used: the new concepts discovered in Step 2, the group of noun-sentences extracted in that step, and the lexical syntactic patterns resulting from the Step 3. For each new concept the process tries to find all of the syntactic relations established in Step 3 in which the concept is involved. The relation is created between the two concepts linked by the syntactic relation. The validation of the process is performed by the user.

5. *Classification and integration*. In this step a new taxonomy is created for the newly acquired concepts. This new taxonomy will be integrated with the existing ontology using the relations discovered in the previous step between a new concept and other concepts in the existing ontology [Harabagiu and Moldovan, 2000].

To carry out the process and for evaluating the learning process, WordNet has been used. The method can be applied to the learning of an ontology from machine readable dictionaries.

4.3.2.3 A brief comparison with our approach

Most of previously described methodologies reveal a number of significant similarities with the RTV approach. More in detail, we combine the advantages of Pattern-based extraction methods that use heuristic rules to acquire domain terms and corpus statistics to measure their relevance, and Conceptual clustering methodologies, that perform better in shaping the hierarchy by smoothing dependencies on raw texts sources collocational distributions. Main features of our approach are the explicit use of terminological and semantic information automatically extracted from the domain corpus. Representing explicitly many-to-one mappings between linguistic forms and their corresponding meaning (i.e. relational concepts) allows to reduce the need for previous specific knowledge that seems mandatory in most other approaches that adopt or definitions or parallel corpora. A synthetic comparison among our and other approaches is synthesized in the table below:

Approach	Goal	Techniques	Other Ontologies required	Sources	Tool	Evaluation
RTV	Build ontology concepts and relations	NLP Analysis Term Extraction Conceptual clustering	NO ONLY IF AVAILABLE	Domain Texts WordNet	CHAOS Term-it …	Empirical measures and by expert
Agirre	Enrich concepts in existing ontologies	Statistics Clustering Topics signatures	YES	Domain Texts WordNet	N/A	User
Gupta	Build sublanguages in WordNet	NLP Techniques Term Extraction Techniques	YES	Domain texts WordNet	SubWN Engineering tool	Expert
Hahn	Learn new concepts	Concept hypothesis based on linguistic and conceptual quality labels Statistical approach	NO	Domain text	N/A	Empirical measures and by expert
Hwang	Elicit taxonomy	NLP Statistical approaches ML	NO	Domain text	N/A	Expert
Kietz	Learn concept and relation to enrich existing onto	NLP Statistical approach	YES	Domain specific and non domain specific texts WordNet Other ontologies	Text-to-Onto	User
Missikoff	Build Taxonomies	NLP Statistical approach ML	YES	Domain Texts WordNet	OntoLEarn	Expert
Moldovan	Enrich existing ontology	NLP analysis	YES	Domain Text Lexical Resources WordNet	N/A	Expert

From the methodological perspective, we can provide our final remarks:

• a shared methodology or method that guides the ontology learning process from text is missing in research community. General guidelines are provided by few approaches.

• All the approaches presented in the previous paragraphs base on natural language analysis techniques, and use a corpus that guide the overall acquisition process. Although the different approaches differ in the processing steps they apply to the domain corpus, most of them only use domain documents to learn new concepts and relations. Our idea of using an existing resource as WordNet to tune the lexical acquisition process to the specific domain is reflected only partially in Maedche and colleagues' work. Other approaches limit the attention to learn domain terminology from text collections without introducing a general knowledge of the language in the process. This could limit the quality of the material extracted and usually requires more efforts in the validation phase.

• The most common ontology used by many methods is WordNet, which is used as start up ontology enriched with new concepts or relations. WordNet si thus assumed as a *de facto* standard resource for general knowledge. Treating WordNet as an ontology instead of a lexical resource could strongly bias the way the final knowledge is represented. A more accurate analysis of WordNet qualities and pitfalls is missing and quite under-evaluated by almost all approaches.

• Our and all other methodologies require the participation of a domain expert to evaluate the final ontology and the accuracy of the learning process. This appears to be unavoidable given the nature of the task and the complexity of the involved inferences.

5. Current Technological Solutions

This sections will describe the possibilities and the limits of the implemented technologies that could be used for our purpose. It will describe each system including features such as:

- Programming languages used
- State of the API
- Purpose of the SW: e.g. commercial system or research prototype
- Performances (when available)
- Accessibility
- Ownership

5.1 IR Systems

In this paragraph a brief description of the main IR systems available will be presented (many on the information are directly taken by the web pages of the systems). Systems are divided in two classes: *On-site*, which refers to systems that can be used for local use on a specific set of document, and *Web-Oriented* system, that can be only used for searches on the Web.

5.1.1 On-site Systems

5.1.1.1 Jakarta Lucene

Jakarta Lucene is a high-performance, full-featured text search engine written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. While it accepts as input only textual files, it is possible to import other kind of documents (like PDF, PS, DOC, etc.) using specific external converters. Jakarta Lucene is a library, not a complete application.

Term, Field and Term Modifiers

A query is broken up into terms and operators. There are two types of terms: Single Terms and Phrases. A Single Term is a single word such as "test" or "hello". A Phrase is a group of words surrounded by double quotes such as "hello dolly".

Lucene supports fielded data. When performing a search it is possible either to specify a field, or to use the default field. The field names and default field is implementation specific. It can be searched any field by typing the field name followed by a colon ":" and then the term that is looked for. If no field is specified, default field is used

e.g.: field_name:word_or_phrase_to_search

Terms are case sensitive unless the lower case token filter is used during indexing and search. Lucene supports modifying query terms to provide a wide range of searching options.

To perform a single character wildcard search the "?" symbol is used; to perform a multiple character wildcard search the "*" symbol. It is not possible to use a * or ? symbol as the first character of a search.

To do a fuzzy search the tilde "~" symbol is used at the end of a Single word Term. Terms found by the fuzzy search will automatically get a boost factor of 0.2 . To do a proximity search the tilde "~" symbol is used at the end of a Phrase with the maximum distance in word.

To boost a term the caret "^" symbol is used with a boost factor (a number) at the end of the searched term. The higher the boost factor, the more relevant the term will be. By default, the boost factor is 1. Although, the boost factor must be positive, it can be less than 1 (i.e. .2)

Boolean operator

Boolean operators allow terms to be combined through logic operators. Lucene supports AND, "+", OR, NOT and "-" as Boolean operators (Note: Boolean operators must be ALL CAPS).

- The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching document if either of the terms exist in a document. The symbol || can be used in place of the word OR.
- The AND operator matches documents where both terms exist anywhere in the text of a single document. The symbol && can be used in place of the word AND.
- The "+" or required operator requires that the term after the "+" symbol exist somewhere in a the field of a single document.
- The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. The symbol ! can be used in place of the word NOT.
- The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

Groupping and Escape characters

Lucene supports using parentheses to group clauses to form sub queries. This can be very useful to control the boolean logic for a query. Lucene supports escaping special characters (currently + - && || ! ( ) { } [ ] ^ " ~ * ? : \) that are part of the query syntax.

5.1.1.2 Alkaline

The Alkaline Search Engine is an all-in-one index and search server. Alkaline searches exact words and word heuristics (parts of words) only. It does not do fuzzy or misspelled words search. Alkaline does not search phrases (yet). Alkaline can index (and search) documents in other format than HTML, such XML, PDF, RTF, MP3 using a command line filter.

The system can be used as an accessible HTTP/1.0 server: retrieval results are made available through an HTML page, that can be personalized using templates. Alkaline is a free service for non commercial use.

Simple Search

A simple search is done by typing a word. Searching for *"light"* will find all pages containing *"light"*, "*light*ning", "de*light*ed", etc. It will also find pages with *"Light"* and *"Lightning"* because searching is case-insensitive by default.

Searching multiple words is done by typing a sequence of words separated by spaces. Pages containing more words will be shown first in the results. Pages with words in the title or in meta tags will be more relevant. Case-sensitive search can be enabled by using a single capital letter inside a word. Entire words can be searched by using quotes.

Boolean Search

Boolean search allows to lookup pages containing some word and not containing other. To express the fact that a page must contain a word, a + sign must be placed in front of the word. To search for all pages not containing a word, a - sign should be used. Note that searching for -word or +foo -foo will produce no results.

Refined boolean search can be done by mixing a boolean expression with normal words. Words without sign (+ or -) are OR searched in document.

Meta Data Search

Meta search can be done by specifying a meta tag followed by a column, for example *author:foo*. Alkaline will index words, such as *price=234* , in a special manner. Consequently, it is possible to perform a numeric data search: *price<234* , *price=234* or *price>234*

Forcing Search Options

By default, Alkaline will choose a case-sensitive search when at least one upper-case letter is present in a word. To search all words case-sensitive, *opt:case* should be added to the search string. To search all words case-insensitive, *opt:insens* should be used.

To search all pages containing all words, *opt:and* should be added to the search string. The default behaviour of Alkaline is to search all pages containing any of the words and producing best results first.

To force searching of whole words only, *opt:whole* should be added to the search string. The default behaviour of Alkaline is to do partial matches.

It is of course possible to specify more than one such option by separating them by commas or by adding multiple opt: entries to the search string, for example: *foo opt:whole,case* will return all pages containing the exact word "foo".

5.1.1.3 phpDig

PhpDig indexes all words of a document, excepting small words (less than 3 letters) an common words, that are defined in a text file. Alone numbers are not indexed, but those included in words. Underscores make part of a word. Occurrences of a word in a document is saved. Words in the title can have a more important weight in ranking results. PhpDig tries to read a robots *txt* file at the server root. It searches meta robots tags too. PhpDig can't perform an exact expression search. Options are:

- An AND operator is applied between each search key, no OR operator is performed ;
- Putting a '-' sign before a word excludes it from the search results. No document containing this word would be displayed ;
- Search is case-insensitive and accent-insensitive. In the other hand, results highlighting is accent-sensitive.

To operate, phpDig needs a web server with PHP support. Retrieval result are made available through an HTML page. The system is released under GNU/GPL licence.

5.1.1.4 Managing Gigabytes

The MG (Managing Gigabytes) system is a collection of programs which comprise a full-text retrieval system. A full-text retrieval system allows one to create a database out of some given documents and then do queries upon it to retrieve any relevant documents. It is "full-text" in the sense that every word in the text is indexed and the query operates only on this index to do the searching.

For example, one could have a database on the book, "Alice in Wonderland." A document could be represented by each paragraph in the book. Having built up the "Alice" database, one could do queries such as "cat Alice grin" and retrieve any paragraphs which match the query. The matching could either be boolean, that is the retrieved paragraphs contain a boolean expression of the query terms e.g. "cat Alice grin"; or the matching could be ranked i.e. the most relevant documents to the query in relevance order, using some standard heuristic measure.

A query consists of two parts. One part is a Boolean or ranked query that identifies documents. The second part is a post-processing pattern matching operation. Any text between the first speech mark (") and the last speech mark (") is considered to be a post-processing pattern.

It is possible to perform four types of queries: boolean query, ranked query , approx-ranked query , *docnums*.

Boolean query

It permits to specify a boolean query using the AND(&), OR(|)NOT(!) operators.

Ranked and approx-ranked query

They are queries ranked by the cosine measure. Approx-ranked uses only the low-precision document lengths, and therefore only produces an approximation to full cosine ranking.

Docnums

Allows the entry of document numbers. Multiple numbers separated by spaces may be specified, or ranges separated by hyphens.

5.1.1.5 A schematic comparison of On-site Systems functionalities

	Lucene	**Alkaline**	**phpDig**	**MG**
Language	Java	C++	PHP	C(unix)
Phrase	in quotes	no	no	no
AND	yes	by option	default	yes
OR	default	default	no	yes
NOT	yes	no	no	yes
Include(+)	yes	yes	no	no
Exclude(-)	yes	yes	yes	no
Wildcard	yes[1]	no	no	no

Near	yes	no	no	no
Fuzzy	yes	no	no	no
Boost	yes	no	no	no
Field(Meta Data)	yes	yes	no	no
Exact match	default	by quote	yes	yes
Case-sensitive	yes	yes[3]	no	no
Groupping	yes	no	no	yes
Numeric Data[0]	no	yes	no	no
Non Plain Text Doc.	yes[2]	yes[4]	no	no

NOTES:

0) It permits to perform check on numeric values (e.g. birth_date > 1900)

1) no strart word;

2) by external program) or Filter class

3) by option or by using a single capital letter inside a word

4) external program (alcuni già disponibili)

5.1.2 Web-oriented Information Retrieval Systems

Search engine technology has had to scale dramatically to keep up with the growth of the web. In 1994, one of the first web search engines, the World Wide Web Worm (WWWW) had an index of 110,000 web pages and web accessible documents. At the end of nineties, the top search engines claim to index from 2 million (WebCrawler) to 100 million web documents (from Search Engine Watch). Today, a comprehensive index of the Web contains over a billion documents. At the same time, the number of queries search engines handle has grown incredibly too. In 1994, the World Wide Web Worm received an average of about 1500 queries per day. In 1997, Altavista claimed it handled roughly 20 million queries per day. With the increasing number of users on the web, and automated systems which query search engines, today top search engines handle hundreds of millions of queries per day. Such systems should address many of the problems, both in quality and scalability, introduced by scaling search engine technology to such extraordinary numbers. Fast crawling technology is needed to gather the web documents and keep them up to date. Storage space must be used efficiently to store indices and, optionally, the documents themselves. The indexing system must process hundreds of gigabytes of data efficiently. Queries must be handled quickly, at a rate of hundreds to thousands per second.

These tasks are becoming increasingly difficult as the Web grows. However, hardware performance and cost have improved dramatically to partially offset the difficulty. There are, however, several notable exceptions to this progress such as disk seek time and operating system robustness.

5.1.2.1 Google

5.1.2.1.1 Google Architecture Overview

In Google, the web crawling (downloading of web pages) is done by several distributed crawlers. There is a URL-server that sends lists of URLs to be fetched to the crawlers. The web pages that are fetched are then sent to the store-server. The storeserver then compresses and stores the web pages into a repository. Every web page has an associated ID number called a docID which is assigned whenever a new URL is parsed out of a web page. The indexing function is performed by the indexer and the sorter. The indexer performs a number of functions. It reads the repository, uncompresses the documents, and parses them. Each document is converted into a set of word occurrences called hits. The hits record the word, position in document, an approximation of font size, and capitalization. The indexer distributes these hits into a set of "barrels", creating a partially sorted forward index. The indexer performs another important function. It parses out all the links in every web page and stores important information about them in an anchors file. This file contains enough information to determine where each link points from and to, and the text of the link.

The URL-resolver reads the anchors file and converts relative URLs into absolute URLs and in turn into docIDs. It puts the anchor text into the forward index, associated with the docID that the anchor points to. It also generates a database of links which are pairs of docIDs. The links database is used to compute PageRanks for all the documents.

The sorter takes the barrels, which are sorted by docID, and resorts them by wordID to generate the inverted index. This is done in place so that little temporary space is needed for this operation. The sorter also produces a list of wordIDs and offsets into the inverted index. A program called DumpLexicon takes this list together with the lexicon produced by the indexer and generates a new lexicon to be used by the searcher. The searcher is run by a web server and uses the lexicon built by DumpLexicon together with the inverted index and the PageRanks to answer queries.

The Google search engine has two important features that help it produce high precision results. First, it makes use of the link structure of the Web to calculate a quality ranking for each web page. This ranking is called PageRank and is described later in detail . Second, Google utilizes link to improve search results.

5.1.2.1.2 PageRank: Bringing Order to the Web

The citation (link) graph of the web is an important resource that has largely gone unused in existing web search engines. We have created maps containing as many as 518 million of these hyperlinks, a significant sample of the total. These maps allow rapid calculation of a web page's "PageRank", an objective measure of its citation importance that corresponds well with people's subjective idea of importance. Because of this correspondence, PageRank is an excellent way to prioritize the results of web keyword searches. For most popular subjects, a simple text matching search that is restricted to web page titles performs admirably when PageRank prioritizes the results. For the type of full text searches in the main Google system, PageRank also helps a great deal. In past researches, academic citation literature has been applied to the web, largely by counting citations or backlinks to a given page. This gives some approximation of a page's importance or quality. PageRank extends this idea by not counting links from all pages equally, and by normalizing by the number of links on a page. PageRank is defined as follows:

Let us assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. Be C(A) defined as the number of links going out of page A. The PageRank of a page A is given as follows:

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.

PageRank or *PR(A)* can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web. Also, a PageRank for 26 million web pages can be computed in a few hours on a medium size workstation.

PageRank can be thought of as a model of user behaviour. There is assumed a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the random surfer visits a page is its PageRank. And, the *d* damping factor is the probability at each page the "random surfer" will get bored and request another random page. One important variation is to only add the damping factor *d* to a single page, or a group of pages. This allows for personalization and can make it nearly impossible to deliberately mislead the system in order to get a higher ranking. Another intuitive justification is that a page can have a high PageRank if there are many pages that point to it, or if there are some pages that point to it and have a high PageRank. Intuitively, pages that are well cited from many places around the web are worth looking at. Also, pages that have perhaps only one citation from something like the Yahoo! homepage are also generally worth looking at. If a page was not high quality, or was a broken link, it is quite likely that Yahoo's homepage would not link to it. PageRank handles both these cases and everything in between by recursively propagating weights through the link structure of the web.

5.1.2.1.3 Anchor Text

The text of links is treated in a special way in the Google search engine. Most search engines associate the text of a link with the page that the link is on. In addition, Google associates it with the page the link points to. This has several advantages. First, anchors often provide more accurate descriptions of web pages than the pages themselves. Second, anchors may exist for documents which cannot be indexed by a text-based search engine, such as images, programs, and databases. This makes it possible to return web pages which have not actually been crawled. Note that pages that have not been crawled can cause problems, since they are never checked for validity before being returned to the user. In this case, the search engine can even return a page that never actually existed, but had hyperlinks pointing to it. However, it is possible to sort the results, so that this particular problem rarely happens.

This idea of propagating anchor text to the page it refers to was implemented in the World Wide Web Worm especially because it helps search non-text information, and expands the search coverage with fewer downloaded documents. Anchor propagation is used mostly because anchor text can help provide better quality results. Using anchor text efficiently is technically difficult because of the large amounts of data which must be processed.

5.1.2.1.4 Other Features

Aside from PageRank and the use of anchor text, Google has several other features. First, it has location information for all hits and so it makes extensive use of proximity in search. Second, Google keeps track of some visual presentation details such as font size of words. Words in a larger or bolder font are weighted higher than other words. Third, full raw HTML of pages is available in a repository.

5.1.2.2 Altavista

5.1.2.2.1 General information

AltaVista can be identified as a search engine, but it's actually a hybrid. It gathers results from its own index of 300,000,000 pages; the directory from LookSmart; listings from Overture.com and Internet Keywords from RealNames. AltaVista recently redesigned its site to reflect its new focus on search. Many of AltaVista's portal services, such as AltaVista Live, Channels and so on, are no longer available. AltaVista's new look is aimed at making searching easier. AltaVista is one of the oldest search engines on the web, having been created back in 1995 by Digital Equipment Corporation. AltaVista was acquired by Compaq in 1998 and by its current owner, CMGI, in August of 1999. According to AltaVista's help page, AltaVista's name was conceived in a laboratory after the words Alto and Visto were accidentally placed together on a white board. The name "AltaVista" means "view from above." Throughout its years of existence AltaVista has led the search engine industry in many areas. It was one of the largest and fastest search engines until FAST search and Google were developed. It was also the first search engine to offer translation services and localize its site for Chinese, Japanese and Korean searchers. AltaVista continues to maintain its strong international presence, and is currently available for natives of over 20 countries.

5.1.2.2.2 AltaVista's Search Technology

AltaVista analyzes keywords and link popularity to rank pages in order of perceived relevance. It's a full-text index, meaning it indexes every word in a page.

Crawl depth

In a perfect world, submitting a main page to a search engine would insure that its spider would visit every other page in your site. Unfortunately, most search engines only crawl down to a certain directory level to find your pages. For instance, supposing that the two following pages come from the same site:

Page 1: www.abcxyz.com/abc/alphabet.html
Page2 : www.abcxyz.com/abc/def/xyz/new/old/archive/alphabet.html

because Page 1 is one directory (folder) down, and Page 2 is six directories down, most search engines will be more likely to index Page 1 than Page 2.

AltaVista's help pages state that its spider considers pages placed closer to the root directory to be more important. Also, the spider "may not venture deeper than three, four, or five directory levels."

Researches shows that AltaVista does rank pages contained in top directories higher than those buried deeper. We tested both popular and rarely used keywords and found that top ranked results are either for a domain name or an index page. For example, chances are that www.xyz.com or www.xyz.com/news.html will rank better than www.xyz.com/news/more/US/another.html

Results ranking algorithm

AltaVista pays most attention to the following elements, which will be discussed later in more depth: HTML titles, keyword placement, link popularity, link text.

Indexing criteria - page elements

AltaVista's help page claims that only the **first two occurrences of a keyword are indexed.** AltaVista prioritizes the following factors: Keywords in the title, Keyword prominence, Keyword proximity, Keywords in the link text. Keyword density and frequency, which are very important for other search engines, make little difference to AltaVista because it only indexes the first two occurrences of a keyword.

AltaVista considers the content of HTML titles to be the most important criterion in scoring your pages. Though AltaVista can index titles over 400 characters long, don't make yours that long -- only about 70 characters will be displayed in the results. Also, short and focused titles seem to score better. Though it indexes meta tags, considering them to be regular text, AltaVista claims it doesn't give them priority over HTML titles and other text. It's generally believed that AltaVista gives some weight to keywords in filenames and URL names. AltaVista indexes ALT tags: they should contain more than the image's description. They should include keywords, especially if the image is at the top of the page.

There's been some debate about how long doorway pages for AltaVista should be. Some researches suggest that short pages rank higher, while others argue that long pages are the way to go. According to AltaVista's help section, it prefers long and informative pages. AltaVista has the ability to index frames, but it sometimes indexes and links to pages intended only as navigation. Moreover, its engine is one of the few to index ASP and other dynamic pages, including CFML, CGI generated and URLs with question marks.

Indexing criteria -- factors external to page

AltaVista does not use click tracking for regular searches. It does track clicks when cost-per-click deals are involved, according to a spokesperson. Inktomi is one of the many search engines that considers link popularity an important ranking criterion. When evaluating link popularity, Inktomi considers the following factors: Number of links, Descriptions that referring sites use, Size of the linking sites. AltaVista also uses theme scoring, evaluating how keywords are consistent throughout the entire site.

Indexing criteria -- elements not indexed

AltaVista does not index the following: Comment tags, Dynamic pages, Text embedded in images (except for ALT tags), Password-protected sites, XML and Acrobat files, Database-driven sites

To filter out as many similar pages as possible, AltaVista's spider follows a two-step procedure. First it checks links. If it finds that two pages have the same links, a red flag goes up. Now the spider will compare the two pages' text. AltaVista knows that a webmaster optimizing for certain keywords will repeat those keywords several times in a page, so it scans both pages to see if they repeat the same words. If the two pages match, they will be removed, and the site can be banned.

Because AltaVista has some natural-language analysis built in, page content must be arranged into logical sentences.

5.2 Natural Language Processing Systems

5.2.1 Robust Syntactic Analysers

5.2.1.1 Gate

GATE has been built over the past eight years in the Sheffield NLP group. The system has been used for many language processing projects; in particular for Information Extraction in many languages. The system supports the full lifecycle of language processing components, from corpus collection and annotation through system evaluation. GATE is funded by the EPSRC and the EU.

GATE [Cunningham, 2002] is an architecture, a framework and a development environment for LE (Language Engineering). As an architecture, it defines the organisation of an LE system and the assignment of responsibilities to different components, and ensures that the component interactions satisfy the system requirements. As a framework, it provides a reusable design for an LE software system and a set of prefabricated software building blocks that language engineers can use, extend and customise for their specific needs. As a development environment, it helps its users to minimise the time they spend building new LE systems or modifying existing ones, by aiding overall development and providing a debugging mechanism for new modules. Because GATE has a component based model, this allows for easy coupling and decoupling of the processors, thereby facilitating comparison of alternative configurations of the system or different implementations of the same module (e.g., different parsers). The availability of tools for easy visualisation of data at each point during the development process aids immediate interpretation of the results.

The GATE framework comprises a core library (analogous to a bus backplane) and a set of reusable LE modules. The framework implements the architecture and provides (amongst other things) facilities for processing and visualising resources, including representation, import and export of data. The reusable modules provided with the backplane are able to perform basic language processing tasks such as POS tagging and semantic tagging. This eliminates the need for users to keep recreating the same kind of resources, and provides a good starting point for new applications. The modules are described in more detail in Section 4. Applications developed within GATE can be deployed outside its Graphical User Interface (GUI), using programmatic access via the GATE API (see http://gate.ac.uk). In addition, the reusable modules, the document and annotation model, and the visualisation components can all be used independently of the development environment.

GATE components may be implemented by a variety of programming languages and databases, but in each case they are represented to the system as a Java class. This class may simply call the underlying program or provide an access layer to a database; alternatively it may implement the whole component. In the rest of this section, we show how the GATE infrastructure takes care of the resource discovery, loading, and execution, and briefly discuss data storage and visualisation. components are one of three types: Language Resources (LRs) represent entities such as lexicons, corpora or ontologies; Processing Resources (PRs) represent entities that are primarily algorithmic, such as parsers, generators or *ngram* modellers; Visual Resources (VRs) represent visualisation and editing components that participate in GUIs. These resources can be local to the user’s machine or remote (available via HTTP), and all can be extended by users without modification

to GATE itself. One of the main advantages of separating the algorithms from the data they require is that the two can be developed independently by language engineers with different types of expertise, e.g. programmers and linguists. Similarly, separating data from its visualisation allows users to develop alternative visual resources, while still using a language resource provided by GATE. Collectively, all resources are known as CREOLE (a Collection of REusable Objects for Language Engineering), and are declared in a repository XML file, which describes their name, implementing class, parameters, icons, etc. This repository is used by the framework to discover and load available resources. A parameters tag describes the parameters which each resource needs when created or executed. Parameters can be optional, e.g. if a document list is provided when the corpus is constructed, it will be populated automatically with these documents. When an application is developed within GATE's graphical environment, the user chooses which processing resources go into it (e.g. tokenizer, POS tagger), in what order they will be executed, and on which data (e.g. document or corpus). The execution parameters of each resource are also set there, e.g. a loaded document is given as a parameter to each PR. When the application is run, the modules will be executed in the specified order on the given document. The results can be viewed in the document viewer/editor.

Provided with GATE is a set of reusable processing resources for common NLP tasks. (None of them are definitive, and the user can replace and/or extend them as necessary.) These are packaged together to form ANNIE, A Nearly- New IE system, but can also be used individually or coupled together with new modules in order to create new applications. For example, many other NLP tasks might require a sentence splitter and POS tagger, but would not necessarily require resources more specific to IE tasks such as a named entity transducer. The system is in use for a variety of IE and other tasks, sometimes in combination with other sets of application-specific modules.

ANNIE consists of the following main processing resources: tokenizer, sentence splitter, POS tagger, gazetteer, finite state transducer (based on GATE's built-in regular expressions over annotations language [Cunningham et al., 2002]), orthomatcher and coreference resolver. The resources communicate via GATE's annotation API, which is a directed graph of arcs bearing arbitrary feature/value data, and nodes rooting this data into document content (in this case text).

The tokenizer splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types (e.g. with an initial capital, all upper case, etc.). The aim is to limit the work of the tokeniser to maximise efficiency, and enable greater flexibility by placing the burden of analysis on the grammars. This means that the tokenizer does not need to be modified for different applications or text types. The sentence splitter is a cascade of finite state transducers which segments the text into sentences. This module is required for the tagger. Both the splitter and tagger are domain and application-independent. The tagger is a modified version of the Brill tagger, which produces a part-of-speech tag as an annotation on each word or symbol. Neither the splitter nor the tagger are a mandatory part of the NE system, but the annotations they produce can be used by the grammar (described below), in order to increase its power and coverage. The gazetteer consists of lists such as cities, organisations, days of the week, etc. It not only consists of entities, but also of names of useful indicators, such as typical company designators (e.g. 'Ltd.'), titles, etc. The gazetteer lists are compiled into finite state machines, which can match text tokens. The semantic tagger consists of handcrafted rules written in the JAPE (Java Annotations Pattern Engine) language [Cunningham et al., 2002], which describe patterns to match and annotations to be created as a result. JAPE is a version of CPSL (Common Pattern Specification Language) [Appelt, 1996], which provides finite state transduction over annotations based on regular expressions. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules, and which run sequentially. Patterns can be specified by describing a specific text string, or annotations previously created by modules such as the tokenizer, gazetteer, or document format analysis. Rule prioritisation (if activated) prevents multiple assignment of annotations to the same text string. The orthomatcher is another optional module for the IE system. Its primary objective is to perform co-reference, or entity tracking, by recognising relations between entities. It also has a secondary role in improving named entity recognition by assigning annotations to previously unclassified names, based on relations with existing entities. The coreferencer finds identity relations between entities in the text. For more details see [Dimitrov, 2002].

5.2.1.2 CHAOS

CHAOS has been already extensively described in Sec.4.2.1.3. The syntactic processor is seen and implemented as a cascade of processing modules (see Fig. 18) applying different rules to the input with different techniques ranging from string matcher to finite state automaton, to discontinuous grammar applicators.

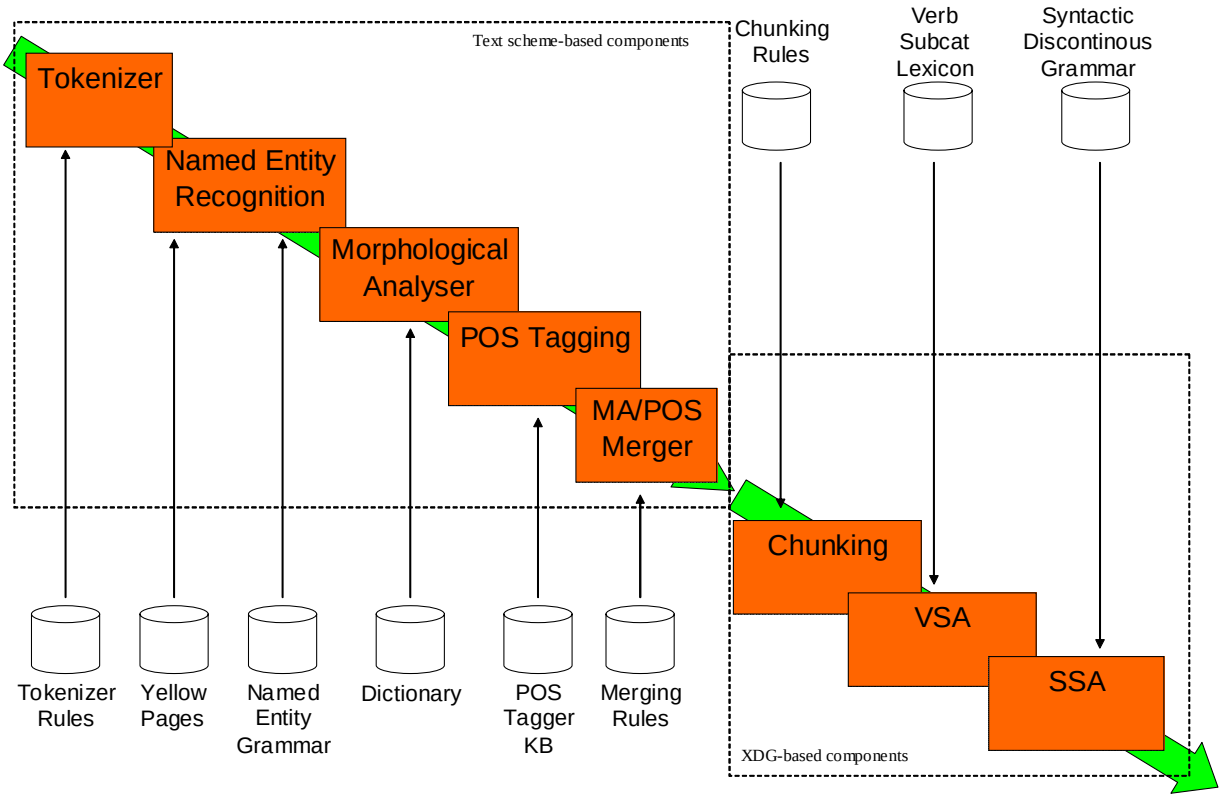


Fig. 18 The Robust Syntactic Parser Modules

The syntactic parser then applies the following cascade of processing modules:

- a ***tokenizer***, matching words from character streams;
- a **named entity recogniser** matching simple named entities existing in catalogues (***yellow pages***) and complex named entities according to special purpose grammars (***NE Grammar***);
- a **morphologic analyzer** that attaches (possibly ambiguous) syntactic categories and morphological interpretations for each word using a ***Dictionary***
- a rule-based **part-of-speech (POS) tagger** and a POS disambiguation module that resolves potential con among the results of the POS tagger and the morphologic analyzer;
- a **chunker** that finds the phrase kernels using a specific grammar (***Chunking Rules***)
- a **shallow syntactic analyser (SSA)** that builds a chunk-based representation of the input text and includes major grammatical dependencies among chunk heads.

In the analysing process, the syntactic information is added to the underlying annotation scheme. The basic annotation scheme is the extended dependency graph (XDG) that builds on the basic notion of lexical handle.

The processing chain is divided in two halves according to the annotation underlying annotation scheme:

- text scheme annotation based modules
- xdg-based modules

5.2.2 Tools for Ontology Learning from text

In this section, we will summarize the most relevant tools used for ontology learning from text. Most of the presented system descriptions have been partially excerpted from the excellent survey of ontology learning methods and techniques delivered by the OntoWeb project Consortium [Gomez and Manzano, 2003].

5.2.2.1 ASIUM

ASIUM [Faure and Nedellec, 1999] is an acronym for "Acquisition of SemantIc knowledge Using Machine learning methods". It has been developed at the Computer Science Research Laboratory (LRI) in the University of Paris-Sud. The main aim of ASIUM is to help the expert in the acquisition of semantic knowledge from technical texts using syntactic analysis. ASIUM takes as input French texts in natural language and associates a rate of appearance in the text. The learning method is based on conceptual and hierarchical clustering. Basic clusters are formed by words that occur with the same verb after the same preposition [Faure and Nedellec, 1998]. The tool uses a metric to compute the semantic similarity between clusters, which is used by the ontologist to decide if a new concept is created. Clusters are

successively aggregated by the conceptual clustering method to form the concepts of the ontology. The ontologist defines a minimum threshold for gathering clusters into concepts. The learning is intertwined and validated by the ontologist.

The tool follows two steps to achieve its performances. The first one, the *Factorisation* (conceptualisation): The head words are associated with its frequency of appearance in the text in order to calculate the distance among concepts. Those who appear in similar contexts are added, by means of an algorithm of conceptual clustering to form the concepts of the ontology. For this purpose, a technique to estimate the semantic similarity among concepts has been used [Liu 1996], [Bisson 1994] and [Bisson 1992]. The second step is *Clustering* (ontology building). Due to the fact that a hierarchy would be too much restricted to represent the complexity of the ontology in many domains, the authors have adopted the skill of pyramidal clustering. The ontology is constructed level-by-level.

Goal: to find taxonomic relations among terms in natural language texts in French without annotating them

Learning technique: conceptual clustering

Method followed for ontology learning: two steps: (1) Factorisation (conceptualisation) and (2) Clustering (ontology building)

User/Expert interaction: the user participation is needed not only to tag the new concepts but also to control the generality level of verbal frames, in order to refine the learned clusters and to handle noise. Each of the conceptualisation and clustering steps are validated by the expert. If a cluster or a pyramid level is updated each process starts again

Sources used by the method: ASIUM uses texts syntactically analysed by SILEX

SW Architecture: not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool:

http://www.lri.fr/~faure/Demonstration/Presentation_Demo.html

5.2.2.2 CORPORUM-Ontobuilder

CORPORUM-Ontobuilder has been developed by CognIT. Its main aim is to be able to extract ontologies (mainly taxonomies) from natural language texts. The tool uses several linguistic techniques that drive the analysis and information extraction functionalities. CORPORUM-Ontobuilder extracts information from structured and unstructured documents using the tools named OntoWrappper [Engels, 2001b] and OntoExtract [Engels, 2001a]. Ontowrapper extracts information from on-line resources (e.g. names, email addresses, telephone numbers, etc.) and OntoExtract obtains taxonomies from natural language texts.

Ontoextract is also able (through semantic analysis of the content of web pages) to provide initial concept taxonomies, to refine existing concept taxonomies (include more concepts), to find relations between key terms in documents and to find concept instances within documents. Concept taxonomies are created in RDF(S).

Goal: to extract an initial ontology and refine it

Learning technique: linguistic and semantic techniques

Method followed for ontology learning: own method

User/Expert interaction: not necessary

Sources used by the method: text

SW Architecture: not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool: http://ontoserver.cognit.no

5.2.2.3 LTG (Language Technology Group) Text Processing Workbench

LTG (Language Technology Group) Text Processing Workbench**12** [Mikheev and Finch, 1997] has been developed by Language Technology Group (LTG) in the University of Edinburgh. It is a set of computational tools for uncovering internal structure in natural language texts written in English. The main idea behind the workbench is the independence of the text representation and text analysis.

In LTG, ontology learning is performed in two sequential steps: representation and analysis. At the representation step, the text is converted from a sequence of characters to features of interest by means of annotation tools. At the analysis step, those features are used by statistics-gathering tools and inference tools for finding significant correlations in the texts. The analysis tools are independent from a particular assumption on the nature of the feature-set and work on the abstract level of feature-elements which are represented as SGML items. The workbench is being used both for lexicographic purposes and for statistical language modelling.

It supports an incremental process of corpus analysis starting from a rough automatic extraction and organization of lexical-semantic regularities and ending with a computer-supported analysis of extracted data and a semi-automatic refinement of obtained hypotheses. To do this the workbench uses methods from computational linguistics, information retrieval and knowledge engineering.

Goal: to discover internal structure of texts in natural language

Learning technique: statistic Inference

Method followed for ontology learning: two steps: representation (annotation of text in SGML) and analysis (the annotated text is analysed by the statistical inference tools)

User/Expert interaction: N/A

Sources used by the method: texts in natural language

SW Architecture not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool:

http://www.ltg.ed.ac.uk/%7Emikheev/workbench.html

5.2.2.4 Mo'K Workbench

Mo'K Workbench [Bisson et al., 2000] is a configurable workbench that supports the semiautomatic construction of ontologies from a corpus using different conceptual clustering methods. it has been

developed by INRIA, LRI Univ. Paris-South13. Mo'k assists ontologists in the exploratory process of defining the most suitable learning method. In this sense, Mo'K supports the elaboration, comparison, characterization and evaluation of different conceptual clustering methods. It also permits fine-grained definitions of similarity measures and class construction operators, easing the task of method instantiation and configuration.

The learning process proposed by this workbench takes a corpus as input. No additional knowledge is used to label the input, to guide the learning, or to validate the learned results. Trough NLP techniques, the tool extracts from the corpus a list of triplets. A triplet is composed by a verb, a word and a syntactic role of this word in a sentence. Using the triplets, Mo'K calculates the number of occurrences of each one. Triplets with low number of occurrences or too many occurrences are removed from that list. Finally, Mo'K calculates the semantic distance between the triplets in the previous list to form conceptual clusters.

Goal: to obtain concept taxonomy from domain tagged text

Learning technique: conceptual clustering

Method followed for ontology learning: following Harris hypotheses, syntactic relations are used among words to derive semantic relations

User/Expert interaction: not available in papers

Sources used by the method: domain tagged texts. In addition, there can be used other ontologies, dictionaries and other similar resources

SW Architecture not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool: not available in papers

5.2.2.5 OntoLearn Tool

The OntoLearn tool [Velardi et al., 2002] aims to extract relevant domain terms from a corpus of text, relate them to appropriate concepts in a general-purpose ontology, and to detect relations among the concepts. To carry out these tasks, natural language analysis and machine learning techniques are used. The tool has been tested inside the Harmonise European project.

OntoLearn extracts terminology from a corpus of domain text, such as specialized Web sites. The system then filters the terms using natural language processing and statistical techniques that perform comparative analysis across different domains, or contrasting corpora. This analysis identifies terminology that is used in the target domain but is not seen in other domains. Next, it uses the WordNet and SemCor lexical knowledge bases to perform semantic interpretation of the terms. The tool then relates concepts according to taxonomic (kind-of) and other semantic relations, generating a domain concept forest. For this purpose, WordNet and a rule-based inductive-learning method have been used to extract such relations. Finally, OntoLearn integrates the domain concept forest with WordNet to create a pruned and specialized view of the domain ontology. The validation of the process is performed by an expert.

Goal: to enrich a domain ontology with concepts and relations

Learning technique: NLP and machine learning (semantic interpretation)

Method followed for ontology learning: OntoLearn method: semantic interpretation

User/Expert interaction: in the evaluation phase

Sources used by the method: text

SW Architecture: not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool: not available in papers

5.2.2.6 SOAT: a Semi-Automatic Domain Ontology Acquisition Tool

SOAT tool [Wu and Hsu, 2002] allows a semi-automatic domain ontology acquisition from a domain corpus. The main objective of the tool is to extract relationships from parsed sentences based on applying phrase-rules to identify keywords with strong semantic links like hyperonym or synonym. The acquisition process is based on using InfoMap [Hsu et al., 2001], a knowledge representation framework, that integrates linguistic, commonsense, and domain knowledge. InfoMap has been developed to perform natural language understanding, and to capture the topic words, usually pairs of noun and verb, or noun and noun in a sentence. InfoMap has two major relations between concepts: taxonomic relations (category and synonym) and non-taxonomic (attribute and event).

The acquisition process carry out by SOAT includes to collect domain keywords and find the relationships among them. To perform this activity, a set of rules has been defined for extracting keywords from a sentence related to concepts in InfoMap with a strong semantic relation between them.

The tool receives as input a domain corpus with the POS-tag. A keyword, usually the name of the domain, is selected in the corpus as root. Then, with this keywords, the process aims to find a new related keyword with the previous by means of applying the extraction rules and add the new keyword into the ontology according to the rules and the structure fixed in InfoMap. This new keyword is now taken as root to repeat the process during a determined number of times or until being impossible to find a new related keyword. The user intervention is necessary to verify the results of the acquisition and to refine and update the extraction rules. The restrictions of SOAT is that the quality of the corpus must be very high in the sense that the sentences must be accurate and enough to include most of the important relationships to be extracted.

Goal: acquisition of relationships using a predefined knowledge representation framework

Learning technique: phrase-patterns, defined as set of linguistic templates

Method followed for ontology learning: not specified, the tool follows its own method

User/Expert interaction: information not available in papers

Sources used by the method: text

SW Architecture: not available in papers

Import and export facilities to ontology languages: none

URL where you can find information about the method or tool:

http://www.iis.sinica.edu.tw/IASL/en/index.htm

5.2.2.7 SubWordNet Engineering Process tool

In [Gupta et al., 2002] is presented an architecture to interactively acquire and maintain sublanguage WordNets follows the Iterative SubWordNet Engineering process approach explained in this deliverable. The architecture builds upon WordNet semantic structure and includes integrated capabilities for concept element discovery, concept identification, and concept maintenance. The architecture to perform each of these capabilities has been modularised into three layers: the graphical user interface (GUI) layer, the process layer, and the data layer.

The *concept discovery capability* includes a Concept Discovery Workbench, a Concept Discovery Engine, and a Discovered Concepts Database modules. This module provides a GUI that allows users to select the documents, manipulate discovered concept elements, and also provides summary distributional information of words and phrase for assisting users. It also includes several NLP components to discover relations using collocation statistics, lexical patterns, etc.

- The *concept identification capability* includes Concept Identification Workbench, Concept Identification Engine, and the Identified Concepts Database modules. This module has been designed to support concept identification, phrase clustering, and to establish concordance between concept nodes and WordNet synsets.
- For the *concept maintenance capabilities* includes the SubWordNet Editor as the GUI layer, the SubWordNet Editor Engine as the process layer, and the SubWordNet Database as the data layer.

Goal: to build Sublanguage WordNets

Learning technique: different NLP techniques and several statistical approaches

Method followed for ontology learning: approach for iterative SubWordNet engineering process

User/Expert interaction: whole process

Sources used by the method: textual documents

SW Architecture not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool:

http://www.aic.nrl.navy.mil/~aha/cbr/luikm.html

5.2.2.8 SVETLAN'

SVETLAN' [Chaelandar and Grau, 2000] has been developed by the LIR research group of the HMC Department of the University of Paris-Sud16. SVETLAN' is a domain-independent tool that creates clusters from words appearing on texts. Its learning method is based on a distributional approach: nouns playing the same syntactic role in sentences with the same verb are aggregated in the same class.

The learning process has the following steps: syntactic analysis, aggregation and filtering. In the syntactic analysis step, the tool retrieves sentences of the original texts in order to find the verb inside the sentence. This is based on the assumption that verbs allow categorizing nouns. The output of this step is a list of triplets that contain the verb, the noun and the syntactic relation between them. The aggregation step constructs groups of nouns with similar meanings. The filtering step is based on the weight of the nouns inside their classes. It removes nouns from these groups if they are not very relevant for the class. The threshold is established by the ontology developer. The process doesn't require validation and it is completely independent from the ontology developer.

Goal: to build a hierarchy of concepts

Learning technique: conceptual clustering

Method followed for ontology learning: syntactic analysis, clustering and filtering

User/Expert interaction: validation

Sources used by the method: French texts in natural language

SW Architecture: not available in papers

Import and export facilities to ontology languages: not available in papers

URL where you can find information about the method or tool:

http://www.limsi.fr/Individu/gael/ManuscritThese/

5.2.2.9 Text-To-Onto

Text-To-Onto [Maedche and Volz, 2000] [Maedche and Staab, 2003] has been developed at the AIFB Institute in the University of Karlsruhe19. The tool integrates an environment for building domain ontologies from an initial core ontology. It also discovers conceptual structures from different German sources using knowledge acquisition and machine learning techniques. Text-To-Onto has implemented some techniques for ontology learning from free text and semi-structured text, dictionaries, legacy ontologies and databases. The result of the learning process is a domain ontology that contains domain specific and domain-independent concepts. Domain-independent concepts are withdrawn to better adjust the vocabulary of the domain ontology. The result of the process is a domain ontology that only contains

domain concepts learnt from the input sources related before. The whole process is supervised by the ontologists. This is a cyclic process, in the sense that it is possible to refine and complete the ontology if we repeat the process.

Goal: to find taxonomic and non-taxonomic relations

Learning technique: statistical approach, pruning techniques and association rules

Method followed for ontology learning: method based on Srikant's and Agrawal [Srikant and Agrawal, 1995] algorithm

User/Expert interaction: validation

Sources used by the method: machine readable dictionaries and other ontologies

SW Architecture is part of KAON tool suite

Import and export facilities to ontology languages: imports from DAML+OIL, RDF(S), and exports to DAML+OIL and RDF(S)

URL where you can find information about the method or tool:

http://ontoserver.aifb.unikarlsruhe.de/texttoonto/

6. References

[Abney, 1996] Steven AbneyPart-of-speech tagging and partial parsing. In G.Bloothooft K.Church, S.Young, editor, *Corpus-based methods in language and speech*. Kluwer academic publishers, Dordrecht, 1996.

[Adriaans, 1992] Pieter Willem Adriaans.Language Learning from a Categorial Perspective. *Academisch Proefschrift*, Universiteit van Amsterdam, 1992

[Adriaans and Zantinge, 1996] Adriaans P, Zantinge D. Data Mining. Addisson-Wesley, 1996.

[Agirre and Rigau, 1996] Eneko Agirre and German Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics*, 1996.

[Agirre et al., 2000] Agirre, E., Ansa, O., Hovy, E., and Martinez, D. Enriching very large ontologies using the WWW. In *Proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00),* 2000.

[Agrawal et al., 1993] Agrawal R, Imielinski T, Swami A Mining association rules between sets of items in large databases. In *Proc. Of the ACM SIGMOD Conference on Management of Data*, 207-216, 1993.

[Alfonseca and Manandhar, 2002]Alfonseca E. and Manandhar S.. Improving an Ontology Refinement Method with Hyponymy Patterns. Language Resources and Evaluation (LREC-2002), Las Palmas, Spain, 2002.

[Appelt, 1996] D.E. Appelt. 1996. The Common Pattern Specification Language. *Technical report, SRI International, Artificial Intelligence Center.*

[Arens et al., 1993] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, Craig A. Knoblock. Retrieving and Integrating Data from Multiple Information Sources, *International Journal of Cooperative Information Systems,* 2(2), 1993

[At-Mokhtar and Chanod, 1997] Salah Aït-Mokhtar and Jean-Pierre Chanod. Incremental finite-state parsing. In *Proceedings of ANLP97*, Washington, 1997.

[Barzilay and McKeown, 2001] Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th ACL Meeting*, Toulouse, France, 2001.

[Basili and Pazienza, 1997] R. Basili and M.T. Pazienza. Lexical acquisition for information extraction. In M.T. Pazienza, editor, *Information Extraction – a multidisciplinary Approach to an emerging Information Technology*. Springer-Verlag - Lecture Notes, 1997.

[Basili et al., 1992]Roberto Basili, Maria Teresa Pazienza, and Paola Velardi. A shallow syntactic analyser to extract word association from corpora. *Literary and linguistic computing*, 7:114–124, 1992.

[Basili et al., 1997] Roberto Basili, Maria Teresa Pazienza, and Michele Vindigni. Corpus-driven unsupervised learning of verb subcategorization frames. In M. Lenzerini, editor, *Fifth Conference of the Italian Association for Artificial Intelligence (AI*IA97)*, number 1321 in LNAI, Heidelberg, Germany. Springer-Verlag, 1997

[Basili et al., 1998a] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Efficient parsing for information extraction. In *Proc. of the ECAI98*, Brighton, UK, 1998

[Basili et al., 1998b] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Evaluating a robust parser for italian language. In *Proc. of the THE EVALUATION OF PARSING SYSTEMS Workshop, held jointly with 1st LREC*, Granada, Spain, 1998

[Basili et al. , 1998c] R. Basili, M. Di Nanni, L. Mazzucchelli, M.V. Marabello, and M.T. Pazienza. Nlp for text classification: the trevi experience. In *Proceedings of the Second International Conference on Natural Language Processing and Industrial Applications, Universite' de Moncton, New Brunswick (Canada)*, 1998.

[Basili et al., 1999]Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Lexicalizing a shallow parser. In *Proc. of the Traitement Automatique de la Langue Naturelle, TALN99*, Cargese, FR, 1999.

[Basili et al., 2000a] Roberto Basili, Maria Teresa Pazienza, and Fabio Massimo Zanzotto. Customizable modular lexicalized parsing. In *Proc. of the 6th International Workshop on Parsing Technology, IWPT2000*,

[Basili et al., 2000b] Roberto Basili, Maria Teresa Pazienza, and Michele Vindigni. Corpus-driven learning of event recognition rules. In *Proceedings of the Workshop on Machine Learning for Information Extraction, held jointly with ECAI 2000*, Berlin, Germany, 2000.

Trento, Italy, 2000.

[Basili et al., 2001] Roberto Basili, Roberta Catizone, Luis Padro, Maria Teresa Pazienza, German Rigau, Andrea Setzer, Nick Webb, Yorick Wilks, and Fabio Massimo Zanzotto. Multilingual authoring: the namic approach. In *Proc. of the Human Language Technology and Knowledge Management held jointly with ACL2001*, Toulose, France, 2001

[Basili and Zanzotto, 2002] Roberto Basili, Fabio Massimo Zanzotto. Parsing Engineering and Empirical Robustness. *Journal of Natural Language Engineering 8/2-3*, June 2002

[Basili et al., 2002] R.Basili, M.T.Pazienza, F.M. Zanzotto. Learning IE patterns: a terminology extraction perspective. *Workshop of Event Modelling for Multilingual Document Linking at LREC 2002*, Canary Islands (Spain), 2002.

[Basili et al. , 2003] Roberto Basili, Michele Vindigni, Fabio Massimo Zanzotto. Integrating ontological and linguistic knowledge for Conceptual Information Extraction, Submitted for publication, 2003

[Basili et al., 2004] Roberto Basili, Marco Cammisa, and Fabio Massimo Zanzotto. A semantic similarity measure for unsupervised semantic disambiguation. In *Proceedings of the Language, Resources and Evaluation LREC 2004 Conference*, Lisbon, Portugal, 2004.

[Batini et al., 1986] C. Batini, M.Lenzerini, S.B. Navathe. A comparative analysis of methodologies for schema integration. *Computing Surveys,* 18(4), 1986

[Beneventano et al., 2001] D. Beneventano, S. Bergamaschi, F. Guerra, M. Vincini. The MOMIS approach to Information Integration, *IEEE and AAAI International Conference on Enterprise Information Systems (ICEIS01)*, Setúbal, Portugal, 2001.

[Bentivogli et al., 2002] Luisa Bentivogli, Emanuele Pianta, Christian Girardi. MultiWordNet: developing an aligned multilingual database, in *Proceedings of the First International Conference on Global WordNet*, Mysore, India, January 21-25, 2002

[Berners-Lee, 1999] Berners-Lee T. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. HarperCollins Publishers, New York, 1999.

[Bisson et al., 2000] Bisson G, Nedellec C, Cañamero D. Designing Clustering Methods for Ontology Building. The Mo'K Workbench. In S. Staab, A. Maedche, C. Nedellec, P. WiemerHasting (eds.), *Proceedings of the Workshop on Ontology Learning,* 14th European Conference on Artificial Intelligence, 2000.

[Bisson, 1992] Bisson G. Learning in FOL with a similarity measure. In *Tenth National Conference of Artificial Intelligence*. San José, California. July 1992. 12-16, 1992.

[Bisson, 1994] Bisson G. *Conceptual Clustering.* In *Mlnet Summer School on Machine Learning and Knowledge acquisition.* Dourdan, France, September 1994. 5-10

[Brent, 1993] Michael R. Brent. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19-2, 1993.

[Brill and Resnik, 1994] E. Brill and P. Resnik. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of COLING-94*, Kyoto, Japan, 1994.

[Cabré, 1998] Maria Theresa Cabré. Terminology. John Benjamins Publishing Company, Amsterdam/Philadelphia, 1998

[Chaelandar and Grau, 2000] Chaelandar G, Grau B. SVETLAN'- A System to Classify Words in Context. In S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.) *Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00, Be*, 2000.

[Chaudhri et al., 1998] V. Chaudhri, R. Fikes, P. Karp, J. Rice. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. *Proceedings of AAAI-98*, Madison, Wisconsin, February, 1998

[Carroll and Briscoe, 1998] G. Carroll, J.and Minnen and E. Briscoe. Can subcategorisation probabilities help a statistical parser? In *In Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*, Montreal, Canada, 1998.

[Collins, 1996] Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34 th Annual Meeting of the Association for Computational Linguistics*, 1996.

[Computerm, 1998] Didier Bourigault, Christian Jacquemin, Marie-Claude L'Homme Editors, *Proceedings of the First Workshop on Computational Terminology COMPUTERM'98, held jointly with COLING-ACL'98*, Montreal, Quebec, Canada, 1998

[Cunningham et al., 2002] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. 2002. *The GATE User Guide*. http://gate.ac.uk/.

[Daille, 1994] Beatrice Daille. Approach mixte pour l'extraction de termilogie: statistique lexicale et filters linguistiques. *PhD Thesis*, C2V, TALANA, Universitè Paris VII, 1994.

[DAML+OIL] www.daml.org

[de Roure and Shadbolt, 2003a] Baker M. Jennings N. R. de Roure, D. and N. Shadbolt. The evolution of the grid. In Fox G. Berman, F. and A. J. G. Hey, editors, *Grid Computing - Making the Global Infrastructure a Reality*, pages 65–100. John Wiley and Sons Ltd, 2003

[de Roure and Shadbolt, 2003b]Jennings N. R. de Roure, D. and N. Shadbolt. The semantic grid: A future e-science infrastructure. In Fox G. Berman, F. and A. J. G. Hey, editors, *Grid Computing - Making the Global Infrastructure a Reality*, pages 437–470. John Wiley and Sons Ltd, 2003.

[Delmonte, 1992] R. Delmonte. *Linguistic and Referential Processes in Text Analaysis by computers*. UNIPRESS, Venezia, 1992.

[Dimitrov, 2002] M. Dimitrov. 2002. A Light-weight Approach to Coreference Resolution for Named Entities in Text. *MSc Thesis, University of Sofia,* Bulgaria. (http://www.ontotext.com/ie/thesis-m.pdf.)

[Doan et al., 2002] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to Map between Ontologies on the Semantic Web. *WWW2002*, Honolulu, Hawaii, USA, 2002

[Engels, 2001a] Engels R CORPORUM-OntoExtract. Ontology Extraction Tool. Deliverable 6 Ontoknowledge, 2001. http://www.ontonowledge.org/del.shtml

[Engels, 2001b] Engels R. CORPORUM-OntoWrapper. Extraction of structured information from web based resources. Deliverable 7, 2001– Ontoknowledge. http://www.ontonowledge.org/del.shtml

[Faure and Nédellec, 1998] Faure D, Nédellec C. A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition. In LREC workshop on adapting lexical and corpus resources to sublanguages and applications, Granada, Spain,1998.

[Faure and Nédellec, 1999] Faure D, Nédellec C. Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. In D. Fensel and R. Studer editors, Proc. Of the 11th European Workshop (EKAW'99), LNAI 1621, pages 329-334, 1999.

[Faure and Poibeau] Faure D, Poibeau T. First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), *Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00, Be*, 2000.

[Fensel et al., 2003] Dieter Fensel, Frank Van Harmelen, and Ian Horrocks. Oil and daml+oil: Ontology languages for the semantic web. In John Davies, Dieter Fensel, and Frank Van Harmelen, editors, *Towards the Semantic Web*, West Sussex, UK, 2003. John Wiley and Sons.

[Foster, 1998] Kesselman C. (eds) Foster, I. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998

[Gaizauskas and Humphreys, 1996] Robert Gaizauskas and Kevin Humphreys. XI: A simple prolog-based language for cross-classification and inheritance. In *Proceedings of the 7th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA96)*, Sozopol, Bulgaria, 1996.

[Gaizauskas et al., 1998] R. Gaizauskas, M. Hepple, and C. Huyck. A scheme for comparative evaluation of diverse parsing systems. In *Proc. of FIRST INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION*, Granada, Spain, 1998.

[Gaizauskas and Wilks, 1998] R. Gaizauskas and Y. Wilks. Information Extraction: Beyond Document Retrieval, In *Journal of Documentation, 54(1):70--105*, 1998.

[Genesereth and Nilsson, 1987] M.R. Genesereth, N.J. Nilsson. Logical Foundation of Artificial Intelligence, Morgan Kaufmann, Los Altos, Clalifornia, 1987

[Gomez and Manzano, 2003] Gomez-Perez, A.; Manzano-Macho, D. A survey of ontology learning methods and techniques. *Deliverable 1.5 of project IST 2000 -29243- OntoWeb*, May 2003.

[Green and Rubin, 1981] Greene, B.B. & G.M. Rubin. Automatic grammatical tagging of English. *Providence, R.I.: Department of Linguistics, Brown University*. 1981

[Green et al, 1963] B. F. Green, A. K. Wolf, C. Chomsky, & K. Laughery, Baseball: An automatic question answerer. In Feigenbaum, E. A., & Feldman, J. (Eds.), *Computers and Thought*, pp. 207—216, 1963.

[Greengrass, 2002] Ed Greengrass. Information Retrieval: A Survey, 2000. (on-line: http://www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf)

[Gupta et al., 2002] Gupta, K.M., Aha, D.W., Marsh, E., and Maney, T. (2002*). An architecture for engineering sublanguage WordNets*. In *Proceedings of the First International Conference On Global WordNet* (pp. 207-215). Mysore, India: Central Institute of Indian Languages, 2002.

[Grinberg et al., 1996] D. Grinberg, J. Lafferty, and D. Sleator. A robust parsing algorithm for link grammar. In *4th International workshop on parsing tecnologies*, Prague, 1996

[Gruber, 1993] T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *International Workshop on Formal Ontology*, Padova, Italy, 1993.

[Guarino, 1998] Nicola Guarino. Formal Ontology and Information Systems, *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, Trento, Italy, IOS Press, 1998.

[Hahn and Schulz, 2000] Hahn U, Schulz S. Towards Very Large Terminological Knowledge Bases: A Case Study from Medicine. In *Canadian Conference on AI 2000:* 176-186, 2000.

[Hahn and Markó, 2001] Hahn U., and Markó K. Joint knowledge capture for grammars and ontologies. *Proceedings of the First International Conference on Knowledge Capture K-CAP* 2001: Victoria, BC, Canada, 2001.

[Hahn and Romacker, 1997] Udo Hahn and Martin Romacker. Text structures in medical text processing: empirical evidence and a text understanding prototype. In *Proceedings of the 1997 AMIA Annual Fall Symposium (formerly SCAMC)*, Nashville, TN, 1997.

[Hahn and Schnattinger, 1998] Hahn U., and Schnattinger K. Towards text knowledge engineering. In: AAAI '98 / IAAI '98 *Proceedings of the 15th National Conference on Artificial Intelligence & 10th Conference on Innovative Applications of Artificial Intelligence.* Madison, Wisco,1998.

[Hahn and Schulz, 2000] Hahn U., and Schulz S. Towards Very Large Terminological Knowledge Bases: A Case Study from Medicine. *Canadian Conference on AI 2000*: 176-186, 2000.

[Harabagiu and Moldovan, 2000] Harabagiu, S. M.; Moldovan D. I. Enriching the WordNet Taxonomy with Contextual Knowledge acquired from text. In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, (Eds) S. Shapiro and L. I, 2000.

[Harabagiu et al.2000] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. Falcon: Boosting knowledge for answer engines. In *Proceedings of the Text Retrieval Conference (TREC-9)*, 2000.

[Hearst, 1998] Hearst M. A.. Automated Discovery of WordNet Relations. In Christiane Fellbaum (Ed.) *WordNet: An Electronic Lexical Database*, MIT Press, pp. 132—15, 1998.

[Hearst, 1992] Hearst M.A. Automatic acquisition of Hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistic*, Nantes, France, July 1992.

[Hendler, 2001] James Hendler. Agents and the Semantic Web. *IEEE Intelligent Systems Journal* (March/April 2001)

[Heflin et al., 1999] Jeff Heflin, James Hendler, Sean Luke. SHOE: A Knowledge Representation Language for Internet Applications, Technical Report CS-TR-4078 (UMIACS TR-99-71), 1999.

[Hobbs, 1993] J.R. Hobbs. The Generic Information Extraction System, In *Proceeding of the 5th Message Understanding Conference (MUC-5), Morgan Kaufman pp.87-91*, 1993.

[Hobbs et al., 1996] Hobbs, Appelt, Bear, Israel, Kameyama, Stickel, and Tyson. Fastus: A cascaded finite-state transducer for extracting information from natural-language text. In Roche and Schabes, editors, *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge MA, 1996.

[Hovy and Lin, 1999] Hovy, E.H. and Lin C.-Y. Automated Text Summarization in SUMMARIST. In M. Maybury and I. Mani (Eds), *Advances in Automatic Text Summarization*. Cambridge: MIT Press, 1999.

[Hsu et al., 2001] Hsu W.L., Wu S.H., and Chen, Y.S.Event identification based on the Information Map – InfoMap. In *symposium NLPKE of the IEEE SMC Conference*, Tuckson, Arizona, USA, 2001.

[Humphreys et al., 1998] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of sheffield: Description of the LASIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*. Morgan Kaufman, 1998.

[Hwang, 1999] Hwang, C. H. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In. *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*, Linköping, Sweden, 1999.

[Kietz,, 2000] Kietz JU, Maedche A, Volz R. A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. In: *Aussenac-Gilles N, Biébow B, Szulman S (eds) EKAW'00 Workshop on Ontologies and Texts*. Juan-Les-Pins, France. CEUR Workshop Proceedings 51:4, 2000.

[IEEE, 1990] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Computer Society, IEEE Std 610.12-1990, 1990.

[Jacquemin, 1997] Christian Jacquemin. Variation terminologique: Reconnaissance et acquisition automatiques de termes et de leurs variantes en corpus. *Mémoire d'Habilitation à Diriger des Recherches en informatique fondamentale*, Universitè de Nantes, France, 1997

[Jacquemin, 2001] Christian Jacquemin. Spotting and Discovering Terms through Natural Language Processing, MIT Press Cambrige, Massachussetts, 2001

[Kaji et al., 2002] Nobuhiro Kaji, Daisuke Kawahara, Sadao Kurohashi, and Satoshi Sato. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania, 2002.

[Kononenko and Bratko, 1991] I. Kononenko and I. Bratko. Information-based evaluation criteria for classifier's performance. *Machine Learning*, 6(1):67–80, 1991.

[Lee, 2001] T.B. Lee. The semantic web. In *Scientific American*, volume May, pages 28–37, 2001.

[Levenshtein, 1966] I.V. Levenshtein. Binary Codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8), 1966

[Lin and Hovy, 2000] Lin, C.-Y. and Hovy E.H. The Automated Acquisition of Topic Signatures for Text Summarization. *Proc. of the COLING Conference*. Strasbourg, France. August 2000.

[Liu, 1996] Liu W. Z. An Integrated Approach for Different Attribute Types in Nearest Neighbour Classification. *The Knowledge Engineering Review*, 1996.

[Maedche and Staab, 2001] Maedche A, Staab S. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, 16(2), 2001.

[Meadche and Staab, 2001] Alexander Meadche and Steffen Staab. Comparing Ontologies-Similarity Measures and Comparison Study, Internal Report No. 408, Institute AIFB, University of Karlsruhe, Germany, 2001

[Maedche and Staab, 2003] Maedche A. and Staab S. Ontology Learning. In S. Staab & R. Studer (eds.) *Handbook on Ontologies in Information Systems*. Springer 2003.

[Maedche and Staab, 2000] Maedche, A. and Staab, S. Discovering Conceptual Relations from Text. In: W.Horn (ed.): ECAI 2000. *Proceedings of the 14th European Conference on Artificial Intelligence*, Berlin, August 21-25, 2000. IOS Press, Amsterdam, 2000. http://www.aifb.uni-k

[Maedche and Volz, 2001] Maedche, A. and Volz, R. The Text-To-Onto Ontology Extraction and Maintenance Environment. In *Proceedings of the ICDM Workshop on integrating data mining and knowledge management*, San Jose, California, USA, 2001.

[Mena and Illarramendi, 2001] E. Mena and A. Illarramendi. Ontology-Based Query Processing for Global Information Systems, Kluwer Academic Publishers, ISBN 0-7923-7375-8, pp. 215, 2001.

[Marcus et al., 1993] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19:313–330, 1993.

[Menzel, 1995] W. Menzel. Robust processing of natural language. *Lecture Notes in Computer Science*, 981:19–34, 1995.

[MESH] MESH, editor. *Medical Subject Headings*, www.nlm.nih.gov/mesh/meshhome.html.

[Mikheev and Finch, 1997] Mikheev, A. Finch, S. A Workbench for Finding Structure in Texts. *Proceedings of ANLP-97* (Washington D.C.). ACL March 1997. pp 8.

[Miller, 1995] G.A. Miller. WordNet: A lexical Database for English. *Communicationof the ACM*, 38(11):39-41, Nov. 1995.

[Milne, 1968] A.A. Milne. Winnie-the-Pooh. Noordhoof, Groningen, 1968

[Missikoff et al., 2002] Missikoff M., Navigli R., and Velardi P. (2002). The Usable Ontology: An Environment for Building and Assessing a Domain Ontology. Research paper at *International Semantic Web Conference (ISWC) 2002*, June 9-12th, 2002 Sardinia, Italia.

[Moldovan and Girju, 2000] Moldovan, D. I.; Girju, R. C. Domain-Specific Knowledge Acquisition and Classification using WordNet. In *Proceedings of International Florida Artificial Intelligence Research Society (FLAIRS-2000) conference*, Orlando, Fl., May 2000.

[Moldovan and Girju, 2001] Moldovan, D. I.; Girju, R. C. An interactive tool for the rapid development of knowledge Bases. In *International Journal on Artificial Intelligence Tools (IJAIT), vol 10., no. 1-2*, March 2001.

[Moldovan et al., 2000] Moldovan, D. I.; Girju, R. C.; Rus, V. Domain-Specific Knowledge Acquisition from Text. In *Proceedings of the Applied Natural Language Processing (ANLP-2000) conference*, Seattle, WA., April-May 2000.

[Moldovan and Surdeanu, 2002] *Dan Moldovan, Mihai Surdeanu,* On the Role of Information Retrieval and Information Extraction in Question Answering Systems, In *Information Extraction in the Web Era*, M.T.Pazienza (ed), Springer, 2003, pag 129-147.

[Montague, 1974] Richard Montague. *Formal Philosophy (Selected Papers of Richard Montague).* Yale UP, New Haven and London, 1974.

[Morik, 1993]Morik K. Balanced Cooperative Modelling. *Machine Learning, 11(1),* 1993, pages 217-235.

[Morin, 1999] Morin E. Automatic acquisition of semantic relations between terms from technical corpora. Proc. *Of the Fifth Int. Congress on Terminology and Knowledge Engineering (TKE-99),* TermNet-Verlag, Vienna, 1999.

[Moschitti and Zazotto, 2002] Alessandro Moschitti, Fabio Massimo Zanzotto A robust summarization system to explain document categorization. In *Proceedings of 2nd Workshop of Robust Methods in Analysis of Natural language Data ROMAND2002,* Frascati (Italy), July 2002.

[MUC, 1995] MUC-1995. Proceedings of the sixth message understanding conference(MUC-6). In *Columbia, MD*. Morgan Kaufmann, 1995.

[MUC, 1997] MUC-7. Proceedings of the seventh message understanding conference(MUC-7). In *Columbia, MD*. Morgan Kaufmann, 1997.

[Navigli et al., 2003] Navigli R., Velardi P, and Gangemi A. Ontology Learning and its application to automated terminology translation. *IEEE Intelligent Systems, vol. 18, n.1,* January February 2003.

[OXF] http://www.orbeon.com/oxf

[OWL] http://www.w3.org/TR/owl-ref/

[Pazienza, 1997] Maria Teresa Pazienza. *Information Extraction. A Multidisciplinary Approach to an Emerging Information Technology*. Number 1299 in LNAI. Springer-Verlag, Heidelberg, Germany, 1997.

[Pazienza, 1998] M.T Pazienza (Ed). *Information Extraction. Towards Scalable, Adaptable Systems*, Number 1714 in LNAI, Springer-Verlag, Heidelberg, Germany, 1998.

[Pazienza, 2003] M.T Pazienza (Ed). *Information Extraction in the Web Era,* Number 2007 in LNAI, Springer-Verlag, Heidelberg, Germany, 2003.

[Pazienza and Vindigni, 2003] Maria Teresa Pazienza, Michele Vindigni,. Agents based ontological mediation in IE systems, forthcoming

[Pearson, 1998] Jennifer Pearson. Terms in Context. John Benjamins Publishing Company, Amsterdam/Philadelphia, 1998

[Peters et al., 2002] Carol Peters, Martin Braschler, Julio Gonzalo, Michael Kluck (Eds.). Advances in Cross-Language Information Retrieval . *Third Workshop of the Cross-Language Evaluation Forum, CLEF 2002*, Rome, Italy, September, 2002. *Lecture Notes in Computer Science* 2785, Springer 2003, 828p. Springer 2002.

[Pollard and Sag, 1994] C. Pollard and I.A. Sag. *Head-driven Phrase Structured Grammar*. Chicago CSLI, Stanford, 1994

[Pustejovsky, 1995] James Pustejovsky. The Generative Lexicon, MIT Press, Cambridge, 1995

[Quinlan, 1993] J. Quinlan. *C4:5:programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[Ratnaparkhi and Roukos, 1994] A. Ratnaparkhi and S. Roukos. A maximum entropy model for prepositional phrase attachment. In *In Proceedings of the ARPA Workshop on Human Language Technology, Morgan Kaufmann*, 1994.

[Ravichandran and Hovy, 2002]Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania, 2002.

[Resnik, 1995] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[Riloff, 1996] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 1044--1049*, 1996.

[Rijsbergen, 1979] Keith van Rijsbergen. Information retrieval. 1979. (on line: http://www.dcs.gla.ac.uk/Keith/Preface.html)

[Salton and McGill, 1983] Gerard Salton and M J McGill. Introduction to Modern Information Retrieval, McGraw-Hill Company, 1983.

[Salton and Buckley, 1989] G. Salton J. Allan C. Buckley. Automatic Text Processing: The transformation, analysis and retrieval of Information by computer ,Addison-Wesley, Reading, MA, 1989.

[Sampson, 1993] Geoffrey Sampson. The susanne corpus. *ICAME Journal*, 17.125-7, 1993.

[Saracevic, 1995] T. Saracevic. Evaluation of Evaluation in Information Retrieval. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 137-146, 1995.

[SIMPLE] http://www.ub.es/gilcub/SIMPLE/simple2.html

[Srikant and Agrawal, 1995] Srikant R, Agrawal R. Mining generalized association rules. *In Proc. Of VLDB'95*, pages 407-419, 1995.

[Steel, 1996] Luc Steel. Emergent Adaptive Lexicons. *Proceedings of the Simulation of Adaptive Behavior Conference,* MIT Press, 1996

[Tesniere, 1959] L. Tesniere. *Elements de syntaxe structural*. Klincksiek, Paris, France, 1959.

[Velardi et al., 2001] Velardi P., Missikoff M., and Fabriani P. Using Text Processing Techniques to Automatically enrich a Domain Ontology. *ACM conference on Formal Ontologies in Information Systems (FOIS 2001),* Maine, USA, 2001.

[Velardi et al., 2002] Velardi P., Navigli R., and Missikoff M. Integrated approach for Web ontology learning and engineering. *IEEE Computer* - November 2002.

[Voorhees, 2003a] Ellen M. Voorhees. Overwiev of TREC 2003, Proceedings of the Text Retrieval Conference 2003, 2003.

[Voorhees, 2003b] E.M. Voorhees. Overview of the TREC 2003 Question Answering Track, Proceedings of the TREC-2003.

[Vossen, 1998] P.Vossen. EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Kluwer Academic Publisher, Dordrecht, 1998.

[Weinstein and Birmingham, 1999] Peter C. Weinstein, William P. Birmingham. Comparing Concepts in Differentiated Ontologies. *Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW 1999),* Alberta, Canada, 1999

[Witten and Frank, 1999] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, Chicago, IL, 1999.

[Woods, 1977] W. A. Woods. Lunar rocks in natural English: Explorations in natural language question answering, In A. Zampolli (ed.) *Linguistic Structure Processing*, North Holland, 1977: 521-569, 1977.

[Wrust, 1931] Eugene Wrust. Die Internationale Sprahnormung in der Technik, besonders in der Elektrotechnik. VDI-Verlag, Berlin, Germany, 1935

[Wu and Hsu, 2002] Wu S.H, Hsu W.L. SOAT: A Semi-Automatic Domain Ontology Acquisition Tool from Chinese Corpus. *In the 19th International Conference on Computational Linguistics, Howard International House and Academia Sinica,* Taipei, Taiwan, 2002.

[Worm and Rupp, 1998] Karsten L. Worm and C. J. Rupp. Towards robust understanding of speech by combination of partial analysis. In *Proc. of the ECAI98*, Brighton, UK, 1998

[Wu et al., 2003] Weili Wu, Hui Xiong, Shashi Shekhar. Clustering and Information Retrieval. Kluwer Academic Publishers, Boston, November 2003.

[Yangarber et al, 2000] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *132 Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 2000.

[Yangarber2003] Roman Yangarber. 2003. Acquistion of domain knowledge. In Maria Teresa Pazienza, editor, *Information Extraction in the Web Era*, pages 1–28. Springer-Verlag.

[Zajac, 2001] Remi Zajac. Towards Ontological Question Answering. *ACL-2001 Workshop on Open-Domain Question Answering,* Toulose, France, 2001

[Zanzotto, 2002] F.M.Zanzotto. L'estrazione della terminologia come strumento per la modellazione di domini conoscitivi. *Tesi di Dottorato di Ricerca, Università degli Studi di Roma Tor Vergata* , 2002.

[Zanzotto and Stellato, 2004] Fabio Massimo Zanzotto, Armando Stellato. Exploiting the Semantic Fingerprint for Tagging "Unseen" Words Beyond Named Entity Recognition: Semantic labelling for NLP tasks. *Workshop held jointly with LREC2004* LISBON, Portugal, May 2004

7. Acronymes List

API	Application Programming Interface
CDF	Current Design Facility
CGI	Common Gateway Interface
CLEF	Cross Language Evaluation Forum
CO	Coreference Resolution
DAML	DARPA Agent Markup Language
DAML-OIL	DARPA Agent Markup Language – Ontology Interchange Language
DARPA	US Defence Advanced Research Project Agency
DB	Database
DBMS	Database Management System
DCH	Domain Concept Hierarchy
DR	Document Retrieval
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
GNU	GNU is Not Unix
GPL	General Purpose Licence
HPSG	Head-Driven Phrase Structure Grammar
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
Idf	Inverse Document Frequency
IE	Information Extraction
IR	Information Retrieval
KB	Knowledge Base
LKB	Linguistic Knowledge Base
LR	Language Resources
LRE	Linguistic Research and Engineering initiative
ML	Machine Learning
MT	Machine Translation
MUC	Message Understanding Conference
NE	Named Entity

NER	Named Entity Recognition
NIST	US National Institute of Standard and Technologies
NL	Natural Language
NLP	Natural Language Processing
OIL	Ontology Interchange Language
OXF	Open XML Framework
OWL	Web Ontology Language
PCFG	Probabilistic Context-Free Grammar
PHP	Personal Home Page
POS	Part of Speech
PR	Processing Resources
PSG	Phrase Structure Grammar
Q/A	Question Answering
Q&A	Question Answering
RDF	Resource Description Framework
RTS	Relation Type System
RTV	University of Rome Tor Vergata
SGML	Standard Generalized Markup Language
SHOE	Simple HTML Ontology Extensions
SHUMI	Support to Human Machine Interaction
SR	Semantic Relations
ST	Scenario Template Production
SW	Semantic Web
TE	Template Element Production
TE	Terminology Extraction
Tf	Term frequency
TREC	Text REtrieval Conference
URL	Uniform Resource Locator
VIT	Verbmobil Interface Term
WM	World Model
WWW	World Wide Web
WWWW	World Wide Web Worm

XDG	Extended Dependency Graph
XML	Extensible Markup Language