Theorie und Praxis des Lexikons

Herausgegeben von
Frank Beckmann und Gerhard Heyer

Sonderdruck

W
DE
G

Walter de Gruyter · Berlin · New York
1993

Encoding Lexicographic Definitions as Typed Feature Structures

Nicoletta Calzolari, Johan Hagman,
Elisabetta Marinai, SimonettaMontemagni,
Antonietta Spanu & Antonio Zampolli, Pisa

1. Introduction

This paper has been written within the framework of Acquilex, an Esprit-Bra project (see Boguraev et al. 1988). The objectives of the project were to extract lexical and conceptual knowledge from machine readable versions of conventional dictionaries (Mrds) and to construct a lexical knowledge base (Lkb) containing multilingual lexical information usable in Natural Language Processing (Nlp) applications. These are two crucial steps in the process of converting dictionary resources available in machine readable form into formalized computerized lexicons (see for example Calzolari 1991). The overall process has been carried out for English, Italian, Dutch and Spanish, on several Mrds, using different techniques and methodologies. The different extraction strategies, adopted in the different sites involved in the project, converge to a common final goal that is the representation of the extracted information in a multilingual lexical knowledge base.

The Acquilex project came to an end in July 92, and follow-up began immediately afterwards, i.e. Acquilex-II. We give here an overview of the main lines of research and development undertaken in Pisa within the project, in view of a first evaluation of the advantages and disadvantages of the chosen approach (i.e. a typed feature structure (Tfs) formalism) for the treatment of lexicographic definitions. In particular, we will concentrate on the process of converting data from a lexical database to a knowledge base. Two central issues should be considered in this respect:

(a) the extraction of semantic information from dictionary definitions;

(b) the representation of such information in a formal and consistent way within a lexical knowledge base.

We will not discuss here the preliminary stage of this process, i.e. the conversion of MRDs into lexical data bases (LDBs), which make their data and their structure fully explicit in a format that lends itself to flexible querying (for the Multilingual Lexical Database, see MARINAI, PETERS, PICCHI 1990).

In the present paper, we will briefly describe the following stages of the process:

- syntactic analysis of definitions;
- extraction of semantic information from parsed definitions;
- 'genus' disambiguation and taxonomy building;
- filtering of the information coming from the 'differentia' part of the definition;
- conversion of the results of the extraction procedure into a typed feature structure (TFS) representation system.

While the first four steps are strictly related with the extraction issue (a), the last is concerned with the question of representation (b). Whenever possible, elements for comparison and evaluation are given with respect to the methodology adopted and the results obtained for each single step.

The MRDs on which the whole procedure has been experimented are the IL NUOVO DIZIONARIO GARZANTI (henceforth, GARZANTI) and the ITALIAN DMI DATABASE, mainly based on the Zingarelli dictionary (henceforth, DMI). The process in its entirety, from extraction to representation within the lexical knowledge base, has been tested on noun definitions. Given the partial domain dependence of the extraction procedure and the needs and constraints of the lexical representation language adopted (typed feature structure representation system), only one semantic domain – that of Food and Drinks – has been fully explored. At the current stage of research, experiments restricted to single steps of the whole process have been carried out in other semantic domains, and for other word classes.

2. The System for the Extraction and Representation of Lexical Knowledge

2.1 Designing the System

The process of extracting and representing the lexical knowledge contained in the MRDs is carried out by a modular system sketched in Figure 1. Different tasks are performed within the system by different modules running under different environments (MS/DOS, CMS/VM, Macintosh). The fact that the whole system has been developed over different environments is due to the availability of already existing tools: for instance, a system designed to store, maintain, and access both mono- and bilingual lexical data under MS/DOS, a general purpose Italian grammar and the core procedures of the semantic parser running under CMS/VM, and the LKB, specifically developed in Cambridge within the project, running on the Macintosh system.

The overall goal of constructing a lexical knowledge base by exploiting the information contained in MRDs transformed this heterogeneous set of modules into a real system. The system construction involved, on the one hand, adapting existing tools to the needs and goals of ACQUILEX and, on the other hand, building new modules, and setting up the appropriate interfaces linking them. The LKB software has been integrated within the system without any intervention. Therefore, the ACQUILEX-Pisa system is composed of:

(a) components which were initially designed independently of the LKB goal, but which appeared very suitable for it, as well as

(b) modules built specifically for the final goal of the project, i.e. the lexical knowledge representation base.

From this, it follows that among the principles which guided the design of our system there is an enlarged notion of 'reusability'. First, the ACQUILEX project can be seen as a prototype of the line of research directed towards exploiting and reusing lexical information implicitly or explicitly present in preexisting lexical resources such as MRDs (see the concept of 'reusable_1' in CALZOLARI 1991). The semantic information extracted from dictionary definitions is 'reused' within the ACQUILEX system for constructing the LKB, and at the same time is stored in the extended lexical data base and made available for different scopes in other NLP projects (thus becoming 'reusable' in the 'reusable_2' sense). Secondly, the system designed at Pisa within the ACQUILEX framework is an example of 'reusability of components'; the lexical database system used

for storing and querying the data, the Italian grammar used for parsing definitions, the core procedures of the semantic parser are all modules conceived within other frameworks which have been adapted and productively integrated within the ACQUILEX-Pisa system.

A system designed in such a way is obviously incrementable. The single existing modules can be further adapted in order to satisfy new requirements as the need arises. For instance, the extraction procedure will be extended to lemmas belonging to parts of speech which are not currently considered (at the moment, the extraction procedure only operates on noun and verb definitions), or the Type System behind the LKB will be increased in order to represent entries related to new semantic domains (as we have done for 'place nouns', see SPANU 1992). Furthermore, additional modules can be developed and added to the system, either to perform new tasks or to integrate and tailor the results of an already existing module. This has been the case of the Dictionary Definition Disambiguator (DDD), a small component operating after the grammar, which has been added to the system in order to refine the syntactic analysis performed on the basis of general grammatical expertise (more details about this will be given in the syntactic analysis section).

2.2 System Components

The main source for the extraction procedure is the Multilingual Lexical Data Base (MLDB) built on the basis of the available MRDs, and particularly a derived dictionary – the definition lexicon – consisting in an MLDB subset containing, for each lemma to be analysed, parts of speech, sense numbers, and definitions. Definitions are in fact, in this context, the main source of information.

The approach being experimented for the extraction of information from dictionary definitions follows a two-stage strategy. First, a general purpose Italian grammar, which has been specialized to handle the language used within dictionary definitions, provides an organized structure corresponding to an initial syntactic analysis for each definition. A pattern-matching procedure is then in charge of mapping lexical and/or structural patterns onto the syntactic description computed at the previous stage, with the result of deriving and making explicit the semantic knowledge implicitly stored in the definition (see the steps of syntactic and semantic analysis in Figure 1).

The results of the semantic analysis are stored as a separate lexicon – the genus and differentiae lexicon. This lexicon contains information which in the MLDB was implicit and which has been made explicit through the extraction procedure; the 'genus' information on the one side, the

semantic relations inferred from the 'differentiae' on the other. The genus information is used to build up taxonomies; a preliminary and necessary step of this stage is the genus disambiguation. The results of the analysis of the differentia part can be divided into two main classes:

(a) data which have been extracted and interpreted as values of recognized semantic relations (a part can be easily converted into the LKB formalism, while another part cannot be adequately represented in it);

(b) intermediate parsing results which need more processing before being safely associated with a given semantic relation (and therefore before they can be represented in the LKB).

Taxonomies and safely identified semantic relations which can be appropriately represented in the LKB will be the input of the conversion procedure whose final output is the LKB lexicon; at the same time, the conversion inputs, together with the data needing further processing (the intermediate parsing results), and the interpreted data lacking a counterpart in the LKB (the intermediate semantic results), are stored in the Extended MLDB. This extension should not be seen as an addition of new data to the original lexical data base, but as an explicit insertion of knowledge already present in an implicit form. In this way, part of this knowledge (taxonomic and all other semantic relations) will be directly reusable for other purposes, while partially processed lexical data will be available as input for further interpretation procedures.

As can be inferred from this brief description, not all the steps carried out by the system are fully automatic; the system combines batch (automatic) and interactive processes. In particular, interactive processes are adopted for building taxonomies, disambiguating the genus terms, and converting the semantic information extracted into an LKB representation. While the genus disambiguation procedure is fully interactive, the interactive environment in the other two procedures is aimed at validating and/or revising the results of fully automatic procedures (in both cases, the results of the semantic information extraction process).

3. The Extraction Procedure

The extraction procedure described here treats information which is only implicitly contained in MRDs and needs to be made explicit in order to be directly accessible, both within the Extended MLDB and the LKB frameworks. The information source consists of the dictionary definitions. This source, rich in detailed lexical information, is encoded in Natural

Language (NL) form. Other rich information sources, also encoded in NL form, could be considered within the dictionary context, e.g. example sentences, or idioms and their explanations. For instance, typical subjects and objects for verbs as well as collocations can be easily extracted from example sentences. At the moment, the extraction procedure operates only on definitions because of the type of information they produce with relation to the LKB goal.

The extraction of semantic information from definitions mainly provides two kinds of data, namely taxonomic information and other semantic relations. Even if this process is carried out within the same extraction procedure operating on the same source, the distinction between the two is relevant in this context for two reasons: first, the two kinds of information are extracted (usually) from different parts of the definition and, second, they are used for different purposes.

Taxonomies are built starting from the 'genus', which is the definition part expressing the class to which the designatum of the definiendum belongs. Feature structures, instead, are derived from the semantic relations specified in the 'differentia' part of the definition, which reports the properties discriminating the definiendum with respect to other members of the same class. With regard to the different role of these two kinds of information within the knowledge base, taxonomies are used to build the skeleton of inheritance chains through which properties of a class are passed on to its subclasses (therefore from general to more specific words), while local features, organized into structures, describe the characteristic properties of the definiendum.

Semantic information is not the only information that can be derived from definitions. This source also contains other kinds of linguistic information, even if not as systematically as for semantic information; for instance, it specifies the domain in which a given sense of the definiendum holds (see the CONTEXT relation in HAGMAN 1991), or gives the coordinates of its use from the pragmatic point of view. This additional information, also derived from definitions, appears as 'noise' when considered with respect to the representation in the lexical knowledge base as it is now. In the future, it will be useful in order to complete the formal definition of words, and in particular in order to specify their pragmatic dimension. At this stage of the research, we also extract this kind of information, and store it, together with additional semantic information not representable in the LKB, within the Extended MLDB, waiting for an adequate LKB representation.

As we have already stated, the results of this extraction phase are loaded into the Extended MLDB as a derived lexicon. Only one part of this derived lexicon can be directly reused and represented within the LKB;

the other part is constituted by the semantic information not convertible in LKB terms, and by intermediate parsing results which still need to be assigned a certain semantic interpretation.

The choice of also storing intermediate results (without starting again from the syntactic analysis) evidences one of the main features of this extraction procedure, i.e. it is an on-going process. The extraction procedure can never be considered complete. The results obtained by means of this procedure are themselves objects of generalizations leading in their turn to an integration or a simple revision of the patterns used to extract knowledge from definitions. Therefore, new parsed data give rise to new semantic relations to be detected and captured. From this point of view, the extraction process is mainly inductive, and the choice of storing and loading its partial as well as complete results in the MLDB testifies this aspect of progressive construction through generalizations from common elements.

3.1 Syntactic Analysis of Dictionary Definitions

The extraction process is, at the same time, related to the genus terms and to the semantic relations that can be derived from the differentiae. For genus terms extraction, two different methods, that is pattern matching at the string level and at the structural analysis level, yield promising results. Instead, if the differentiae are to be identified and organized into feature structures, only one method is feasible and reliable in our opinion: patterns based on structural information must be mapped onto a syntactic description of the definition. Only in this way can a satisfactory semantic accuracy in the extraction process be achieved.

There are two main advantages in basing this knowledge acquisition procedure on parsed syntactic structures. On the one hand, it is possible to abstract away from most of the variations in the surface realization of the same pattern. Even if, within the language of dictionary definitions, there are recurring defining formulae systematically used to express conceptual categories as well as semantic relations, these formulae undergo variations which can be better captured by means of patterns operating on syntactic structures than by means of patterns (typical of a text retrieval system) operating on the raw sequence of strings within the definition text. On the other hand, the results are expected to be more reliable.

It is possible to specify at which level of embedding, within the syntactic structure assigned to the definition, a given pattern has to be recognized; moreover, the relevant terms of the semantic relations detected by means of the structural patterns can be safely identified almost automatically (for a detailed discussion of the advantages of structural patterns

with respect to string ones see MONTEMAGNI & VANDERWENDE 1991).

This section focuses on the first stage of the extraction procedure, i.e. computing a syntactic analysis for each dictionary definition; the pattern matching procedure will map structural patterns onto the output of this stage, thereby deriving the semantic knowledge implicitly stored within the definition. The analysis produced at this stage is provided by a general text parser, with a general purpose grammar, and is subsequently refined and reshaped to allow for the peculiarities of dictionary text.

This is the approach taken in Pisa, and can be compared with the choice of the other partners of using dictionary specific parsing tools: a robust and flexible pattern matching and parsing tool has been applied to the Spanish Vox dictionary (see AGENO et al. 1990), while special purpose grammars developed by utilizing a general purpose parser have been experimented with LDOCE and the Dutch VAN DALE dictionaries (see VOSSEN 1990).

There are several reasons for approaching our Italian dictionaries with a parser and a grammar which are both domain independent. First of all, neither of the two Italian dictionaries which have been considered uses a restricted vocabulary in the definition texts; therefore, the scope of the vocabulary is the same as that of unrestricted texts.

The same happens at the syntactic level; the variety of phrasal constructions used within the definition text is comparable to that of textual corpora. In fact, the regularity of the lexically and syntactically constrained language used within our dictionary definitions lies in the frequent occurrence of lexical and syntactic patterns to express particular conceptual categories or semantic relations, rather than in a restricted vocabulary and limited range of syntactic constructions.

These are two linguistically motivated reasons for choosing to adopt a general text parser and grammar to parse definitions. Moreover, given that we are operating on two monolingual dictionaries, it was not the case to use a grammar designed for just one dictionary (e.g. the 'Longmanese' grammar, ALSHAWI 1989). This choice would have led to the development of two different and parallel grammars, namely for GARZANTI and for DMI definitions, even though one could probably have been partially derived from the other. Last but not least, the choice was made possible by the availability of an existing general purpose Italian grammar (see MONTEMAGNI 1991).

However, there are two main disadvantages in using a general purpose grammar for parsing dictionary text. First, at the end of the syntactic analysis performed with a general text grammar, ambiguity still remains. The fact that we are operating on dictionary definitions is helpful in this respect, as constructions which appear ambiguous in unrestricted texts

can often be disambiguated in the dictionary context. For instance, the ambiguity observed in free text in the attachment of a prepositional phrase can often be solved, in the context of dictionary definitions, on the basis of lexical and/or syntactic conditions which disambiguate the potential ambiguity.

The same happens with functional role assignment which may be ambiguous in Italian in some cases. In this respect, we can assume that constructions used within dictionary definitions are always unmarked, and therefore the ambiguity deriving from also considering marked orders of sentence constituents (such as Subject Object Verb, Object Verb Subject, and so forth) is very unlikely to occur in the dictionary text type. Second, there are specific dictionary-language constructions which are considered syntactically deviant from the point of view of the general grammar but occur typically within dictionary definitions. This is the case with definitions appearing as condensed fragments of wider texts; for instance, this occurs when obligatory complements are omitted, therefore resulting in ellipsis. While a general grammar would reject these constructions as ill-formed, a dictionary specific grammar has to parse them as typical occurrences of the dictionary language.

Instead of encouraging us to build a dictionary specific grammar, these observations induced us to use the general grammar with only general grammatical expertise in the beginning in order to discover the peculiarities of dictionary language exhaustively. A language that, in spite of the features for which it differs from the language of general texts, cannot be defined as specialized given that it does not operate in a specialized domain; it can be considered a sublanguage, remembering that this classification is only based on syntactic and lexical factors (see CALZOLARI 1984).

What emerged after parsing a significant subset of definitions with the general grammar was then exploited in a post-processor operating on the initial syntactic analysis in order to refine and reshape the analysis produced on the basis of general grammatical expertise. From this, it can be noticed that, even at the syntactic analysis stage, the module in charge of parsing definitions has been built progressively starting from the evaluation of the first parsing results obtained using the general purpose grammar.

Let us now briefly consider how the general strategy described above has been carried out. First, the syntactic analyses have been computed by a broad-coverage Italian grammar, making use of very limited lexical information (parts of speech, morphology and basic word class features) to produce a syntactic sketch of the input string that is syntactically, but not necessarily semantically, valid. The general approach adopted within

the grammar for dealing with unrestricted texts is that of supporting this initial syntactic analysis stage with relatively poor lexical information, which is not used to constrain the analysis but just to direct it (the approach, originally developed by JENSEN, is described in JENSEN 1988, 1989).

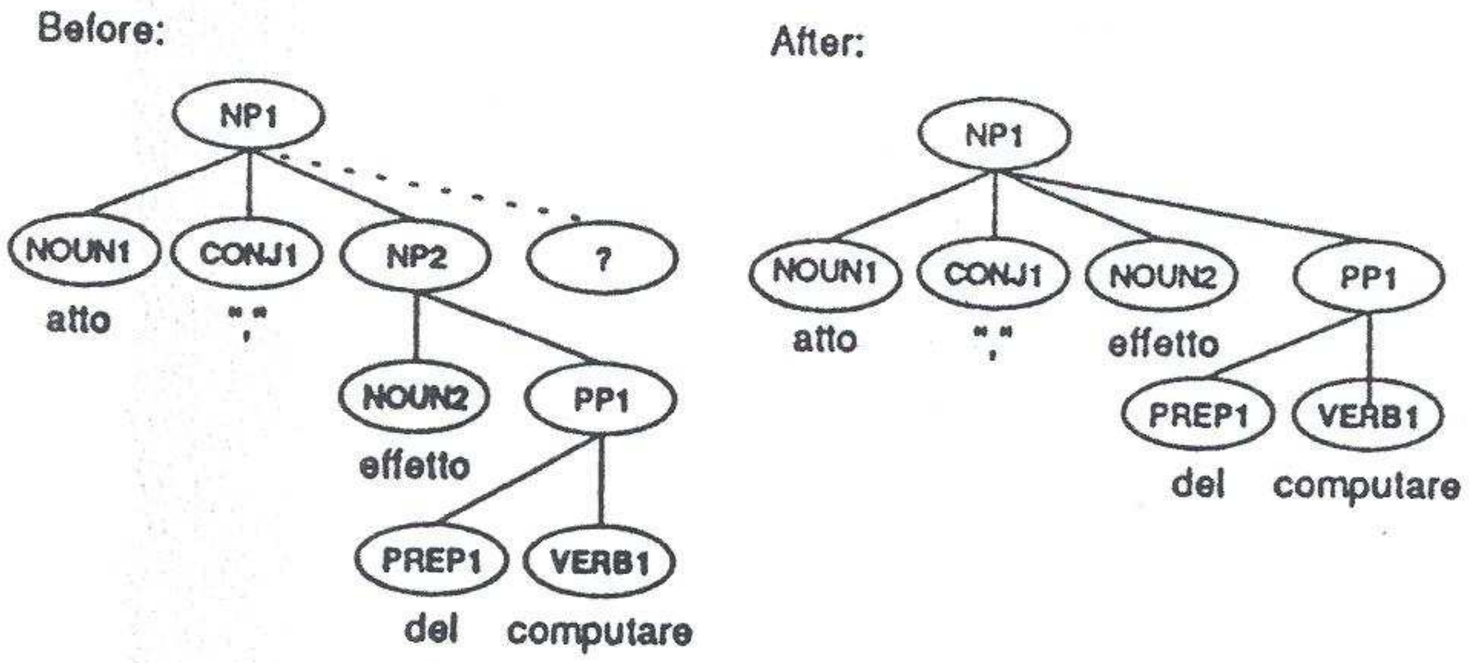
This approach appears particularly suitable in the dictionary context, where valency information, for instance, would have no effect because of the frequency of elliptical constructions, or simply because of the fact that most verbs, and especially the most frequent ones, are both transitive and intransitive, and can take a wide range of complementations. The parses produced by this grammar, on the basis of this restricted lexical information, contain syntactic and, whenever possible, functional information, but no semantic or other information beyond the functional level. The strategy followed within the general grammar for ambiguities, in assigning functional roles and attaching modifiers to their appropriate heads, is that of packing within the same structure the alternative parses. In this way, any combinatorial explosion is eliminated and, at the same time, all the necessary information is preserved for further processing stages.

After making an extensive inventory of the peculiarities of the language used within dictionary definitions, two main areas of the grammar needing to be tailored in order to give more appropriate results with respect to dictionary text were identified. We decided to perform the needed refinements in a post-processing stage rather than changing the general grammar itself (which describes the core structures of the language) which would have modified its output, independently of the type of text. For this dictionary specific revision task, two kinds of refinements (implemented in two different post-processors operating on the output of the general grammar) have been devised:

(a) rule out ambiguities, not applicable in the context of dictionary definitions, in modifier attachment or functional role assignment;

(b) handle incomplete parses, due to either dictionary specific constructions not occurring in free text, or – more generally – to gaps in the lexical or grammatical knowledge of the system.

The first refinement operates on a complete analysis and aims at reducing the high degree of ambiguity typical of free text by exploiting peculiarities of dictionary language. Conditions and heuristics have been formalized within a smaller (if compared to the size of the general grammar) dictionary specific component, the Dictionary Definition Disambiguator (DDD), operating on the output of the general grammar with the task of disambiguating it.

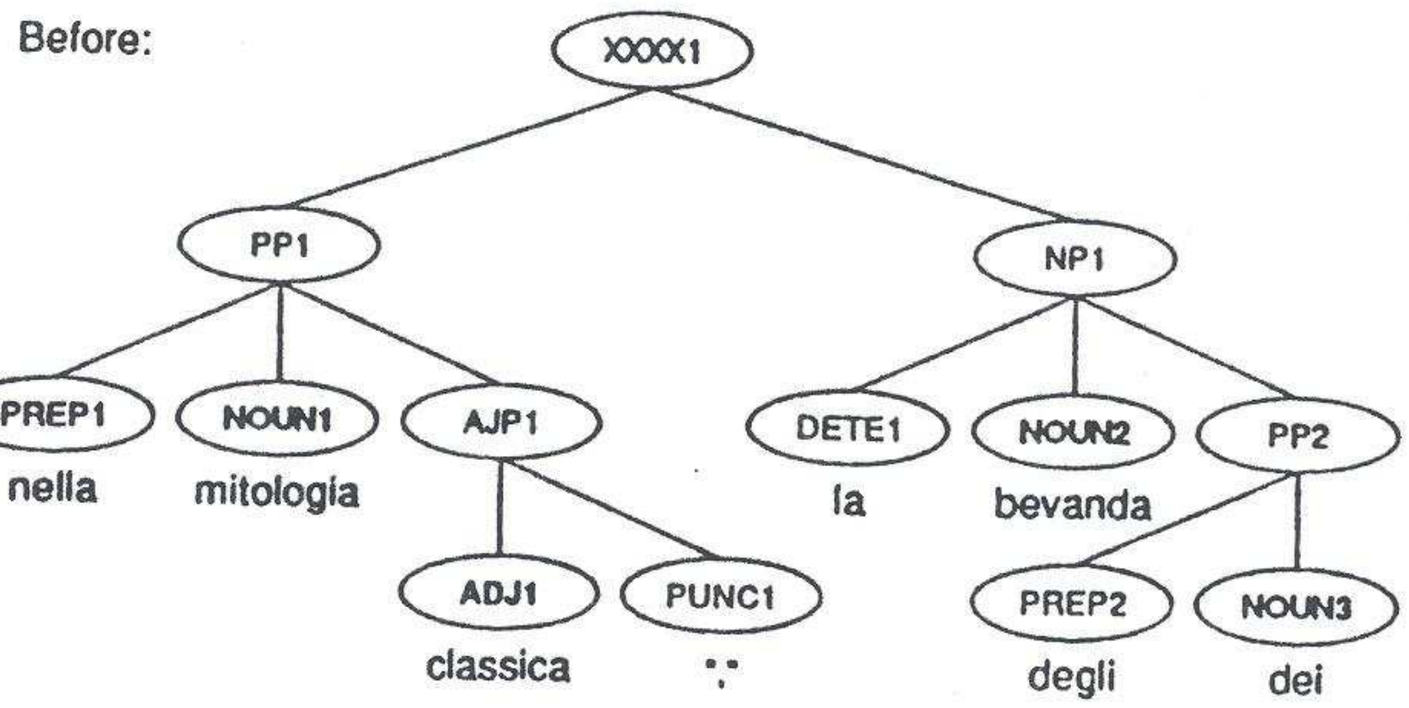


As an example of the refinement to reduce ambiguity, consider the GARZANTI definition of "computazione" (*computation*): "atto, effetto del computare"(*act, effect of computing*). The first structural description above shows the NP parse for general text. This default analysis shows PP1 "del computare" to be attached to the closest available head, "effetto", while the alternative attachment site is marked with a question mark (this illustrates the general strategy adopted within the general grammar for packing attachment ambiguities within the same structure). The second parse below shows the resolution of the PP attachment ambiguity; PP1 now modifies the coordinated nominal phrase covering the coordinated genus terms.

This refinement is made when a prepositional phrase or an infinitival clause post-modifies coordinated head nouns that are the top nodes of the syntactic analysis. This is the typical pattern of the definitions of deverbal nouns and the PP indicates which verb the definiendum is derived from. The lexical and syntactic conditions which make the disambiguation possible are defined in the post-processor to the general text analysis.

The second refinement, instead, concerns the robustness of the system in absence of a complete parse. This can be seen and faced from two different perspectives: the first is dictionary specific and deals with input which would be considered ungrammatical outside the context of dictionary definitions; the second copes with incomplete knowledge of language by exploiting the general technique of 'fitted parsing' provided by the parsing system for handling ill-formed input (JENSEN et al. 1984). Both cases are handled by the 'fitting procedure', already existing within the parsing system, either by using it as it is for coping with grammar or dictionary gaps, or by adapting it for dealing with parsing failures specifically due to peculiarities of dictionary language.

An example of this second kind of refinement, and particularly the dictionary specific one, follows. Dictionary definitions are quite often formulated as condensed fragments of real texts, with elided elements which make the definition syntactically ill-formed and interpretable only by reference to a wider context. This is the case with noun definitions consisting of a noun phrase pre-modified by a prepositional phrase, where the latter specifies the usage domain of the word sense expressed by the former. The general grammar is unable to produce an NP node covering the whole input string given that the sequence PP NP does not freely occur within ordinary texts. It is the refinement stage that should reshape the analysis and restore it as regular input on the basis of specialized dictionary use. The analysis below of the GARZANTI definition for "nettare" (*nectar*), defined as "nella mitologia classica, la bevanda degli dei" (*in classical mythology, the drink of gods*) exemplifies this kind of refinement.



The first of the two parses above has been generated by the general grammar; the XXXX label at the top node shows that the parse is incomplete. The second has been rebuilt during the refinement stage: the XXXX has been replaced by the proper label NP. In this case, knowledge of dictionary

peculiarities resolves the initial partial parse and converts it into a complete and successful analysis. But not all incomplete parses can be so easily restructured. Others are due to gaps in the system with respect to lexical as well as phrasal construction knowledge. These cases are handled by facilities in the fitting procedure provided by the system to cope with unrestricted input. When the grammar is unable to produce a complete analysis, then a reasonably approximate but incomplete structure is assigned to the input. Such a rough parse can still be used as input for further processing stages and for the extraction procedure itself (as we will see later in the semantic analysis section).

In the table below, we report statistical data which support our decision to use, in this initial analysis stage, a general purpose grammar whose output is then revised, disambiguated or reshaped, on the basis of peculiarities of dictionary language.

	no. of parsed definitions	average no. of words per definition	no. of parses	%
Garzanti Noun definitions	997	9	0	23
			1	65
			>1	12
DMI Noun definitions	403	6	0	14
			1	73
			>1	13
Garzanti verb definitions	614	5	0	9
			1	75
			>1	16

The percentages reported above regard the parsing performance of the general purpose grammar for a definition corpus differentiated on the basis of the part of speech of the definiendum and of the dictionary. The average number of words per definition evidences the different degree of complexity of the definitions, according to the dictionary they are extracted from or the part of speech of the words being defined.

The Italian grammar failed to provide complete parses for about a

quarter of the definitions or even less (see in the table how this percentage differs according to the part of speech or the dictionary). An improvement of about 10-15% was achieved during the refinement stage (the improvement refers both to complete but ambiguous analyses and to incomplete ones). For the unresolved incomplete parses, a different extraction procedure has been partially experimented, and partially hypothesized. Because of this robust strategy, the extraction procedure can be applied to the entire corpus of definitions, without the worry that incomplete parses would affect the extraction of semantic information. Some information is extracted in any case; in the worst case the information is not very deep or detailed (at least the genus term is always extracted). The results are differentiated by degree of detail, but the extraction procedure never fails to produce some results.

3.2 Semantic Analysis of Dictionary Definitions

As described above, both parser and grammar for the syntactic analysis of dictionary definitions are domain independent. The only dictionary specific intervention has been aimed at tailoring the syntactic parsing with respect to the dictionary language.

However, the domain dependency of the semantic parser has to be evaluated according to the kind of information to be extracted and a single straightforward answer is not possible. First of all, the semantic information which can be extracted from definitions varies depending on the part of speech of the definiendum; this led to the decision to develop, for this level of analysis, different parsers for each individual part of speech to be considered. Although the individual parsers can be partially derived one from another, the general design of each is independent, given that it is subordinated to the kinds of semantic information to be acquired and their different possible realizations in natural language form within definitions. The technique and the methodology adopted for the semantic analysis of meaning descriptions described in this section have been so far experimented on noun and verb definitions. In the following, the extraction strategy will be illustrated with reference to the parser developed specifically for handling noun definitions.

Let us now consider the domain dependency of the semantic parser; this varies according to the different kinds of semantic information to be extracted, which in the case of noun definitions is distinguished respectively in first, second, and third order relations (see HAGMAN 1991). First order relations typically refer to hyperonym and synonym relations, while second order relations are related by 'indirect' links expressing indirect hyperonym relations, such as ELEMENT_OF, SET_OF, AMOUNT_OF,

PART_OF. Both first and second order type relations are used to build taxonomies (to be intended here in a broad sense, that is including indirect taxonomic links), and are domain independent. Their number seems to be fixed in the case of first order relations, and to be restricted but subject to (few) possible extensions in the case of second order ones. The situation with respect to third order relations is more complex.

The set of third order relations is open; they are all extracted from the differentia part of the definition and correspond to the properties characterizing the definiendum. As such, they are often typical of a particular semantic domain, and are to a large extent identified and defined along with the definition of a lexical subset corresponding to a given semantic domain. However, this is not always the case. There are semantic relations, still located within the differentia part of the definition, which are domain independent; this is the case, for instance, of the SIMILAR2 relation, describing the definiendum by referring to a kind of prototype (i.e. something which is defined as similar to something else for its use, function, or features).

Moreover, there are many relations which have been extracted along with the analysis of a given semantic domain, but which are also valid for other semantic domains. In spite of the fact that they can be valid for more than one semantic domain, these relations cannot be considered as domain independent; the same property can be related to more than one semantic domain, and therefore be shared by individuals belonging to different classes (and this does not entail its general validity for all noun entries). Typical examples are SHAPE, SIZE, COLOUR, as well as USED_FOR, USED_IN relations. Although in our experience these relations emerged along with the analysis of the Food and Drinks dictionary subset, they can be easily extended to other semantic domains (but not to all). There are also relations which are prototypically domain specific (i.e. without considering metaphorical usages) and very unlikely to hold for other semantic domains; this is the case, for instance, of TASTE, restricted to the Food and Drinks subset.

The architecture of the semantic parser has been designed by differentiating the part coping with domain independent relations from the part in charge of dependent ones. Therefore, extending the semantic parser to other semantic domains should not entail substantial changes, but just the addition of the domain specific conditions for the third order relations. However, the extension to other semantic domains does not necessarily imply the addition of conditions specific to the new semantic domain.

As stated before, there are relations which are valid for more than one semantic domain. When extending the semantic parser, if the new

relation to be detected has already been properly expressed for another previously treated semantic domain, nothing else needs to be done (this semantic relation must just be recorded as relevant for the domain under consideration).

On the contrary, there are relations whose recognition has to be disactivated because of conflicts arising between conditions related to different domains. This happens when the conditions to be checked are the same, but the interpretation they are associated with differs. Consider, for example, the different interpretation associated with an adjective such as "maturo" which in Italian can be referred to fruits (with the meaning of *ripe*) as well as to persons (with the meaning of *mature*); this adjective can be a value of more than one semantic relation, and the ambiguity will be solved by knowing the semantic domain we are operating in.

However, ambiguities of this kind can still remain even within the same semantic domain, although they have been to a great extent reduced by differentiating the extraction procedure for the different semantic fields. This is the case of an adjective such as "salato" (*salty*), which within the Food domain can be interpreted either as a value of TASTE, or as a kind of PRESERVATION_PROCESS, or as a simple INGREDIENT (and here the final decision can only be made through the use of human intervention). From these brief remarks, it follows that the more the semantic domains considered, the fewer the relations to be added; the additions will be limited to only the domain specific relations. Therefore, the domain dependency of the procedure as a whole will decrease as the number of semantic domains treated increases.

The differentiated domain dependency of the semantic relations to be detected within the definitions is one of the factors which influenced the parser architecture. Other issues, related to the patterns and the structures to be searched through the definition text in order to extract the semantic information, had also to be considered in devising the semantic parser architecture.

First, there are the kinds of information on which the extraction of semantic information from parsed definitions is based, namely syntactic structures and lexical items. These elements, which can be variously combined, are formalized in the form of patterns used for triggering the recognition of a given semantic relation and consequently the extraction of the value to be assigned to it. On the one hand, there are semantic relations whose extraction is based only on the syntactic structure associated with the definition; this holds only for first order relations (i.e. ISA, SYN, and TGT) whose value is the syntactic head of the top noun phrase, that is the one covering the whole definition. On the other hand, second and third order relations are extracted by means of patterns taking into

account both syntactic structures and lexical items.

For this second group, a distinction can be made on the basis of the value to be assigned to the semantic relation detected: we either have patterns introducing the value of the semantic relation, or patterns, and namely the lexical part of them, which are themselves the value of the semantic relation identified.

The first case can be exemplified with the MADE_OF relation, conveyed by phrases such as "a base di", "fatto con" (both can be translated as *made with*), "costituito da", "formato da" (both can be translated as *formed by*), followed by a noun phrase; here the value of the relation is the head of the complement of the pattern (and this makes the advantage of operating on syntactic structures instead of on the raw definition text clearer). In the second case, the value of the semantic relation detected is the head of the pattern itself; this happens, for instance, with colour adjectives which are at the same time the pattern for identifying the semantic relation and its value.

Moreover, for each pattern to be recognized, the embedding level must be specified. There are patterns to be identified only at the top level of the syntactic structure assigned to the definition; for instance, patterns corresponding to SET_OF, MEMBER_OF, TYPE_OF relations include, together with the lexical conditions, the specification of their position as head of the top noun phrase.

Similarly, there are patterns to be found only at a lower level, among the modifiers of the head of the top noun phrase; this holds for COLOUR or SHAPE patterns whose lexical conditions, when satisfied at the top level, do not give rise to the application of the corresponding patterns. From this, it follows that the recognition of trigger words is always subordinated to tests at the syntactic level, concerning the embedding level as well as the existence of given syntactic structures.

The precedence of tests at the syntactic level over lexical tests is also motivated by another reason. From the syntactic point of view, the same semantic relation can be expressed in the definition context by means of different syntactic constructions. On the other hand, the same syntactic construction is used to express different semantic relations; from this it follows that the single semantic relations are conveyed by the trigger words.

For instance, the recognition of the syntactic structure of a noun post-modified by a prepositional phrase headed by the preposition "di" (*of*) at the top node level is the first step towards the detection of a second order relation such as TYPE_OF, SET_OF, MEMBER_OF as well as at other levels of a third order relation such as MADE_OF, or ORIGIN. The kind of relation is afterwards identified on the basis of lexical tests on the head

of the top noun phrase (for the second order relations above) or on the semantic relation between it and the head of the noun phrase governed by the preposition "di" (for the third order relations above).

Therefore, the architecture of the semantic parser is the result of trying to apply as productively and economically as possible quite a large number of semantic patterns to the output of the syntactic analysis. As we saw above, domain dependent patterns should be separated from domain independent ones to facilitate the extension of the procedure to new semantic domains. Syntactic conditions concerning the embedding level as well as the existence of given syntactic constructions should be tested before applying the lexical tests characterizing the single semantic relation. These different levels of tests led to the distinction between 'pre-patterns' – testing the syntactic conditions – and 'patterns' – operating at the lexical level. All this suggested a top-down, left-to-right approach moving over NP, PP, ADJP, and VP blocks in their syntactical hierarchy and keeping track of what has been read and from where the current embedding has originated.

When implementing the parser, the semantic patterns have been divided into groups according to their typical syntactic realizations and positions in the syntactic parse tree. Each 'syntactic' group, defined in terms of syntactic structures with conditions of embedding and special words, could be viewed as another type of tree, an algorithmic 'tree of conditions' in which there are common trunks of conditions for all patterns of a particular group and branches of patterns with extra conditions leading to twigs of single patterns with their very specific conditions regarding the presence of special trigger words, alone or combined with others. The result is that each algorithmic tree of conditions contains syntactically similar patterns for a number of different semantic relations and each semantic relation can be represented by various patterns which are thus situated in different parts of the tree accompanied by a set of conditions stipulating where in the parse tree they are valid.

Since the algorithmic tree with its trunks, branches and twigs corresponds more or less to the semantic content of the definitions as expressed in first, second and third order relations, an adaptation of the parser to other subsets merely means a trimming of the tree (for a detailed description of the semantic parser see HAGMAN 1992). It should be clear at this point that a parser designed in such a way is data-driven, and easily open to expansions and updates. As already mentioned, new semantic domains can be approached without substantial changes. But as the research continues within the same semantic domain, the results of the semantic analysis can also be easily corrected and refined.

The results produced by the parser described above can be divided into

different classes. The first distinction is linked to the definition part from which the information is extracted: as already stated, taxonomic information is derived from the genus part, while the other semantic relations are extracted from the differentia part.

The extraction procedure operating on the genus part is applied to all definitions, those successfully parsed as well as those without a complete analysis. In the latter case, the information extracted is marked as derived from an incomplete structure, so that it can be manually revised afterwards at the taxonomy building stage. By allowing the genus extraction to apply to incomplete parses as well, the coverage of the semantic analysis is not subordinated to the coverage of the syntactic one. Such a robust procedure, overcoming the variability of parsing performances at the syntactic level, is giving good results; we obtain about 99% of genuses automatically and most of them are correctly identified.

At this stage of research, the procedures taking into account the differentia part of the definition only operate on complete analyses. We are thinking of extending the extraction of the differentia information to incomplete analyses, but this entails a different extraction strategy, based not only on patterns relying on the syntactic structure but also operating at the string level. The kinds of results derived from the differentia part can be distinguished into two classes (the number of classes varies depending on the point of view: from the extraction point of view they are two, while from the LKB point of view, as we will see later, they are three): semantic relations safely indentified on the one hand, and intermediate parsing results on the other.

In the case of safely identified semantic relations, the information extracted is assigned as values of semantic relations, previously discovered and defined as relevant with respect to a given semantic domain. These relations are not necessarily directly related with the definiendum, i.e. they are not always features of the definiendum but they can also be interpreted as further specifications of the words extracted as values of semantic relations. Features such as COLOUR, TASTE, or SIZE are often specified within definitions with relation to other words, standing in some other relation (for instance, MADE_OF, HAS_PART, and so forth) with the definiendum. An example of this is the definition of "caviale" (*caviar*), i.e. "alimento costituito da uova di storione e di altri pesci, salate e lavorate" (*a food made up of the eggs of sturgeon and of other fish, both salted and processed*), where "salate e lavorate" (*salted and processed*) do not refer to the definiendum, but to the "uova" (*eggs*).

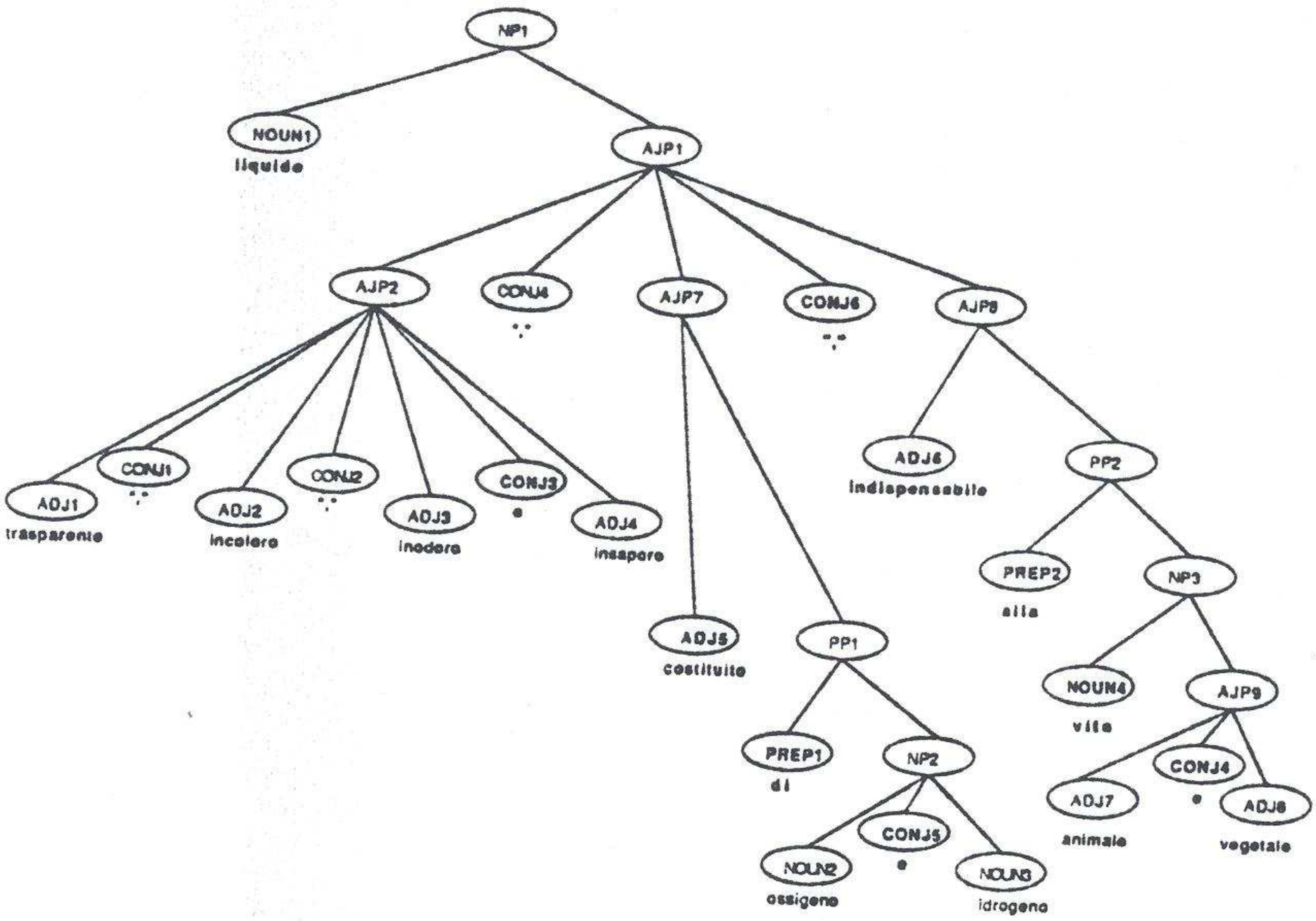
This is possible because we are operating on the syntactic description of the definition, not on the raw sequence of words; in particular, this assignment reflects the attachment point of modifiers. As we saw at the

syntactic level, not all attachment ambiguities can be solved on the basis of the peculiarities of the dictionary language; therefore, some of these embedded relations, the ones based on ambiguous attachments, should be revised interactively afterwards.

But it is not always the case that a safe semantic interpretation can be achieved. For words and constructions which are not listed as triggers for any particular semantic relation, the choice has been that of storing the intermediate parsing results as a starting point for deriving new relations, for better understanding some complex ones, or for making the patterns more adequate and expressive. Another natural advantage of this choice is that the syntactic analysis does not have to be replicated.

These intermediate parsing results are not the same as the output of the syntactic analysis stage; they result from a further step, performed at the semantic analysis stage, which abstracts from the superficial representation of the semantic information within the definition, and formulates it in a more abstract form, specifying the relationships between head words of phrases and their arguments and/or modifiers. Therefore, the structures produced at this stage have generic attribute labels such as ADJ(ectival)_MODIFIER, REL(ative)_CL(ause) _MODIFIER, P(repositional)_P(hrase)_MODIFIER, or OBJECT and SUBJECT, reflecting on the one hand the role of the constituent, on the other the form of its syntactic realization.

In the following, the two stages of the extraction procedure are illustrated with an example. Consider the GARZANTI definition for "acqua" (*water*), sense 1: "liquido trasparente, incoloro, inodoro e insaporo, costituito di ossigeno e idrogeno, indispensabile alla vita animale e vegetale" (*transparent, colourless, odourless, and tasteless liquid, composed of oxygen and hydrogen, indispensable for animal and plant life*). The result of the syntactic analysis is shown on the next page.



The output of the second step of the extraction process for the same lexical entry follows. The semantic information extracted from the definition is presented in the form of a semantic frame. The value of the IS_A attribute is the genus term, while all the other attributes refer to the differentia, that is to those elements which complement, restrict, and further describe the genus term, with the result of characterizing the definiendum with respect to its hyperonym. This output also shows an example of an intermediate parsing result labeled as ADJ_MODIFIER.

IS_A	'LIQUIDO'	
APPEARNC	'TRASPARENTE'	
COLOUR	'INCOLORO'	
SMELL	'INODORO'	
TASTE	'INSAPORO'	
MADEOF	'OSSIGENO'	
	'IDROGENO'	
USEDFOR	'VITA'	
	ADJ_MODIFIER	'ANIMALE'
		'VEGETALE'

3.3 Taxonomies

3.3.1 Genus Disambiguation

As already stated, the genus disambiguation procedure works interactively; it is the only completely interactive procedure in the ACQUILEX-Pisa system. The necessity for a fully interactive procedure rather than a semi-automatic procedure, proposing solutions which must be checked and revised interactively, was caused by the lack of explicit semantic information (such as the LDOCE semantic codes) within the Italian dictionaries.

Although the procedure has been used principally for genus term sense disambiguation, it can be used on any term of interest resulting from analyses on the lexical databases. It includes functions to correct the automatically assigned genus term when necessary, and to add normalized terms or conceptual labels to the genus term to permit cross-dictionary or cross-language links (for a detailed description of the procedure see MARINAI & PICCHI, 1991). In this context, our interest is focused on examples of dictionary inconsistencies, which are made evident from the extraction procedure, and appear as crucial with respect to the representation within the lexical knowledge base.

We find words used within definition texts which are not given headword status in the same dictionary. For instance, the word "latticino" (*dairy product*) appears twice as a genus term in the GARZANTI definitions, although it does not appear itself as an entry ("latticino" is recorded only as a variant form of another entry, "latticinio"). This case is solved within the lexical knowledge base by making the entries with "latticino" as genus inherit from an abstract concept instead of from an actual – in this case nonexisting – lexical entry.

Another example of inconsistency, similar to the previous one, concerns word senses which are attested in the usage within definitions, but are not given the status of word sense within the dictionary. This is the case of "bacon" (*bacon*) and "ventresca" (*white meat tunny*), respectively defined in the GARZANTI dictionary as follows:

bacon (sense 1): "ventresca di maiale affumicata" (*smoked meat taken from the stomach of a pork*);

ventresca (sense 1): "la carne del ventre del tonno" (*meat taken from the stomach of a tuna fish*).

In the definition of "bacon" the genus "ventresca" is used in the more generic sense of meat taken from the stomach of any animal whatsoever, which is not attested in the GARZANTI dictionary. This sense does not

take into account the origin of the meat, i.e. the animal; in linguistic terms, we would say that the feature ORIGIN has been neutralized.

Even if this kind of inconsistency can be easily solved in the context of the lexical knowledge base (by overriding, in the "bacon" entry, the ORIGIN value inherited from "ventresca" – which is 'tuna fish' – with the new ORIGIN value specified within the differentia part – which is 'pork'), the problem remains as far as genus disambiguation is concerned. In the example above, the non attested sense was simply more generic, and it subsumed the recorded one; there was no conflict between the two, just a different degree of specification. But in principle a non attested sense may be in conflict with the existing ones; in this case, the only possible solution with respect to the LKB representation is to again make the entry inherit from an abstract concept.

Both kinds of inconsistency reported above could be given an easy solution – which we expect to work often but not always – by merging and integrating the data coming from different dictionaries. This is confirmed by the fact that one of the examples cited, derived from the GARZANTI dictionary, would have been solved by data merged from GARZANTI and DMI. The entry "ventresca" in the DMI is recorded as having two different senses, one with ORIGIN = 'tuna' and the other with ORIGIN = 'pork'. What we are working towards is a semi-automatic merging of the data coming from the two Italian Dictionaries.

3.3.2 Tools for Taxonomy Building and Browsing

A top-down Taxonomy Lister has been developed in order to give an overview of all the material and to facilitate parallel linguistic research. Figure 2 shows the type of output that is created by the Taxonomy Lister, which at this stage exists as a Prolog prototype operating on the extracted taxonomy data.

This tool cannot really be called a 'browser' if by browser we intend a real-time working device which permits the user to scan the tree in both directions. The output (which can be very complex) is instead written on a plain ASCII file which can be easily scanned in later studies or sense disambiguation sessions (when the senses have not yet been disambiguated).

The definition strings in the figure have been added here by hand in order to show exactly which relations have been considered. In the prototype, only selected semantic information has been extracted and stored – this was principally to respect time and memory constraints. In the near future, we may incorporate this tool into an LDB version – in an environment that already has a user interface, and which is better equipped to host and quickly access the large amount of definitions and

other lexical information constituting an MRD.

Another Prolog implementation – a bottom-up Taxonomy Display – also runs on the same taxonomy data. The tool, which is in the form of a separate program, asks the user to enter a root lemma, to give the maximum number of hyperonyms for each lemma, and to set the number of levels for which the search is to continue. The output is shown in Figure 3. The arcs must be followed from right to left since they lead back from a word to a word at either the same level or a previous level.

The program was started with the parameters {"tetto" (*roof*), 5, 4} and the result is displayed as a tree on the screen, as shown in Figure 3. The tree clearly shows the definition circularities among the lemmas tested. Let us follow a track: "tetto" is a "tratto" is a "linea" is a "contorno" or "sagoma". Both the last two are a "linea" and "segmento" is a "tratto" is a ...

The branches of the tree are derived from both First and Second Order Relations, but only those introducing a hyperonym whose part of is not the same as the definiendum are specially marked – by a dotted line and the lemma written in italics.

3.3.3 Assessing and Revising Taxonomies

Taxonomies have been built on the basis of the genus terms which were automatically extracted in the semantic analysis stage, and whose senses have been afterwards interactively disambiguated. These taxonomies must then be assessed and, when necessary, revised. As already stated, taxonomic information is translated in the lexical knowledge base into inheritance chains, in which the semantic information is passed on from generic to more specific words.

The Lexical Knowledge Base we are constructing, starting from the lexical data extracted from dictionaries, uses a typed feature structure representation system (for which see 4.1 below). The design of the lexical representation language and its implementation have been carried out in Cambridge and are fully described in COPESTAKE, DE PAIVA, SANFILIPPO (1991). The constraints imposed by the inheritance mechanism, which holds for the lexical representation level, influenced the principles behind this revision phase. In particular, the semi-automatically built taxonomies were revised in order to:

(a) assign the most appropriate abstract concept to the top taxonomy entry;

(b) revise genus assignments when they are insufficient or misleading in the movement of information from hyperonym to the definiendum.

The assignment of what we call the 'relativized qualia structure' (RQS) (see CALZOLARI 1991), i.e. semantic information expressed in a frame-like structure, to the top taxonomy entry is crucial, given that it determines the properties which will be passed through the inheritance chain to all the taxonomy entries. This choice must balance the actual content of the definition of the top taxonomy entry against the properties shared by all hyponyms, or at least most of them. It is very often the case that the actual content of the top entry is undefined in many respects; this is due, usually, to the vagueness of the definition of generic terms.

However, if we accept this vague definition and we assign it as the common information core shared by all taxonomy entries, we will be obliged within the definition of the single entries to specify features which were not explicitly expressed within the definition of the top entry but which emerge as common to all hyponyms. As a result of this conflict, we chose to assign the RQS (or semantic feature structure) of the top entries of the taxonomies by means of manual encoding, not necessarily reflecting the original formulation of the definition.

In those cases in which the RQS assigned on the basis of the actual definition and the data-driven one (i.e. derived following a bottom-up strategy) did not coincide, we created two different entries: one defined on the basis of the actual content of the definition of the top taxonomy entry, the other derived on the basis of the properties shared by its descendants. This second abstract entry will be the one used as the top taxonomy entry, while the real one will remain to testify the gap existing between the dictionary definition of words and their actual use.

Therefore, in order to avoid repeating the specification of the same feature value within all the members of the taxonomy, we decided to assign to the top abstract entry the RQS with the specifications which most frequently recur within the hyponym chains. For instance, the top entry "alimento" (*food*) defined as "quanto serve a mantenere in vita e a far crescere animali e vegetali" (*that which serves to maintain animals and vegetable alive and permit them to grow*) takes the RQS which has been defined for the type FOOD_ART_OBJ, i.e. artifact food in the form of object.

This attribution has been made arbitrarily on the basis of the most recurring features within the taxonomy. If the RQS attribution had been made on the basis of the actual definition, the RQS would have been so vague that, for each single taxonomy entry, the specifications of 'edible', 'artifact', and 'object' would have had to be added. In spite of this manual encoding of the top entries, we have been obliged to interrupt the inheritance chain for a few entries. This is the case of the "liquido" (*liquid*) taxonomy, where the RQS of a DRINK_ART (artifact drink) has been

assigned to the top entry "liquido" (*liquid*) following the bottom-up approach, while its hyponym "acqua" (*water*) has the RQS of DRINK_NAT, i.e. natural drink.

With reference to the revision of the genus assignments as mentioned in point (b) above, the following case must be taken into account. Often, information contained within the differentia part of the definition, when linked to the genus, causes the shifting of the taxonomy for the definiendum. For instance, "cacio" (*cheese*) is defined as "latte cagliato ..." (*curdled milk*).

However, it does not seem correct to make "cacio" inherit the RQS of its genus "latte", as it would then inherit the liquid state of milk, which is a substance and not an object, natural and not artifact. The presence of "cagliato" within the differentia is the information which justifies the taxonomy shifting. From the viewpoint of the LKB representation, it would be necessary to have the possibility of defining 'complex genus terms', such as "latte cagliato"; for the time being the solution adopted is that of reassigning an abstract concept such as genus, in order to avoid the inheritance of odd features from a genus which when considered alone is misleading. This is a problem which usually arises when genuses are not single words, but some kind of multi-word expression.

A final issue to be considered in this context is that of the differentia-based taxonomies. The Food and Drinks subset is made up not only of entries belonging to taxonomies in which the *edibility* is inherited from the head taxonomy (for instance, "bevanda" (*drink*), "cibo" (*food*), "alimento" (*food*)) but also of the entries belonging to taxonomies in which the edibility is only deduced from information contained in the differentia of each entry (see also SPANU, forthcoming). This is the case of entries belonging to the taxonomies of "animale" (*animal*), "pianta" (*plant*) and "prodotto" (*product*):

coniglio (*rabbit*): "mammifero roditore commestibile ..." (*rodent edible mammal ...*);

cicoria (*chicory*): "pianta erbacea coltivata per le foglie commestibili" (*herbaceous plant cultivated for its edible leaves*);

salume (*cured pork meat*): "... ogni prodotto della lavorazione della carne suina" (*... all products derived from cured pork*).

The problems connected with these taxonomies are of a different nature. "Pianta" (*plant*) and "animale" (*animal*) behave in the same way because their edibility senses are examples of sense-extension. For example, from the point of view of linguistic (dictionary-based) taxonomy,

"coniglio" (*rabbit*) only belongs to the animal taxonomy whereas from the point of view of the real world it belongs both to the animal taxonomy and to the Food subset. The same is true of "cicoria" (*chicory*) as it is an edible plant.

One way to respect the integrity of the taxonomy and at the same time to preserve the information connected with edibility is to apply lexical rules to the relevant entries in order to produce LKB lexical entries belonging to the Food subset. This solution especially concerns the taxonomies of "pianta" (*plant*) and "animale" (*animal*) (a rule to transform from animal to food has already been implemented, the animal-grinding rule). Applying the lexical rule is the best solution for these taxonomies as it permits the entry to belong both to the Food subset and to the original one.

Instead, the problem concerning the "prodotto" (*product*) taxonomy, presents itself in another way. Under the same genus term, different sorts of taxonomies are constructed. The complete taxonomy of "prodotto" actually includes both entries belonging to the Food and Drinks subset and entries belonging to other semantic fields. In this case, the RQS assignment must be decided on the basis of the analysis of the differentia, in the sense that in the differentia there are elements which can be used as clues on how to assign the word sense.

The modalities for revising the taxonomies described above have been considerably influenced by the way the taxonomic data is used in the LKB, i.e. to construct the skeleton of the inheritance chains. At this point the problem of the lack of equivalence between linguistic and real world taxonomies emerges.

3.4 Filtering the Results of the Extraction Procedure

So far (see in particular 3.2 above), we have seen the results of the extraction procedure as divided up into two classes: one to which a reliable and final semantic interpretation has been assigned, and one which needs to be further processed, the so-called group of intermediate parsing results. This can be seen as a consequence of the extraction procedure as an ongoing process. But when we decide to represent the extracted information within a lexical knowledge base, only the first class is taken into account. At this point, the information contained in this class falls into two main groups: taxonomic information on the one hand, and semantic relations extracted from the differentia part on the other; the second group can be further distinguished on the basis of the existence of a suitable representation of such information within the lexical knowledge base, since not all the information extracted can be assigned an appropriate representation

in the LKB as it currently exists.

This filtering phase is the preliminary step of the conversion procedure, which will be illustrated in the following section. It prepares the input for the conversion process, by excluding the intermediate parsing results, and the 'intermediate' semantic relations which have been identified; the latter are defined as intermediate because despite their correct and final interpretation they cannot be represented within the currently existing lexical knowledge base.

4. The Lexical Knowledge Base: Formulating Lexical Entries as Typed Feature Structures

The extraction process illustrated above can be seen as the first step in the translation from a 'natural' knowledge representation language (NL) to a 'formal' one. All the information extracted must, at this point, to be openly and clearly expressed. What has been extracted for each word sense of a lexical entry must be inserted into the general framework of the lexicon, and must be linked to other lexical entries via taxonomic relationships as well as via shared phonological, morphological, syntactic, or semantic features (in the latter case, not necessarily entailing a taxonomic link). For this purpose, a formalism was needed which would permit the representation of different aspects of lexical information (phonological, morphological, syntactic, and semantic) and encoding of the inter-dependencies among word senses of lexical entries (the word sense is the minimal unit of information of our LKB).

The lexical knowledge base we have been constructing, beginning from the lexical data extracted from machine readable dictionaries, uses – as stated above – a typed feature structure representation system. The lexical representation language has been designed and implemented at Cambridge (for a detailed description see COPESTAKE, SANFILIPPO, BRISCOE, DE PAIVA 1991). In the following sections, after giving a brief overview of the typed feature structure formalism as it has been implemented within ACQUILEX, we present our experience with the encoding of dictionary entries using this formalism, pointing out – whenever possible – the advantages and disadvantages that, so far, have emerged.

4.1 A Brief Overview of the Typed Feature Structure Formalism

Typed feature structure representation systems are unification-based formalisms augmented with the notion of type. Feature structures (Fss) are the basic data structure used in this class of formalisms, describing linguistic objects in the form of bundles of attribute-value pairs. Since

different bundles of attribute-value pairs make sense for different classes of objects, feature structures have been divided into different types. Therefore, feature structures in this context become typed feature structures; in particular, they are given type names, standing for classes of linguistic objects, which are described by the feature structures associated with them. For each type, a fixed configuration of attributes is defined and constraints are imposed over the range of values each attribute can take; these are the appropriate features for the intensional definition of a given class of objects. Values of attributes refer themselves to types, either atomic or complex (i.e. feature structures).

Types are ordered in a type hierarchy with subtypes and supertypes, according to a subsumption relation: a type t_1 is a subtype of another type t_2 if t_1 contains at least the same information as t_2. This implies that the feature structure of type t_1 inherits all features (i.e. attributes and restrictions over their range of values) defined for its supertype t_2. Subtype specialization can be obtained by adding more specific information, which can be defined either by adding new specific attributes – local to the subtype – or by further restricting the range of possible values of an attribute already defined in the supertype. All the information contained in t_1 must be fully consistent with that of t_2. This is the inheritance mechanism for transporting information from a type to its subtypes; this kind of inheritance is monotonic.

In the ACQUILEX TFS system, a type can inherit from one or more supertypes. When a type inherits from one supertype only, the mechanism is as described above. On the other hand, when a type inherits from more than one supertype, the inheritance – called 'multiple inheritance' – is more complex and constrained, given that it is obtained by unifying the feature structures associated with the supertypes. The type hierarchy defines an ordering of the types and specifies which types are consistent. Only feature structures with mutually consistent types can be unified, and two types which are unordered in the hierarchy are assumed to be inconsistent, unless the user explicitly specifies a common subtype. Concerning this last case of a subtype common to more than one type, the system allows the unification of FSs only if the meet of their types exists. This multiple inheritance mechanism is further constrained by the condition that any consistent set of types must have a unique meet.

The first step to be taken before representing any linguistic object in the form of TFS is the declaration of types. The basic building blocks out of which feature structures, corresponding to linguistic objects, are constructed must be defined. In this phase the definition concerns:

- the types and the inheritance hierarchy defined over their set;

- the features which are appropriate for each type and their appropriate values.

After this brief outline of the basic principles behind TFS formalisms, let us consider whether and how the lexicon can be encoded as a TFS hierarchy. Most of the information contained in a fully specified lexical entry is not unique to this given entry. Related lexical items will share some of the properties defined for this entry. Exploiting this structured organization of the lexicon, lexical entries can be grouped into classes formed on the basis of the properties which are common to all members of the class. This is the reason why TFSs appear as a suitable formalism helping to reduce redundant specifications within the lexicon. The advantages of representing the lexicon as an inheritance network are its succinctness and its tendency to highlight significant clusters of linguistic properties. The typed feature structure representation system permits the specification of information shared by different classes of lexical items only once, within the definition of the superclass parent of these classes. Therefore, all lexical entries having the same property will be members of the same class. On the other hand, the same lexical entry will be a member of different classes, according to the linguistic dimension under consideration (syntactic, semantic, etc.).

In lexicon representation, the hierarchical organization of the dictionary into taxonomies can be exploited by translating them into inheritance hierarchies. But the inheritance mechanism operating within the type hierarchy is too rigid and restrictive for dealing with lexical information. For this reason, a different kind of inheritance has been specifically introduced, the default inheritance (see BRISCOE, COPESTAKE, DE PAIVA 1991). With default inheritance, a class does not inherit all properties from its superclasse(s), but only those properties for which there is no information defined in the class itself. The default inheritance is formalized as default unification of feature structures, which operates on a non-default feature structure to be unified with a default feature structure. Values in the non-default feature structure which conflict with values in the default feature structure are allowed to override the default conflicting values (hence default inheritance is non-monotonic). In this way, default unification never fails. Therefore, the inheritance mechanism within the ACQUILEX LKB is differentiated; the Type System provides a non-default inheritance mechanism, while the default inheritance mechanism holds for lexical representation.

4.2 Conversion Procedure

The crucial step towards representing the extracted information within the lexical knowledge base is the conversion of the reliable semantic information extracted (that filtered at the previous stage and classified as convertible) into the TFS formalism. For this purpose, a semi-automatic conversion procedure, operating at two levels, has been designed and implemented. At the first level, this conversion procedure identifies, through the genus information, the default feature structure from which the lexical entry being defined inherits the general properties. At the second level, it interprets the local information extracted from the differentia part of the definition by assigning it as value of attributes within the specific feature structure. In this way, taxonomic information and other semantic relations are combined and organized in the form of feature structures.

The conversion procedure is divided into two steps:

1. the first step is fully automatic and translates what has been extracted in the form of typed feature structures;

2. the second step is carried out interactively; each lexical entry produced in the previous stage is assessed and, when necessary, revised.

The main issues to be considered at this point are:

- the relevance of the extracted and converted information with respect to the definiendum; as we stated before, we are also extracting semantic information which has to be seen as a further specification of the values assigned to other semantic relations directly related to the definiendum (we could refer to these cases as embedded semantic relations). At this stage, the LKB does not support the representation of information which is not directly linked to the word being described. Moreover, it must also be remembered that not all the ambiguous modifier attachments have been solved during the syntactic analysis stage. From this, it follows that it is necessary to check the relevance of the represented information with respect to the definiendum; this test sometimes leads to discard part of the information extracted;

- the errors in assigning the extracted information as values of a given semantic relation, which can derive from unsolved ambiguities. As we saw in the semantic analysis section, the same trigger word, even within the same semantic domain, can be ambiguous; see, for instance, the multiple interpretation to be assigned to the "salato"

(*salty*) adjective (as TASTE, PRESERVATION_PROCESS, and as INGREDIENT). In order to solve this kind of ambiguity, human intervention is needed.

Therefore, during the interactive step of the conversion procedure, ambiguous elements are interpreted and the relevance of the information extracted with respect to the definiendum is evaluated.

At this point, the entries generated by this long and complex procedure are well formed with respect to the representation language adopted for the lexical knowledge base. In the following section, we illustrate the main problems we have encountered in encoding the semantic information extracted from natural language definitions into the current LKB.

4.3 Representation

Some of the advantages of this formalism for lexical representation have already been mentioned: assigning a hierarchical organization to the lexicon helps to reduce redundancy and, at the same time, to highlight significant clusters of properties within the lexicon. Moreover, it allows a uniform representation of the information at different levels of linguistic description (phonetic, morphologic, syntactic, lexical, semantic), thus blurring part of the traditional dichotomy between grammar and lexicon. With respect to the lexicon, it is also possible to use the same formalism to encode lexical rules for representing linguistic generalizations about sets of lexical entries; this makes the lexicon less redundant and more robust with respect to unforeseen and unexpected uses of words. At a more technical level, it permits consistency checking, which is a necessary feature when encoding large amounts of data, such as lexicons. These are some of the features, widely recognized, which make this kind of formalism appropriate for representing lexical information (on this point, see the final report of the ET-7 project).

The ACQUILEX implementation of the TFS formalism has been specifically conceived for lexical representation, and this explains some of its features; first of all, the choice of differentiating the kind of inheritance at a more abstract level, that of the Type System, and at the Lexicon level. In this way, a general TFS representation system has been tailored to the needs specific to lexical representation.

But in spite of the fact that the formalism has been specialized with respect to the needs of lexical representation, at the moment of translating what was extracted from the definitions into the TFS formalism, a conflict between empirical and theoretical approaches arises. The main problem at the representation stage is how to combine the empirical results of the knowledge extraction process with the theoretical hypotheses made with

respect to the lexical information common to a class of words (see CALZOLARI 1991). Here we observed a gap between the information actually found and what we expected to be represented within the TFS entry. In the following sections, we give some examples of this different expressive capability of two representation languages: natural language on the one hand and, on the other hand, the TFS representation language, as implemented in the ACQUILEX LKB.

(1) *Lack of representation due to the current* LKB *System.* In this section, we list cases of extracted information lacking a corresponding formalization due to the current LKB system. All these cases refer to the representation of disjunction and negation over attribute values, either atomic or complex (i.e. FSs). Although in this context we refer to the specific ACQUILEX implementation, these points are widely recognized as critical and crucial aspects of this class of formalisms since they are very expensive from a computational point of view. Therefore, they should be seen as suggestions for further developments and improvements of the system, being aware, obviously, of their formal complexity.

(1a) *The necessity of handling* AND *and* OR *logical operators within values of attributes.* The situation differs depending on the kinds of attributes.

Attributes whose value is a list, such as CONSTITUENCY (representing the 'constituents' of the object being designed by the definiendum), only permits the expression of the conjunction of the elements of the list. Thus we can correctly represent cases such as, for example, "acqua" (*water*) which is defined as made of "ossigeno e idrogeno" (*oxygen and hydrogen*), or "caffellatte" (*white coffee*) defined as made of "latte e caffè" (*milk and coffee*), i.e. cases for which the conjunction is explicit. But beside these cases we have cases in which the kind of relation linking the single constituents is the disjunction. For example, the cases of "gnocco" (*small dumpling*), defined as made of "farina di patate o di semolino" (*potatoes or semolina flour*), or of "cioccolatino" (*chocolate*), defined as filled with "crema o liquore" (*cream or liqueur*). Currently, these cases are treated in the same way as the preceding ones, using the CONSTITUENCY attribute, with the loss of a pertinent information, i.e. the fact that "farina di patate" and "semolino", "crema" and "liquore" are disjunct ingredients.

The contrary happens with attributes that have atomic types as values. In this case, when the value is a list of atomic types, it is assumed that they are disjunct elements. But, again, this is not always the case. It can happen that more than one value is true at the same time, and this cannot be expressed.

Moreover, it should be observed that until now the only possible spe-

cification of the OR relation has been between atomic types. For the kind of information we extract, it is vital for us to extend this kind of specification to non atomic types.

(1b) *The same need to handle* AND *and* OR *logical operators does not concern only attribute values, but also genus terms.* Here the problem is different, given that the genus information is used to make the definiendum inherit the RQS from its hyperonym. A possible solution would be to generate two subentries (to be formally distinguished) starting from the entry with coordinated genuses. But at this point, again, there is the problem of specifying which kind of relation links the two genuses (the AND or the OR relation).

At the moment, genus terms coordinated by means of an AND/OR relation are treated in the following way: for example, "senape" (*mustard*) which is defined as "salsa o farina di ..." (*sauce or flour of ...*) is recorded under two different taxonomies, that of "salsa" and that of "farina". But the two entries refer to the same word sense, so only one of the two entries will come out when querying the LKB. A possible solution would be, on the one hand, to differentiate the entries on the basis of the genus and, on the other hand, to allow both of them to refer to the same word sense. In this way, it is possible to keep intact the multiple genus information as well the reference to the same entry. But the link between the two genuses remains undefined.

(1c) *Need to handle negation.* Within definitions, features are sometimes expressed in a negative way; we think that in these cases the LKB formalization should reflect the original formulation within the definition. If something is defined as "non bianco" (*not white*), this does not give us the right to say that it is red, pink, blue and so on. In cases in which the negation operates on a term member of an opposition, the inference should be less dangerous, but still arbitrary. It seems to us that, for now, the LKB does not support this type of specification. For instance, "galletta" (*cracker*) is defined as "pasta di pane ...poco o punto lievitata" (*little or not leavened dough*). In this case we have been obliged to discard the information of "non lievitata" (*not leavened*).

(2) *Lack of representation due to the Type System.* In this section, we list cases for which formalization is lacking due to the system of types. They are in principle easier to be solved, given that they do not involve any modification of the lexical representation language and its implementation, but just of the system of types (which is defined by the user). We decided to list them here because they pose crucial inheritance problems,

to be taken into account when building a hierarchical lexicon.

(2a) *Relations of the so-called First Order (*ISA, SYN, TGT*).* The distinction between ISA and SYN cannot be formalized. The synonym relation (SYN) may be deduced from the absence of additional features which define the special characteristics of the definiendum with respect to the genus; but unfortunately this absence of specific features is also very frequent in the case of ISA relations, considering the difficulty of both the extraction and the representation of information from the definitions.

Particular taxonomy relations, like the TGT ('technical genus term', generally consisting of pronouns), entail the inheritance of only one attribute, and not of a complete RQS. In the case where TGT = 'chi' (*who*), the RQS is inherited from HUMAN, but where the TGT = 'che' or 'ciò che' (*which*) it is possible to define the value of only one attribute, i.e. ANIMATE = 'false', without being able to distinguish whether it is concrete or abstract. Moreover in the Type System, ANIMATE is only a relevant attribute in PHYSICAL (used for concrete nouns), whereas for the definition of ABSTRACT such an attribute does not occur. This, using the current Type System, can only be resolved interactively. Therefore, when the definiendum is concrete, the RQS is assigned the value PHYSICAL, with the ANIMATE attribute equal to 'false'; when the definiendum is abstract, the RQS is assigned the value 'abstract'.

(2b) *Relations of the so-called Second Order (*TYPE_OF, SET_OF, PART_OF, AMOUNT_OF*).* The representation of this kind of relation requires a careful study of the inheritance mechanism. Some of these relations are discussed in OESTLING (1992) in which a central problem is the fact that some trigger words are common to more than one relation and these intersections complicate the identification of the relation itself.

(2c) *Relations of the so-called Third Order (corresponding to the properties specific to the definiendum).* It has been often necessary to flatten the semantic information extracted, which, in order to be represented within the LKB, needs to find corresponding features and values within the current Type System. For instance, different relations detected using our extraction procedure (HAS_PART, MADE_OF, OBTAINED_FROM) are all recorded as values of the CONSTITUENCY attribute, although they have a different meaning, and may have different entailments.

Moreover, the current Type System does not support the specification of RQS values for embedded attribute values. At the moment, it is only possible to specify semantic features directly related to the definiendum. Instead, we would also like to specify semantic features for words which

stand in some semantic relation (for instance, HAS_PART, MADE_OF) with the definiendum. Let us take as an example the definition for "acagiù" defined as "albero tropicale dai frutti saporiti" (*tropical tree with tasty fruits*). Here the value "saporito" (*tasty*) of the semantic feature TASTE has to be related – as a restriction – to "frutto" (*fruit*) which is in its turn the value of the HAS_PART relation (this latter is directly related to the definiendum).

4.4 Extension of the Type System to another subset

One way in which a Type System designed to represent the information contained in a given semantic subset can be evaluated is to test its validity for another subset. In this way it is possible to see which types, attributes, and attribute values can be reused as they are, being valid for both subsets, which can be reused if certain modifications are made, which are superfluous for the new subset, and especially what needs to be added. The Type System designed to represent the Food and Drinks subset has thus been tested on the Place subset (SPANU 1992). In particular, we analysed the definitions of lexical entries which refer to places associated to foods, i.e. "agrumeto" (*citrus plantation*), "pasticceria" (*confectioner's*), "pastificio" (*pasta factory*). We present here briefly the results of the attempt to extend the Type System.

The semantic information contained in the definitions of lexical entries, referring to the Place subset, can be represented by means of a list of pertinent attributes. Most of the required attributes are already implemented in the Type System, and only a minority need to be added. The already implemented attributes could generally be reused as they are, but a displacement is necessary for some of them, in order to allow the inheritance of only the pertinent attributes for each type. Some of the new attributes were already proposed during a first phase of the Type System preparation, but discarded because the modality of their application was not always clear. The fact that these attributes reoccur in another subset demonstrate that it will be reasonable to takethem into account in the future. Obviously the extension of the system involves the addition of new types,in this case it would seem necessary to introduce more general types, aswell as more specific ones.

In conclusion, we can state that an extension of the Type System from the Food and Drinks subset to the Place subset is possible without substantial modifications, needing only a few displacements or additions of attributes and types. As far as common attributes are concerned, we can use the already existing attributes, without having to introduce a copy of them.

4.5 Final Remarks

In general, it can be noticed that the rigour and lack of flexibility of the Type System can cause difficulties which are not easy to overcome when one tries to map natural language words into it, which may well be ambiguous and flexible. It is difficult to constrain word meanings within a rigorously defined organization: by their very nature they tend to evade any strict boundary and the solution of one part of a problem often causes another problem to arise.

A final observation should be made on the content of the Type System itself. The current Type System is far too limited with respect to the amount of information which can be automatically extracted from natural language definitions. Part of the work dedicated to the meaning extraction and acquisition phase in Pisa is therefore – as already stated – not exploited at the level of the common Type System. No feature and/or value has been found to fit a rather large amount of data, and often – even when a possible mapping is found – many meaning distinctions, which can be generalised over lexicographic definitions and automatically captured, must be blurred into unique features and values.

The LKB, however, is a powerful tool especially in:

(a) checking the consistency of the data loaded in it;

(b) constraining the lexicographer to well-formed templates for each typed feature structure or its descendants, thus eliminating or reducing incoherency;

(c) providing ways for the automatic comparison of entries along their feature structures.

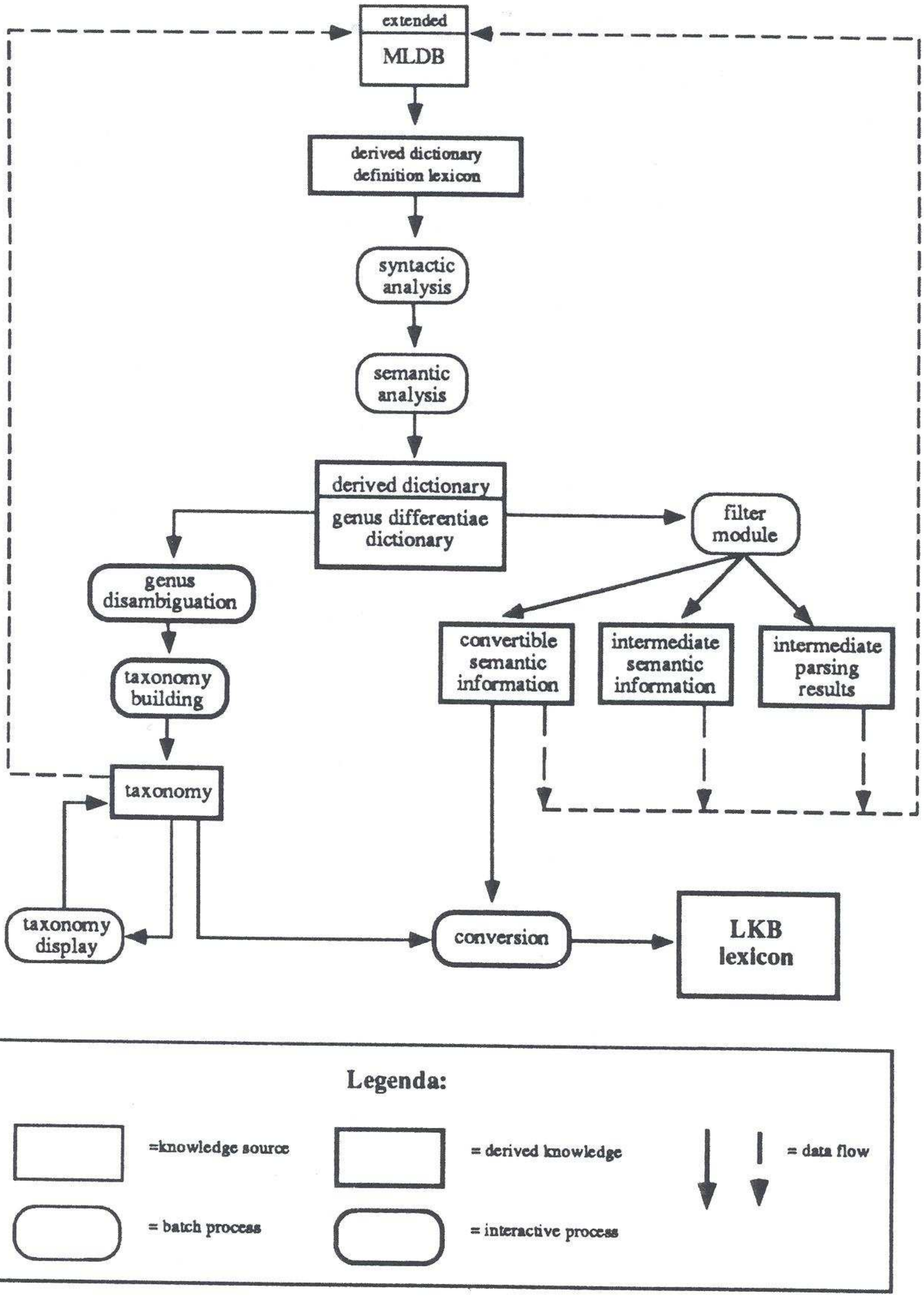


Figure 1: The Acquilex-Pisa System for the extraction and representation of lexical knowledge

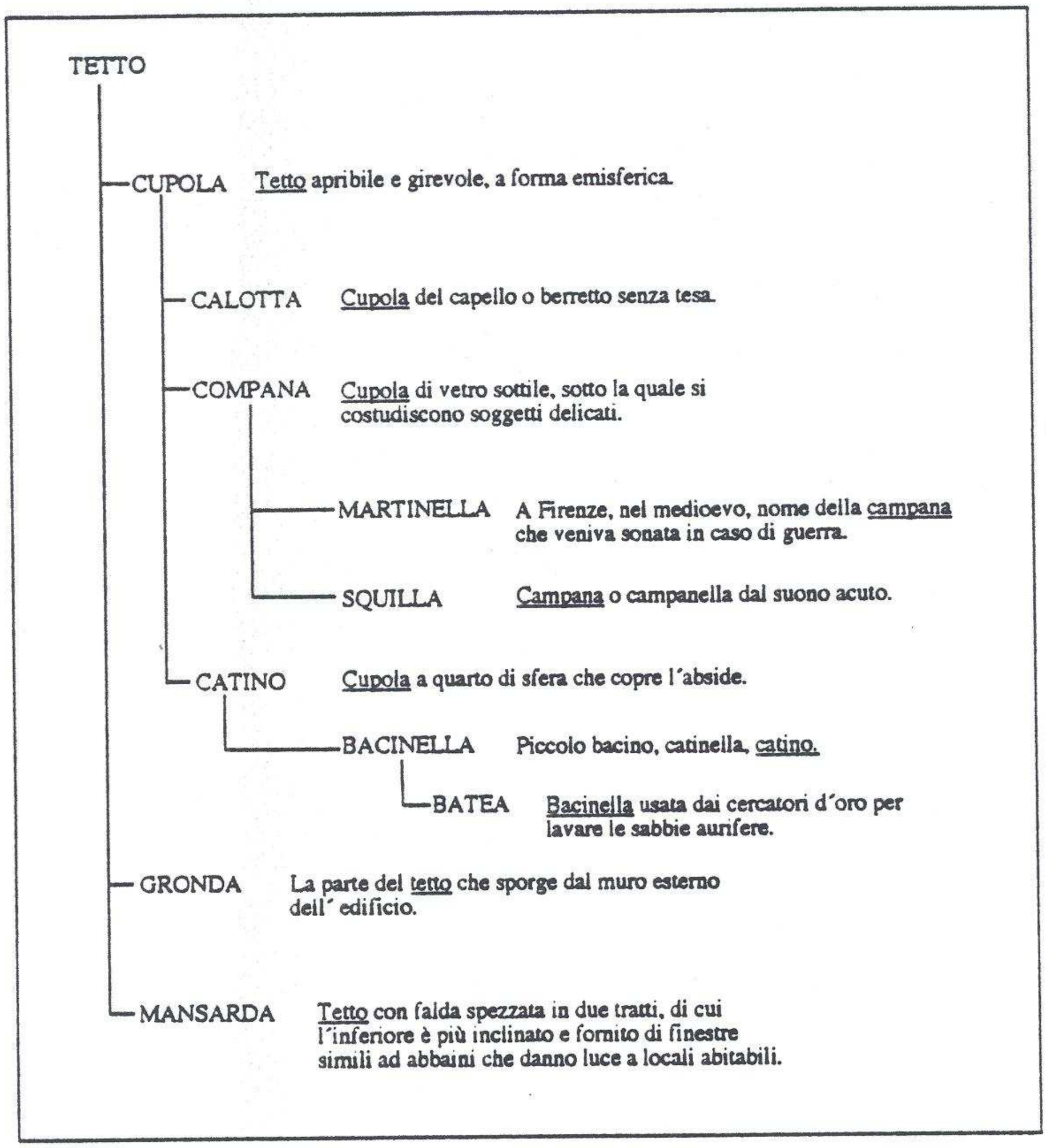


Figure 2: Output from the Taxonomy Lister

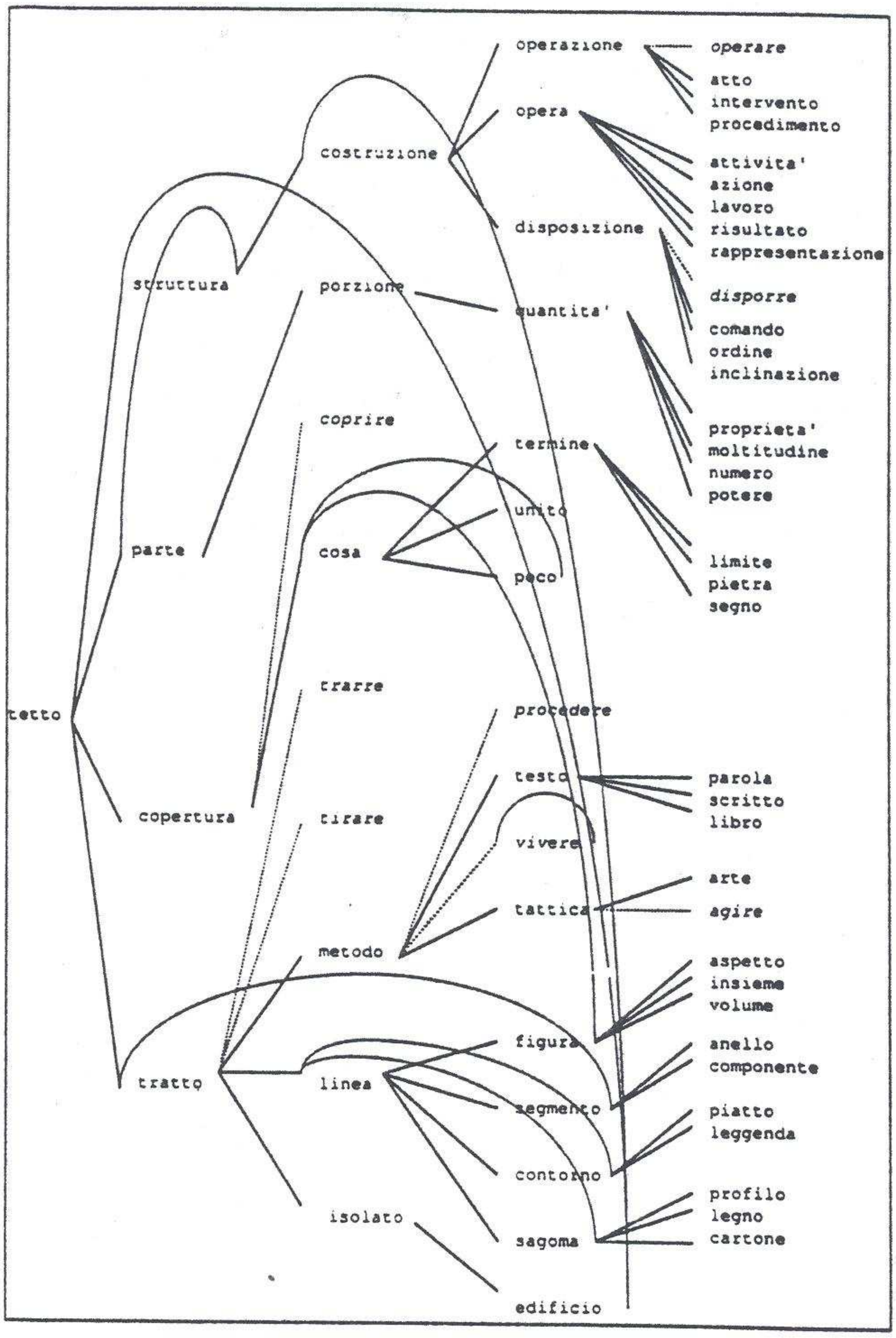


Figure 3: Output from the taxonomy display

References

Ageno, A., S. Cardoze, I. Castellon, M. A. Marti, G. Rigau, H. Rodriguez, M. Taule & M. F. Verdejo (1991): "An Environment for Management and Extraction of Taxonomies from On-line Dictionaries". ESPRIT BRA-3030 ACQUILEX WP No. 020.

Alshawi, H. (1989):"Analysing the Dictionary Definitions". In: B. Boguraev & E. J. Briscoe (Eds.): *Computational Lexicography for Natural Language Processing*. Longman, London.

Boguraev, B., E. J. Briscoe, N. Calzolari, A. Cater, W. Meijs & A. Zampolli(1988): "Acquisition of Lexical Knowledge for Natural Language Processing Systems, (ACQUILEX)." Technical Annex, ESPRIT Basic Research Action No.3030, Cambridge.

Briscoe, T., A. Copestake & V. de Paiva (Eds.) (1991): *Proceedings of the ACQUILEX Workshop on Default Inheritance in the Lexicon*. University of Cambridge Computer Laboratory, Technical ReportNo. 238, October 1991 (to be published by CUP, 1992). ESPRIT BRA-3030 ACQUILEX WP No. 040.

Calzolari, N. (1984): "Detecting Patterns in a Lexical Database". In: *Proceedings of the 10th International Conference on Computational Linguistics*. Stanford, California, pp. 170-173.

Calzolari, N. (1991): "Acquiring and Representing Semantic Information in a Lexical Knowledge Base". In: J. Pustejovsky (Ed.): *Proceedings of the Workshop on Lexical Semantics and Knowledge Representation.* Berkeley, California. ESPRIT BRA-3030 ACQUILEX WP No.016.

Copestake, A., A. Sanfilippo, T. Briscoe & V. de Paiva (1991): "The ACQUILEX LKB: An Introduction". In: T. Briscoe, A. Copestake & V. de Paiva (Eds.): *Default Inheritance in Unification Based Approaches to the Lexicon*. Esprit BRA ACQUILEX PROJECT (Action 3030), pp. 182-202.

Hagman, J. (1991): "Common and Odd Relations in Italian Dictionaries and their Treatment in Taxonomy Building". ESPRIT BRA-3030 ACQUILEX WP No. 044.

Hagman, J., (1992): "Semantic Parsing of Italian Dictionary Definitions". ESPRIT BRA-3030 ACQUILEX WP No. 047.

Jensen, K. (1988): "Issues in Parsing". In: A. Blaser (Ed.): *Proceedings of the Symposium on Natural Language at the Computer.* (pp. 65-83) Springer Verlag, Berlin/Heidelberg/New York.

Jensen, K. (1989): "A Broad-coverage Natural Language Analysis System". In: *Proceedings ofthe International Workshop on Parsing Technologies.* Carnegie Mellon University, 28-31 August 1989.

Jensen K., G. E. Heidorn, L. A. Miller & Y. Ravin (1984): "Parse fitting and prose fixing". *American Journal of Computational Linguistics*, 9 (3-4), 147-160.

Marinai, E., Peters, C. & E. Picchi (1990): *The Pisa Multi-Lexical Database System.* Pisa, November 1990, ACQUILEX Deliverable No. 4.

Marinai, E., C. Peters & E. Picchi (1990): *The Pisa Multilingual Lexical Database System.* Pisa, November 1990, ACQUILEX Deliverable No. 4.

Marinai E. & E. Picchi (1991): "A Procedure for Interactive Sense Disambiguation". ESPRIT BRA-3030 ACQUILEX WP N0.029.

Montemagni, S. (1991): "Tailoring a Broad Coverage Grammar for the Analysis of Dictionary Definitions". In: *Fifth Euralex International Conference..* Tampere, Finland. [Also in: Jensen, K., G. Heidorn & S. Richardson (Eds.) (forthcoming): *Natural Language Processing: The PLNLP Approach.* Kluwer Academic Press, Dordrecht.]

Montemagni, S., & L. Vanderwende (1991): "Structural Patterns versus String Patterns for Extracting Semantic Information from Dictionaries". In: *COLING 14, Nantes..* [Also in: Jensen, K., G. Heidorn & S. Richardson (Eds.)(forthcoming): *Natural Language Processing: The PLNLP Approach.* Kluwer Academic Press, Dordrecht.]

Oestling, A. (1992): "Parts and Wholes in Dictionary Definitions". ESPRIT BRA-3030 ACQUILEX WP No.046.

Spanu, A. (1992): "Extending the Type System from the Food-Subset to the Place-Subset". ESPRIT BRA-3030 ACQUILEX WP No. 051.

Spanu, A. (forthcoming): "Defining semantic relations on the basis of the analysis of the differentia". Pisa, ILC Technical Report.

Vossen, P. (1990): "The end of the chain: Where does decomposition of lexical knowledge lead us eventually?". ESPRIT BRA-3030 ACQUILEX WP NO.010. [Also in: *Proceedings of the 4th Conference of Functional Grammar*, June 1990, Copenhagen].