THE PISA MULTI-LEXICAL DATABASE SYSTEM

An integrated system for the acquisition, maintenance and interrogation of mono- and bilingual LDBs

Elisabetta Marinai, Carol Peters, Eugenio Picchi

ACQUILEX

ESPRIT BASIC RESEARCH ACTION No. 3030
Twelve Month Deliverable

Pisa, November 1990

ILC-ACQ-2-90

Introduction

The objective of the present report is to provide documentation for the 12 month release of the Multi-lexical Database (MLDB) system software, which is now being built at the "Istituto di Linguistica Computazionale", CNR, Pisa, within the context of the AQUILEX project. The documentation will consist of a description of the various stages of system design and development and includes a preliminary version of the MLDB Query System User Manual. The report thus constitutes the technical description for the Twelve Month Deliverable of the Lexical Database Software for the Pisa partners, demonstrated at the project review workshop in Cambridge, UK, 27-28 November 1990.

The scope of the system is to provide fast and flexible access to all the lexical information contained in the component machine-readable dictionaries: morphological, syntactic, semantic and conceptual. It should constitute a valid instrument for linguistic and lexicological research and in many natural language processing activities. The construction of the Pisa MLDB (undertaken in parallel with the construction of LDBs by the other project members for their own monolingual and bilingual source dictionaries) can thus be considered as a first step towards the final goal of the ACQUILEX project: "the development of a multilingual knowledge base containing the most general and domain-independent aspects of lexical knowledge represented in a fashion which makes it maximally reusable" (Boguraev et al., 1989).

The MLDB has been designed as an integrated modular system, i.e. it is made up of a number of basic modules which are combined to compose more complex components. At the same time, it has been conceived as incrementable, i.e. additional modules can be developed and added to the system to satisfy particular requirements as the need arises. The primary system is based on three main components: the set of parsing procedures which analyse the lexical data contained in the project machine-readable dictionaries (MRDs) and

process them onto lexical database (LDB) structures; the management system which controls the procedures for the storage, modifying and updating of the data; the data access and interrogation procedures. These modules have been generalized as far as possible to permit the loading and manipulation of different dictionaries in different machine-readable formats. In particular, the access and query system has been designed to be highly flexible and independent, with a user-friendly interface. A morphological analyser and generator for Italian can also be used in conjunction with the system. We hope to add a similar component for English soon.

The current lexical data of the system is derived from lexical material available at the Institute in machine-readable dictionary form: the Italian Machine Dictionary, the Garzanti "Nuovo Dizionario Italiano", and the Collins Concise Italian/English, English/Italian Dictionary. In the report, we shall refer to these dictionaries as the DMI, Garzanti and Collins, respectively. It is our intention to integrate other dictionaries into the system. In fact, the parsing procedures have been designed to be flexible and easily adaptable for the processing of different lexical structures onto a common database representation schema.

Research is now underway to enable the linking, comparison and merging of the information from the different component dictionaries of our MLDB to construct a derived lexicon. The aim is to provide not only a tool which makes it easier to compare and study lexical data derived from different sources but also to structure the data in such a way that further linguistic and lexical analyses at a higher level are facilitated. A new composite lexical entry will be created in which the data from each source can be examined, compared and verified, equivalent information can be merged and unified, and redundant information can be eliminated, while information which, for example, is only given in one of the sources becomes an integral part of the merged entry. A new structure of this type should not only be more complete than that represented by the single source LDBs but can be continuously enriched as new LDBs or other lexical data are added to the system.

The MLDB is implemented at the ILC on personal computers running the MS/DOS operating system and also on a Local Area Network of PCs with a host acting as server. By implementing the system under MS/DOS, we hope to ensure the maximum possible diffusion of the project results and the maximum possibility of experimentation. In fact, in recent years much experience has been accumulated at the Institute in the design and construction of lexical tools and prototypes running under MS/DOS systems. Many of these products have been distributed widely and are used in linguistic and literary research applications and studies throughout Italy and abroad. For the sake of our users, it was decided to maintain continuity in our design philosophy. However, great care has been taken to guarantee the transportability of the data and the Pisa lexical databases are also being loaded into the standardized LDB, which has been designed and developed at Cambridge running on Macintosh systems, so that the results gained from studies over the lexical data of all the project source LDBs can converge in the creation of the lexical knowledge base.

The first three sections of the report will describe the design and development of the system software: the parsing procedures used for conversion of the MRD sources; the database management system; the indexing, access and retrieval procedures. Section 4 will briefly present the work now underway to construct a derived lexicon by creating a tool to link and merge information from the separate LDBs. In the Appendix, a preliminary draft version of the MLDB Query System User Manual will be given. Figure 1 shows the structure of the integrated system.

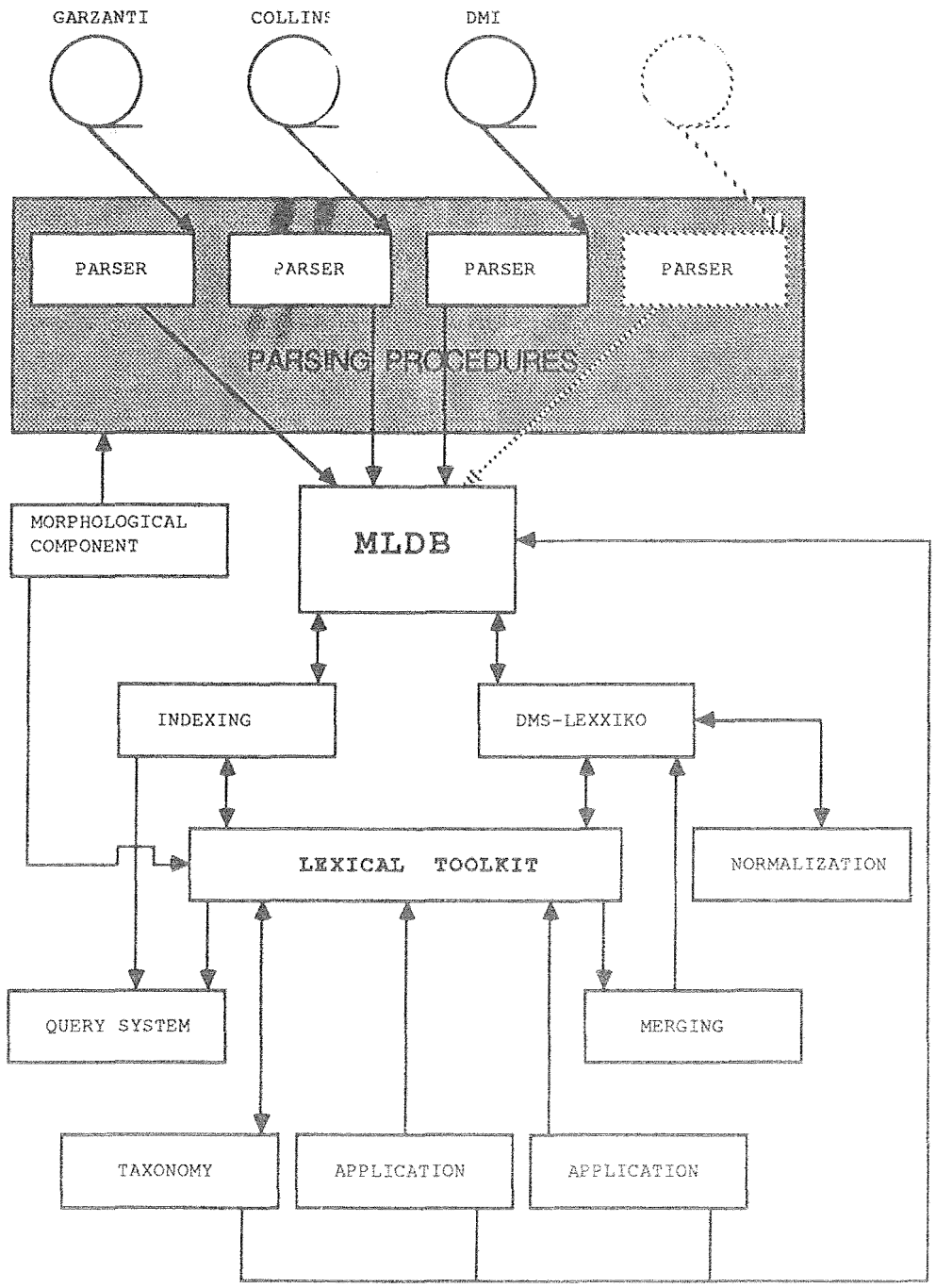


Figure 1. Schema of the Integrated Multi-Lexical Database System

1. The Parsing Procedures

The system parser consists of a group of procedures which have been designed to acquire and structure the source data from each machine-readable dictionary tape according to a set of "grammar-based" rules. The procedures work in two main stages: in the first, the information contained on the source tape is analysed, in successive steps, in order to identify the lexical information from the various other coded data present on the tape, and recognize and tag the separate data fields in each lexical entry; in the second stage, the data are further processed and analysed in order to map them onto the lexical entry templates which have been studied for the project. The lexical data in this format provides the input for the next set of procedures, known as LEXXIKO and described in Section 2: the Database Management System.

At the present, neither stage of the parsing process can be completely generalized to cover all dictionary tapes, as idiosyncratic solutions, related to the various ways information is coded on the tapes, must be used.

1.1 Acquisition of the Source Data

Machine-readable dictionaries can be in different formats according to the scope for which they were created. Our Garzanti and Collins dictionaries were provided by the publishers, on tapes prepared and coded for computer typesetting, and consisted of unsegmented text interposed with font-change codes and other printing instructions. Tapes of this type are much more difficult to treat and require more processing than publishers' tapes in which information on the nature and structure of the data is recorded not only implicitly via type-setting commands but also explicitly via codes intended for future electronic processing of specific information fields, e.g. the LDOCE tape. The Garzanti tape contained approximately 70,000 main entries. The printed version of Collins bilingual contains approximately 55,000 entries. The tape which we have been given actually contains about 20% of additional entries; these are mainly

specialist or scientific terms, proper nouns, and entries for derivatives. Certain entries on the tape also contain extra information which does not appear in the printed dictionary entry. The tape contains the two datasets: English-Italian and Italian-English.

The DMI instead had been built up at the ILC, with information extracted from a number of Italian dictionaries, and had already been analysed and organized on a simple structure; it consisted of three types of information: the lemmas - about 106,000 records, each containing a lemma plus associated phonetic, morpho-syntactic and semantic information and pointers to related lemmas and usage codes; the definitions - 180,000 definitions for the nouns, adjectives and verbs present in the file of lemmas, each definition classified according to its type; a morphological procedure which can analyse and generate all the inflected forms - approximately 1 million word-forms - for each lemma in the dictionary (see Gruppo di Pisa, 1979; Zampolli, 1983).

- *Parsing the computer-typesetting tapes*

The first step was the development of a set of procedures for the acquisition of the lexical data from the printing tapes, processing it so that it can then be structured in database form. In a publisher's type-setting tape, the organization of the data follows certain lexicographic and typographic principles; the underlying assumption being that the dictionary is a printed volume to be consulted by an intelligent human (Boguraev, 1986). Consequently, the tapes consist of a character stream containing a mixture of type-setting commands and lexical data. The lexical data must be extracted from the rest of the information: the font-change codes, typographical instructions, punctuation marks, etc., and the type-setting codes are essential keys to interpret its value. It is thus necessary to disambiguate the codes on the source tapes in order to segment the data into distinct entries and then separate and identify the different information fields within each entry. This was done mainly on the basis of recognizing the different commands used for changes of font, but for certain fields also according to their order of appearance in the entry.

```
§33§donna%106
§SG§s.f.
§11§%123kdo%121n-%123l %192s.f.%208
§99§%241
§11§essere umano di sesso femminile %192/ * di casa%208,
quella che attende alle faccende domestiche.
%095DIM.%208 %192donnina%208, %192donnino%208.
%095VEZZ.%208 e %095SPREG.%208 %192donnetta%208.
%095ACCR.%208 %192donnona%208, %192donnone%208. %09
§254§G.%208 %192donnicciola%208, %192donnaccia%208
§99§%242
§11§moglie, compagna, amante: %192la mia *%208
§99§%243
§11§titolo di rispetto anteposto al nome,
oggi di uso raro o region., come il maschile %192don%208
§99§%244
§11§persona di servizio %192/ * di cucina%208, cuoca, sguattera
§99§%245
§11§attrice: %192prima *%208
§99§%246
§11§una delle tre figure delle carte francesi,
detta anche %192regina%208; negli scacchi, il pezzo principale dopo il re.%074
```

Figure 2. Example of Typesetting Tape for "donna" for Garzanti

Clearly, in each case, the procedures must follow the logic which had been adopted by the publisher's when preparing the data for printing. This means that they must be adapted each time to cater for the special features of the particular dictionary being analysed. An example of how the data appear on the typesetting tape is given for the Garzanti dictionary in Figure 2.

The first step is thus a careful analysis of both the printed dictionary entry and the tape in input in order to identify the particular structure of the entry so that an initial classification of the information fields can be assumed. Each different information field is assigned an identification number. An example of the preliminary assumptions after a first analysis of the Collins bilingual dictionary tape on the basis of the font-changes is given below:

Assumed Information Field	*Description*
Headword	First bold type character string appearing in entry.
Homograph/Subentry	Bold type numerals.
Sense Division	A bold type letter between brackets.
Pronunciation	Roman type between square brackets.
Grammatical Category	First sequence of characters in italics without brackets, following a headword or subentry field.
Semantic Indicator	In italics between brackets.
Example of Usage	In bold type without brackets
Translations	In roman type. When this field is preceded by an Example of Usage field, it is an Example Translation; otherwise it is the Direct Translation of the headword.

Figure 3 shows the results obtained from a first run-through of the parsing procedure on the Collins tape for the entry for **accesso**. The equivalent results for Garzanti are shown in the first part of Figure 5.

accesso [at't esso] *sm* (a) access; **vietato l'~** no entry, no admittance; **di facile ~** *(luogo)* (easily) accessible; **avere ~ a** to have access to; **~ casuale** *(Inform)* random access. (b) *(di stizza, gelosia, tosse)* fit; *(di febbre)* attack, bout. (c) *(TV)*: **programmi dell' ~** educational programmes.

```
  *   **
  1    8accesso
  7   21[at<18>t<16><13>sso]
  6    3sm
  2    4(a)
  5    9access;
  4   15vietato l'=**=
  5   24no entry, no admittance
  4   15di facile =**=
  8    8(luogo)
  5   21(easily) accessible;
  4   13avere =**= a
  5   19to have access to;
  4   17ad =**= multiplo
  5   13multiaccess;
  4   13=**= casuale
  8    8(Inform)
  5   15random access;
  4   17=**= sequenziale
  8    8(Inform)
  5   11sequential
207    2o
  5   15serial access.
  2    4(b)
  8   28(di stizza, gelosia, tosse)
  5    5fit;
  8   12(di febbre)
  5   13attack, bout.
  2    4(c)
  8    4(TV)
  5    2:
  4   20programmi dell'=**=
  5   23educational programmes.


  *   Identification numbers for the different information
      fields. 1=Headword, 2=Homograph No. and Sense Divi-
      sion, 4=Example of Usage, 5=Translation,7=Pronuncia-
      tion, 8=Semantic Field, 207=Unidentified Information

  **  Numbers indicate length of information field (blank
      spaces are included in the count)
```

Figure 3. An example of first parsing of the entry for "accesso" from the Collins dictionary: Italian-English dataset
Printed dictionary entry and parsed dictionary tape entry

Once the preliminary separation of the information contained in the entry is completed, the results must be further refined. An examination of the tagged data was necessary in order to ensure that the first hypothesis was sufficient. For example, in the case of the Collins tape, from Figure 3 it can be seen that the homograph numbers and sense divisions were initially tagged with the same Id.No. Similarly, no distinction had been made between direct translations of the headword and translations of the examples. In a second run of the parsing procedure, these different fields were distinguished: a data field tagged by Id.No.2 is recognized as a sense division if the data appeared between brackets, otherwise as a Homograph. Similarly, the example translations (a data field (Id.No.5) immediately preceded by an example field (Id.No.4)) were distinguished from the other fields tagged by Id.No.5 which were direct translations of the headword. Furthermore, it was noted that the Semantic Indicator was not the only information which appears in the dictionary entry between brackets. A number of other fairly easily identifiable types of information are also given in this way, e.g. cross-references such as *(see adj)*, *(vedi ag)*, and some grammatical information. In order to separate these very different types of information, a table was created which listed all the possible data which could appear in italics between brackets but which was not to be classified as semantic information. The procedure thus refers to this table in order to classify these information types (e.g. *see ag*, *vedi ag* are tagged as cross-references (X-ref)).

Again referring to the Collins data, another example of the type of interventions necessary at this stage was the treatment of the *or* and *o* links which are frequently found in the Translation or Examples of Usage fields, e.g. under **admit** we have "I must admit that ..." translated by "Devo ammettere *or* confessare che ..." This *or* was recognized by our initial parsing procedure as being a separate information field for which no information field had been assumed and given its own Id.No. This meant both examples and translations which contained *o/or* were divided. This can be seen in Figure 3 where "sequential *o* serial access", the translation of the example "accesso sequenziale", has been divided by the procedure into three separate fields. Therefore, in a second run of the tape parsing

procedure, it was necessary to recompose this field. We united the data here at this stage so that any subsequent query searching a translation equivalent for "accesso sequenziale" will retrieve "sequential *o* serial access". This will cause no problem to the human user of our database who will easily recognize that "accesso sequenziale" can be translated either by "sequential access" or by "serial access". This will not be true, however, for any application program. It will thus be necessary to find a way to disambiguate such information in a later stage so that, for example, "sequential access" can be retrieved independently of "serial access". The same sort of situation arises with the use of the "/". For example, under **brillare** we find "brilla per la sua bellezza/intelligenza - she is outstandingly beautiful/intelligent". We actually have two examples here, each with its own translation. We can easily find even more involved cases; under **abitudine** we find both "*o*" and "/" in the example "Buona *o* bella/cattiva *o* brutta abitudine" translated by good/bad habit. Here the lexicographer has managed to compress in one complex example and translation 4 source language examples and 2 possible translations. In all such cases, the human reader of the dictionary or user of the database will have no problem in disambiguating immediately the different possibilities offered him, whereas the machine will need very careful instructions in order make any attempt to disentangle the different pairs of Examples and Translations. The same sort of situations, although not quite so complex as we are dealing with a monolingual dictionary, are found in Garzanti.

Certainly, we cannot expect to be able to reconstruct all such information automatically in this first parsing stage. If it should be decided, later on, to intervene in such cases, we will probably devise a semi-automatic, interactive procedure in which simple, unambiguous situations presenting the *o* or the "/" in Examples and Example Translation fields will be resolved automatically, with a separate reconstruction of all possible pairs of Examples and their Translations, whereas the more complex cases will be signalled together with a numbered list indicating the most probable solutions.

Other problems which arise during this stage are caused by unexpected font-changes and inconsistencies in the data. For example, very trivial "errors" in the printed data, e.g. a comma in italics instead of roman font or vice versa, which are certainly irrelevant to the reader of the dictionary obviously cause problems in a procedure which analyses the typesetting codes in order to recognize the different data fields. We also, although much more rarely, find more serious problems in the data 'such as a whole piece of information in the wrong type, or the omission of an essential data field. These cases are marked not only for correction by us but so that they can also be eventually signalled back to the dictionary editors. All this means that, after the initial parsing procedure is run, the results have to be checked and cleaned up. When we find recurring errors of a particular type, it may be possible to correct them automatically; in other cases manual intervention is obligatory. Even so the dictionary tape parsing procedures are very productive; certainly there is no comparison with the costs and time which would have been needed to input all the data contained in the dictionary from scratch.

- *Parsing the DMI tapes*

The first parsing of the DMI was much less complex than that for the Garzanti and Collins type-setting tapes. The DMI had been conceived as an electronic dictionary and consequently the data was already structured and coded according to its particular value, i.e. according to its function in the lexical entry. The first step was to identify the codes which had been used to tag the lexical data and then to merge the different types of information into one tagged file, ready for mapping onto the DB representation schema. All the lexical information contained in the DMI is identified by numerical keys and linked by pointers; these keys are used to insert the data in the template. The DMI has already been used at the ILC in many research applications; it was possible to insert some of the results from these studies into the tagged file ready for inclusion in the database representation structure, e.g. the results of a first study on the taxonomic relationships and of a study on derivation.

1.2 Structuring the Tagged Data onto a Database Representation Schema

The next step was to construct suitably coded structures to represent the lexical entries from each source dictionary in LDB format. The aim is to represent precisely all the information contained in the printed entry - and, wherever possible, to represent explicitly information which is only given implicitly. In this way, the retrieval of "new" information or information which is normally difficult to access can be facilitated. The representation structures or schemas had to be comprehensive, i.e. capable of covering all the information which can be found in the entries of the dictionary for which they were designed. However, in view of the intention to implement a totally integrated system in which merging and mapping between the monolingual and the monolingual and bilingual systems is possible, the representation schemas for the different source dictionaries had to be compatible. The tagged dictionary data resulting from the first parsing stage was thus structured onto the templates which have been studied for the project dictionaries and which are described in detail in Section 1.2 of the "Computational Model of the Dictionary Entry" (Calzolari et al, 1990). This report describes the maximal entries for Collins and Garzanti, i.e. all the possible fields represented in all possible positions. The template makes the hierarchical structure of the dictionary entry evident by using Node tags which group semantically and logically connected constituents. The use of a standardized representation language ensures compatibility between the entries from different source dictionaries. This format is thus already a useful vehicle for the exchange of dictionary data between different projects. It also represents the intermediate stage in the process of mapping the data from all the different project MRDs onto the Common Lexical Entry, designed to facilitate the organization of a common ACQUILEX lexical database in which not only the exchange of lexical data but also data processing activities in collaboration are possible.

The database representation schema which we have used at this stage can be considered as a subset of these templates, as so far not all of the implicit information has been derived and some of the

fields have not yet been analysed. The reason for this is that database development has necessarily proceeded simultaneously with the study of the database representation structure; it was considered essential to study and and construct all the system components as quickly as possible in order to be able to provide not only the tools but also the data in a suitable format for subsequent studies and analysis, e.g. the procedures to disambiguate the definitions, to construct the taxonomies, etc. Thus, while much attention has been given to the analysis of certain Nodes which contain the information which is of interest to these procedures (see, in particular, the work to analyse the Sense Label Nodes described below), other types of information will be examined in detail at a later stage. Clearly, at the end of the project, it is our intention to have represented all the relevant lexical information, explicit and implicit, which can be extracted from our dictionaries on the project templates.

Once we had defined our database representation structures, procedures were developed to automatically map the parsed dictionary tape data onto these structures. This posed no real problems for the DMI. As has already been stated, the DMI tapes contained coded and structured information. It was a simple matter to map this information directly to the lexical template. Figure 4 shows an entry for the DMI structured on the database representation schema.

The process was not so simple when dealing with the Collins and Garzanti publisher's tapes. The main problem which had to be tackled is that when mapping the parsed and tagged tape data onto the DB representation structures certain information which only appears implicitly in the entry now has to be analysed in order to be represented explicitly. Each tagged field resulting from the first stage had to be carefully analysed in order to extract the information required for the representation schema. Therefore, the procedures which were written for the Garzanti and Collins data were far more complex than those for the DMI. Figure 5 shows the results of the two parsing stages for the entry for donna (woman) from the Garzanti dictionary

```
DONNA
000     LemmaNum 32201000
001        Lemma DONNA
003          PoS SF
000       DefNum 57103000
007          Def FEMMINA FISICAMENTE ADULTA DELLA SPECIE UMANA
000       DefTyp 3
000       RSem_I FEMMINA
000       DefNum 57104000
007          Def PERSONA INDETERMINATA DI SESSO FEMMINILE
000       DefTyp 3
000       RSem_I PERSONA
000       DefNum 57105000
007          Def ESSERE UMANO FEMMINILE SECONDO LE QUALITA' MORALI
000       DefTyp 3
000       RSem_I ESSERE
000       DefNum 57106000
007          Def ESSERE UMANO FEMMINILE IN RELAZIONE CONTESTO SOCIALE
000       DefTyp 3
000       RSem_I ESSERE
000       DefNum 57107000
007          Def DONNA DI UNA STESSA FAMIGLIA
000       DefTyp 3
000       RSem_I DONNA
000       DefNum 57108000
007          Def DONNA DELLA STESSA CITTA' CUI APPARTIENE CHI PARLA
000       DefTyp 3
000       RSem_I DONNA
000       DefNum 57109000
007          Def DONNA DI SERVIZIO
000       DefTyp 3
000   SemProcCod L
000       RSem_I DONNA
000       DefNum 57110000
007          Def SIGNORA,PADRONA
000       DefTyp 2
000    DefUseCod 5
010        Synon SIGNORA
010        Synon PADRONA
000       DefNum 57111000
007          Def TITOLO RISERVATO ANTICAMENTE ALLE RELIGIOSE DI CERTI ORDINI
000       DefTyp 3
000       RSem_I TITOLO
000       DefNum 57112000
007          Def TITOLO RISERVATO ALLE NOBILDONNE E ALLE SIGNORE DI RIGUARDO
000       DefTyp 3
000       RSem_I TITOLO
000       DefNum 57113000
007          Def MARIA VERGINE
000       DefTyp 7
000    DefUseCod 1
000       RSem_I MARIA
000       DefNum 57114000
007          Def ATTRICE
000       DefTyp 2
010        Synon ATTRICE
000       DefNum 57115000
007          Def REGINA,DAMA/ FIGURA DELLE CARTE DA GIOCO
000       DefTyp 5
000       RSem_I FIGURA
010        Synon REGINA
010        Synon DAMA
```

Figure 4. An example of mapping onto the DB Representation Structure for "donna" for the DMI

Dizionario Garzanti Primo Livello DONNA

DONNA
Entry donna
CatGr s.f.
11 [dòn-] {s.f.}
99 1
11 essere umano di sesso femm.r e {/ ■ di casa}, quella che attende alle faccende domestiche. «DIM.» {donnina}, {donnino}. «VEZZ.» e «SPREG.» {donnetta}. «ACCR.» {d.. na}, {donnone}. «SPRE
Cont G.» {donnicciola}, {donna ..}
99 2
11 moglie, compagna, amante: {la mia ■}
99 3
11 titolo di rispetto antepost al nome, oggi di uso raro o region., come il maschile {don}
99 4
11 persona di servizio {/ ■ di cucina}, cuoca, sguattera
99 5
11 attrice: {prima ■}
99 6
11 una delle tre figure delle .rte francesi, detta anche {regina}; negli scacchi, il pezzo principa. d.po il re.

Dizionario Garzanti Secondo Livello DONNA

DONNA
Entry donna
PoS s.f.
Pron dòn-
Sense 1
Def essere umano di sesso femminile
Idiom ■ di casa
Expl quella che attende alle faccende domestiche.
Alter DIM. donnina
Alter DIM. donnino
Alter VEZZ. e SPREG. donnetta
Alter ACCR. donnona
Alter ACCR. donnone
Alter SPREG. donnicciola
Alter SPREG. donnaccia
Sense 2
Def moglie, compagna, amante
Examp la mia ■
Sense 3
Def titolo di rispetto anteposto al nome, oggi di uso raro o region., come il maschile {don}
Sense 4
Def persona di servizio
Idiom ■ di cucina
Expl cuoca, sguattera
Sense 5
Def attrice
Examp prima ■
Sense 6
Def una delle tre figure delle carte francesi, detta anche {regina} negli scacchi, il pezzo principale dopo il re.

Figure 5. Example of Parsing and Mapping onto DB Representation Structure for "donna" from the Garzanti dictionary

Figure 5 shows clearly the differences in analysis between the first and second levels of parsing. In the following, we shall attempt to give an idea of the kind of problems to be faced and the interventions necessary when mapping the tagged data onto the db representation structures for Collins and Garzanti.

- *Reconstructing Information*

For an idea of one type of problem to be addressed, i.e. reconstructing information which has been given only implicitly, let us look at the entry in Collins for the Italian adjective **candido.** The printed dictionary entry is as follows:

> **candido,a**: ['kandido] *ag*:**(a)** *(bianco):*, (pure) white; ** **come la neve** as white as snow. **(b)** *(fig: ingenuo):* ingenuous, naive; : *(:sincero):* candid, frank; *(:innocente):* pure, innocent.

The first sense division does not pose any real problems. However, sense (b) needs some interpretation if all the information given in the semantic glosses is to be captured. The user must understand it as follows: **candido** used in the figurative sense as a near synonym for "ingenuo" is translated as "ingenuous, naive"; **candido** used in the figurative sense as a near synonym for "sincero" is translated as "candid, frank"; **candido** used in the figurative sense as a near synonym for "innocente" is translated as "pure, innocent".

Thus, the user must not only recognize what is implied by "fig" and by the use of "ingenuo, sincero, innocente" but he must also understand that the use of the colon preceded directly by a parenthesis as with *(:sincero)* and *(:innocente)*: is a metalinguistic symbol and means that the indication (in this case "fig") which appears in front of the first sense label, *(fig:ingenuo)* is to be understood as repeated also for the other two cases.

Even if the human user is easily able to recognize the value of all this information, we were clearly obliged to disambiguate and tag it explicitly for storage in the database. The procedure we developed to tag the semantic labels, near synonyms and other contextual

information for Collins is described below. To deal with the information omitted and suggested by the use of the colon, we had first to reconstruct for each sense label field the total information given. Thus instructions were included in the procedure which stated that when the sequence"*(:*" was encountered then the program should backtrack to find the first preceding sequence *(xxx:* where *xxx:* stands for a character string of any length representing the information omitted. The missing information must be reconstructed before the entry can be mapped correctly onto the template

Figure 6 shows the output of the first parsing for **candido, a** followed by the results of the mapping procedure in which the omitted information has been reconstructed.

- *Variant, Cross-reference and Run-on Group Disambiguation*

The examples given here refer to the interventions necessary for the Garzanti dictionary to treat this kind of information. After the first parsing stage, we had a field tagged by Id.No.13; this field contained the only information appearing in bold type in the entry after the Headword: variant, X-reference and run-on information. In order to map the data onto the db representation schema, we had thus to distinguish between these three different types of data:

Variant forms

The following conditions determined the presence of a variant form:
1. If the content of the field which has been tagged by the first stage of the parsing procedures as Id.No. 33 (the Headword field) contains a comma (,), e.g **defogliante, defoliante**; then the string preceding the "," is taken as the Headword and that following is tagged as the variant form;

```
1   10candido,a
7   10['kandido]
6    3ag
2    4(a)
8    9(bianco)
5   14(pure) white;
4   15~ come la neve
5   19(as) white as snow.
2    4(b)
8   14(fig:ingenuo)
5   18ingenuous, naive;
8   11(:sincero)
5   14candid, frank
8   13(:innocente)
5   15pure, innocent.
```

Results of Dictionary Tape Parsing Procedure

```
ENTRY = candido,a
   L1 = Italiano
   Hdwd_form = candido,a
   Pronunciation = ['kandido]

      POS = ag

         Sense_no.= (a)
              SI_type = syn
              SI_text = bianco
              Trans = (pure) white
              Example = ~ come la neve
              Ex_Trans = (as) white as snow

         Sense_no = (b)
              Semantic_Code = fig
              SI_type = syn
              SI_text = ingenuo
              Trans = ingenuous, naive
              Semantic_Code = fig
              SI_type = syn
              SI_text = sincero
              Trans = candid, frank
              Semantic_Code = fig
              SI_type = syn
              SI_text = innocente
              Trans = pure, innocent
```

Results of Template Mapping Procedure

Figure 6. Example of Parsing Mapping onto DB Representation Structure for "candido" from Collins Dictionary

2. If a field which has been tagged as Id.No. 11 by the first stage of the parsing procedures (field normally containing definitions, examples, idioms, proverbs, etc.) contains: (i) only a comma and a blank and there is a following field no.13, then the content of the field no.13 will be a variant form; (ii) a } followed by a comma as the last or penultimate character in the string, then the content of the following field no.13 will be a variant form; (iii) a] followed by a comma as the last or penultimate character in the string, then the content of the following field no.13 will be a variant form, e.g. **nocchiere** [.....] , **nocchiero**;

Cross-references

In Garzanti we found three types of Cross-references: (i) when the headword is an abbreviated form or an acronym, in the POS field we find the string "abbr" or "abbr.", and the content of field no. 13 will be the X-referenced item, i.e. the extended form; (ii) when the headword is a comparative or superlative adjective, in the POS field we find the string "agg.comparativ." or "agg.superl." and the content of field no. 13 will be the X-referenced item, i.e. the adjective in its base form; (iii) when in addition to the presence of field code no. 13, we also have a field code no. 34, then the content of field no. 13 will be a X-referenced item, e.g. **eguale** *agg.* → **uguale**, where → is the content of field no.34.

Run-ons

If the entry contains a field no. 13 and it is neither a variant form nor a cross-reference, then it will be a run-on. The content of the field will be string which must be attached to the headword using ad-hoc rules to give the run-on form. Garzanti gives only two types of values for the run-on field: either the suffix **-mente** for adverbs which can be formed from adjectives, or a suffix which transforms transitive and intransitive verbs into pronominal or reflexive forms.

- *Sense Label Disambiguation*

Our examples for this section will be taken from Collins which is much richer than Garzanti in this type of information.

In Collins the scope of the Sense Label fields is to give the user information to help him to select an appropriate translation: subject codes, register and usage codes, general contextual indicators and restrictional collocations, semantically related words such as synonyms or near synonyms, superordinates, etc. In order to use these labels correctly, the dictionary user must be able to recognize intuitively, case-by-case, exactly what information he is being given. We had to identify this information explicitly so that it can be accessed directly and exploited in various ways. Therefore, a sense label parsing procedure was designed to disambiguate this field automatically, as far as possible.

There are no problems involved in tagging the semantic labels; the procedure searches them in previously created lists which we derived in the first place from the information given in the User's Guide to the dictionary and then by scanning our first results to capture labels which had been omitted the first time round. All the semantic labels have been tagged for both dictionaries and are thus ready for direct access and retrieval.

In Collins, in order to disambiguate between the different types of Semantic Indicators, i.e. contextual indicators or collocates and the near synonyms, for the Italian entries, we have introduced a procedure using the Italian Machine Dictionary - DMI - to recognize the near synonyms; if the word given as a semantic indicator has the same part-of-speech as the headword, then it is tagged as a possible synonym, otherwise it is a contextual indicator. An example of the results obtained from this procedure is given in Figure 7 for the Italian verb **abbandonare** where the printed dictionary entry can be compared with the explicitly tagged LDB entry. Of course problems arise when we have homography; in this case manual intervention is necessary to achieve 100% success rate. We have not yet perfected the procedures which we are using to disambiguate this sort of information, especially for the English entries.

abbandonare [abbando'nare] 1 *vt* (a) *(gen)* to abandon; *(famiglia, paese)* to abandon, desert; ~ qn a se stesso to leave sb to his *(o* her) own devices; il coraggio lo abbandonò his courage deserted him; ~ la nave to abandon ship; ~ il campo *(Mil)* to retreat. (b) *(lasciare indietro)* to leave behind, abandon. (c) *(trascurare: casa, lavoro)* to neglect. (d) *(rinunciare a)* to give up; *(: studi, progetto)* to abandon, give up; ~ la speranza to give up hope. (e) *(lasciare andare: redini)* to loosen; ~ la presa to let go; abbandonò la testa sul cuscino he let his head fall back on the cushion.

```
ENTRY = abbandonare
  L1 = Italiano
   Hdwd_form = abbandonare
   Pronunciation = [abbando'nare]

     Hom_No. = 1
       POS = v
         subcat = t

          Sense_no. = (a)
                Usage_code = gen
                Trans = to abandon
                SI_type = context
                SI_text = famiglia, paese
                Trans = to abandon, desert
                Example = ---- qn a se stesso
                Ex_Trans = to leave sb to his (o her) own devices;
                Example = il coraggio lo abbandono'
                Ex_Trans = his courage deserted him;
                Example = ---- la nave
                Ex_Trans = to abandon ship;
                Example = ---- il campo
                Subject_Code = Mil
                Ex_Trans = to retreat.

          Sense_no. = (b)
                SI_type = syn
                SI_text = lasciare indietro
                Trans = to leave behind, abandon.

          Sense_no. = (c)
                SI_type = syn
                SI_text = trascurare
                SI_type = context
                SI_text = casa, lavoro
                Trans = to neglect

          Sense_no. = (d)
                SI_type = syn
                SI text = rinunciare a
                Trans = to give up
                SI_type = syn
                SI_text = rinunciare a
                SI_type = context
                SI_text = studi, progetto
                Trans = to abandon, give up;
                Example = ---- la speranza
                Ex_Trans = to give up hope.

          Sense_no. = (e)
                SI_type = syn
                SI_text = lasciare andare
                SI_type = context
                SI _text = redini
                Trans = to loosen
                Example = ---- la presa
                Ex_Trans = to let go
                Example = abbandono' la testa sul cuscino
                Ex_Trans = he let his head fall back on the cushion.
```

Note that 4 possible synonyms have been identified for abbandonare:
lasciare indietro, trascurare, rinunciare a, lasciare andare.

Figure 7. Disambiguation of Semantic Indicators for an Italian verb in Collins bilingual dictionary

An example of how we eventually hope to use this information is to make it possible to call up a list of all items which have been automatically tagged by our parsing procedure as being possible near synonyms or words related to a given searched item, together with their translations. In this way, a search could be expanded from a single lexical item to a set of associated items together with their translations and thus concept clusters could be created. For instance, a search could begin with the concept represented by the Italian lexical entry **abilità** in the sense characterized in Collins by the semantic indicators **accortezza** and **astuzia** - words which our procedure would tag as being near synonyms of **abilità**. Following the path which will be traced through the dictionary by recognizing the items tagged in their turn as near synonyms of **accortezza** and **astuzia** and so on, lists are produced which include for Italian **abilità, accortezza, astuzia, avvedutezza and scaltrezza** translated in English by **cleverness, shrewdness, good sense, astuteness, prudence, slyness** and **cunning**. The English list can then be used to expand the Italian list. If the procedure is run interactively, at each step, the user can decide whether a processed word is relevant, including it in the list or rejecting it. If the procedure is run automatically instead of interactively, noise will be created but irrelevant words can easily be disregarded in a successive step.

In this way, the potential of the dictionary as a source of lexical knowledge can be exploited, e.g. in this case a particular semantic relationship between words, which the dictionary had not made explicitly evident, i.e. that of synonymy, has been identified, and knowledge which at a first glance is not present, or not accessible, can be identified for future retrieval. This type of information is important for the procedures which are being designed to map and merge information between the monolinguals and monolingual and bilingual datasets (see Section 4).

It can be seen from the figures that, so far, we have not resolved the problems involved in printing those symbols of the IPA phonemic set which are not included in the character sets of our printing fonts. At the moment, they have been coded, i.e. each

symbol which could not be printed has been represented by a numeric code, so that they can be recognized and searched if necessary. In the future, if specific researches mean that there is a need to display this information clearly, we intend to use a graphic display set which will make it possible to design the missing symbols and add them to the set of available characters.

The lexical data structured and tagged on the Database Representation Structures is the input for the next component of the MLDB: the Database Management System.

2. Database Management System

This system module, known as LEXXIKO, manages the data storage, modification and updating procedures and provides simple dictionary access functions. It accepts as input the lexical data structured on database representation schemas as described in the previous section. The system works dynamically, permitting the storage, update and retrieval of the lexical data using easy-to-learn access and editing techniques. For each dictionary included in the system, the storage and maintenance procedures are based on two main files: 1) the lexical data file, containing the structured and tagged entries; 2) the index file. At this stage, the access key to the data is the dictionary headword, optionally, other indices can be created on internal fields. The dictionary can be scanned in alphabetical order, or the entries can be accessed directly by entering the headword at the keyboard. Operating on the files structured in LEXXIKO-mode and using the ad-hoc designed access functions, it is possible to intervene on the lexical material in various ways, e.g. for the automatic correction of certain recurring incongruities in the data; the normalization of the POS fields of the different dictionaries and the insertion of a new field containing the standardized POS in each entry; the creation of MISCLEXX which merges the separate source dictionaries into a single structure (see section 4); the extraction of certain fields or elements from the entries to perform particular studies on the data; the insertion of new fields generated by particular analyses on the data, e.g. taxonomy data, etc..

The module consists of the following component procedures:

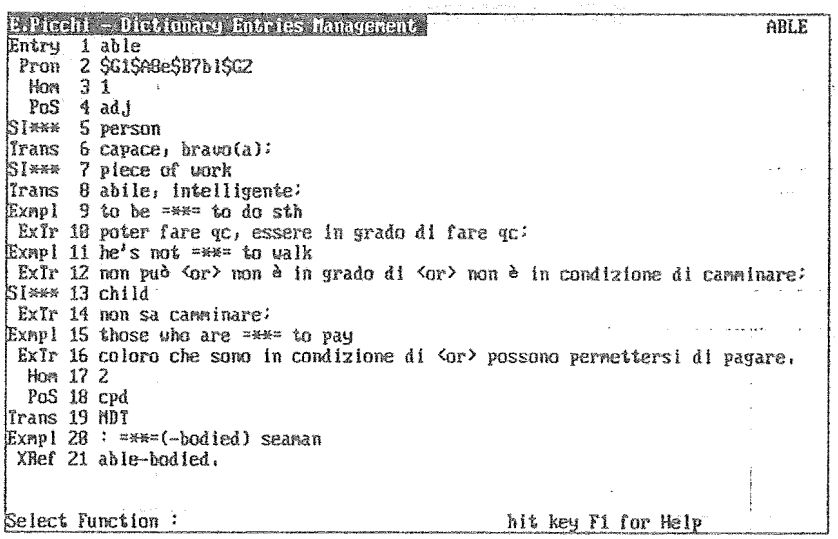
LEXXREST: this procedure is used to acquire the lexical data structured on the representation schemas as described in the previous section and to compact them on a variable length record in LEXXIKO structure; primary indices are created on the headword field, optionally, other indices can be created on internal fields.

LEXXIKO: this is the core procedure of the whole system. It permits the editing of the lexical entries, which are called using the access

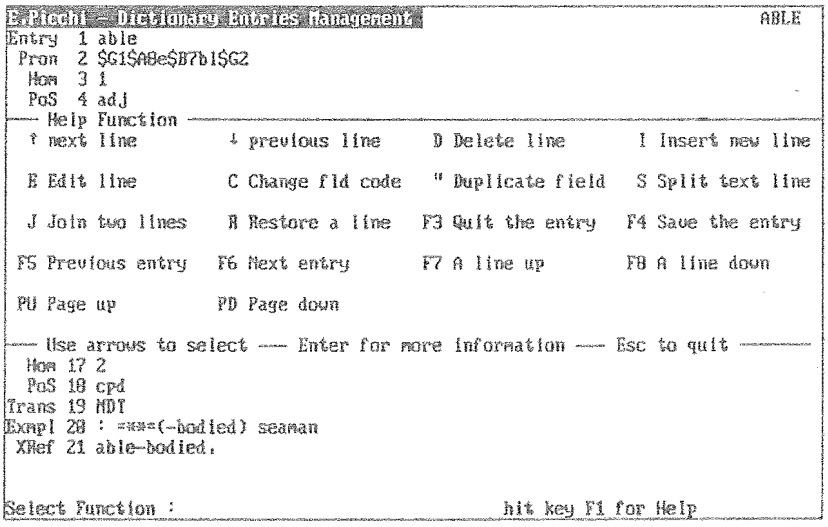
key. Existing entries can be modified or corrected, new entries can be inserted. A series of functions are available to operate on the data in the entry. A Help function can be called as needed. The following functions have been implemented:

Insert: inserts a new field; the user is prompted to indicate the field-code from a list of all the possible Field-codes and then to enter the content of the field.
Edit: the content of a chosen field can be modified.
Delete: eliminates a field (Field-code + content).
Change: changes a Field-code, leaving the field content unaltered.
Split: the content of a field can be divided into two; each part will have the same Field-code.
Join: two successive fields can be united; the Field-code of the first will be maintained.
Restore: any changed or deleted field can be restored and will be inserted immediately after the location indicated by the cursor.
Next: the following entry becomes the current entry.
Up: the previous entry become the current entry.
Eliminate: an entire entry is eliminated from the archives.
Copy Entry: a copy of the current entry is created and can be inserted within another entry.
Restore Entry: used to insert an entry copied or eliminated, using the **Copy Entry** or the **Eliminate** functions above, into another entry.

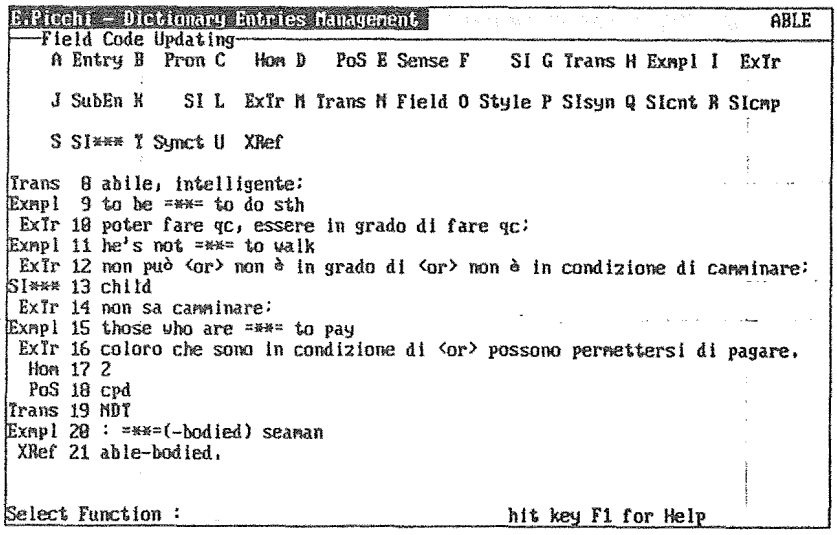
An example of a session at the terminal, in which the Collins bilingual entry for the English lemma **able** is modified, is shown in the next three pages.



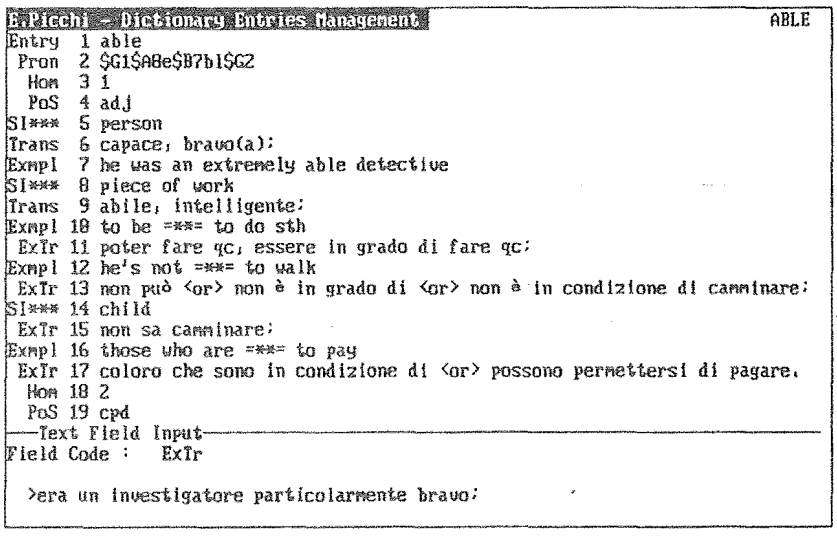
The user calls the dictionary entry which he wants to modify by entering its headword **able** on the keyboard. The structured and tagged entry for **able** in the LEXXICO format is displayed on the screen.



The user calls up the **HELP** function to display the different functions he can use to intervene on the entry.



The user wants to add an Example and Example Translation to the Entry to illustrate the use of able to describe a person. He must position the cursor at the point in which he wishes to insert new data. In this case, immediately after Translation field No.6. By entering I (Insert New Line) at the keyboard, he is given a display of the Data Fields which can be selected. He enters H to indicate that he wants to insert an Example data field.



The Field Code for the selected Data Field is displayed and the user enters his new example: "he was an extremely able detective" at the keyboard.

```
E:Picchi - Dictionary Entries Management                    ABLE
Entry  1 able
 Pron  2 $G1$A0e$B7b1$G2
  Hom  3 1
  PoS  4 adj
SI***  5 person
Trans  6 capace, bravo(a);
SI***  7 piece of work
Trans  8 abile, intelligente;
Exmpl  9 to be =**= to do sth
 ExTr 10 poter fare qc, essere in grado di fare qc;
Exmpl 11 he's not =**= to walk
 ExTr 12 non può <or> non è in grado di <or> non è in condizione di camminare;
SI*** 13 child
 ExTr 14 non sa camminare;
Exmpl 15 those who are =**= to pay
 ExTr 16 coloro che sono in condizione di <or> possono permettersi di pagare.
  Hom 17 2
  PoS 18 cpd
Trans 19 NDT
---Text Field Input-----------------------------------------------
Field Code :  Exmpl

 >he was an extremely able detective
```

The user must now insert the translation for the new example. He has again entered I (for Insert New Line) at the keyboard, and has selected the "ExTr" Field Code. He now enters his translation: "era un investigatore particolarmente bravo" at the keyboard.

```
E:Picchi - Dictionary Entries Management                    ABLE
Entry  1 able
 Pron  2 $G1$A0e$B7b1$G2
  Hom  3 1
  PoS  4 adj
SI***  5 person
Trans  6 capace, bravo(a);
Exmpl  7 he was an extremely able detective
 ExTr  8 era un investigatore particolarmente bravo;
SI***  9 piece of work
Trans 10 abile, intelligente;
Exmpl 11 to be =**= to do sth
 ExTr 12 poter fare qc, essere in grado di fare qc;
Exmpl 13 he's not =**= to walk
 ExTr 14 non può <or> non è in grado di <or> non è in condizione di camminare;
SI*** 15 child
 ExTr 16 non sa camminare;
Exmpl 17 those who are =**= to pay
 ExTr 18 coloro che sono in condizione di <or> possono permettersi di pagare.
  Hom 19 2
  PoS 20 cpd
Trans 21 NDT
Exmpl 22 : =**=(-bodied) seaman

Select Function :                              hit key F1 for Help
```

The new entry for able is displayed. It can be seen that in Data Fields Nos. 7 and 8 we have a new Example and Example Translation.

LEXXIDUMP: this function is used to extract from the LEXXIKO file part of the dictionary in the same format as that generated by the MRD parsing procedures. This subset of the dictionary can then be inserted in a new file using LEXXREST described above.

LEXXIKST: used to print entries of the chosen dictionary.

Operating on the files structured in LEXXIKO-mode and using the access functions described above, it is possible to intervene on the lexical material in various ways, e.g. for the automatic correction of certain recurring incongruities in the data; the normalization of the POS fields of the different dictionaries and the insertion of a new field containing the standardized POS in each entry; the creation of MISCLEXX which merges the separate source dictionaries into a single structure; the extraction of certain fields or elements from the entries to perform particular studies on the data; the insertion of new fields generated by particular analyses on the data, e.g. taxonomy data, etc..

The data files managed by LEXXIKO can now be mapped onto the Common Lexical Template

The Common Lexical Entry Template

The templates described for the parsing procedures of Section 1 simply represent the contents of the lexical entries for each dictionary, while ensuring a certain compatibility between the different templates by the use of a common representation language. The next stage is to map the single, idiosyncratic dictionary representations onto the project Common Lexical Entry (CLE) illustrated in Section 2.2 of the "Computational Model of the Dictionary Entry" six-month project deliverable mentioned above. This template has been designed to facilitate the organization of a common project lexical database in which not only the exchange of lexical data but also data processing activities in collaboration are possible. The procedures which maps the data from the standardized templates onto the CLE are quite complex as not only is the structure

of the entry different but information which still remained un-analysed at the end of the first parsing stage must now be interpreted. The entries in the CLE are structured on a "one-entry one-major-part-of-speech" basis whereas in the source dictionaries, homographs representing more than one POS are frequently grouped together; in the same way, the source dictionaries entries often contain additional information such as variant entries, derivatives, phrasal verbs. All these are represented on the CLE as separate entries. It is thus essential that on the CLE relations between lexical items which were given implicitly in the source dictionaries are represented explicitly This means that the source entries must be subjected to considerable processing in order to interpret their contents for a correct mapping to the CLE. For this reason, owing to lack of time, a complete mapping of our source dictionaries onto the CLE has not yet been possible.

The files structured in LEXXIKO-mode constitute the input for the next component of the system: the indexing access and retrieval procedures for the MLDB query system.

3. Indexing and Access and Retrieval Procedures

The design of the MLDB access and interrogation procedures has followed the philosophy already adopted at the ILC in the development of other systems, in particular, the DBT, a full-text retrieval system studied with the specific requirements of literary and linguistic research in mind (Picchi, 1983; 1990). The query language which has been implemented is thus deliberately similar to that used in the DBT in order to facilitate interrogation of the MLDB by our users.

The MLDB query system must permit fast and flexible access to the dictionary data via any one of the information categories present in the entry and which have now been identified explicitly and organized on the project computational model by our dictionary parsing and mapping procedures. The indexing procedure takes as input the dictionary data in LEXXIKO format and creates sets of indices on the values for all the selected attributes so that the lexical entries can now be accessed on the basis of their contents and not only their headword. In addition, for bilingual dictionary data, each data field is coded with respect to its language in order to permit the user to select the particular language on which he wishes to operate a query.

The query system provides the user with a series of functions which can be used to access a dictionary and look-up various lexical elements or combinations of elements. The user can search given items or character strings, define search functions in which items or character strings are associated in different ways, retrieve all the entries satisfying the search conditions in the dictionary, display, print or store on file all or a selected part of the results, define restriction rules to condition his search and, when working on a bilingual dictionary, select the language on which the search must operate.

Functions to perform the following operations have thus been implemented:

- Search an item
- Search all items containing one (or more) strings of characters
- View results for the item searched
- Define search functions combining more than one item
- Search the dictionary using a pre-defined search function
- Display and change the default parameters
- Select the language of interest
- Display the list of all the items already searched
- Display the currently active item

The scope of these functions is described in this section; their mode of use is illustrated in detail in the Query System User Manual given in the Appendix, which can be used independently of the rest of the report.

The query system has been implemented at the Institute on a Local Area Network of personal computers running under the MS/DOS operating system with a host PC acting as server.

The Search Function

An item can be looked-up in the MLDB by entering it on the keyboard; it can be entered in lower or upper case and without accents as the system searches according to the "blind accent" philosophy, i.e. accents are ignored during the search but in those cases in which ambiguity occurs then the forms retrieved with and without accents are displayed, and the item or items on which to operate the search can be selected interactively. The look-up is made on a Attribute-Value basis. This means that each data field is searched to see whether it satisfies the search conditions and the system returns the occurrences of the item searched for each data field in which it is found. Alternatively, the user can invoke the field code selection command in order to select the particular fields on which he wishes to operate his search.

In the same way, all the items which contain one or more given strings of characters can be searched in a text. When the items satisfying the search conditions have been retrieved, the ones which interest the user can be inserted on the list of items already searched during the session by digiting the number which identifies them on the keyboard. These items can then be used in other searches involving more sophisticated functions.
This type of look-up can be made in the following ways:
s* searches all items beginning with string s;
***s** searches all items ending with string s;
s searches all items containing the string s;
(s/t) searches all items containing the string s or the string t
***(s/t)** searches all items ending with the string s or the string t
(s/t)* searches all items beginning with the string s or the string t

The View Function

The results of a search can be viewed and browsed by issuing the view command. The user selects the results he wishes to view in detail and can then choose to view all or just some of their occurrences in the LDB. The occurrences of each item looked-up are viewed in alphabetic order, Entry by Entry, with the "extended text format", i.e. the Headword plus a section of the Entry containing the item searched. The default length of the section of the Entry viewed is 30 tokens, where tokens are all those elements present in the Entry and include words, field names, punctuation marks, etc.. This length can be changed by the user whenever he wants. By default, all the fields contained in each entry are viewed. Alternatively, the user can select the particular fields on which he wishes to operate his view. The VIEW command is always referred to the current item being queried. The current item can be displayed and changed by the user.

When the VIEW command is issued, the system prompts the user to set the Parameters which determine the results to be viewed. By default, i.e. if no parameter is given, all the occurrences for the

entries in which the current item being searched is found are viewed in alphabetic order.

CEnnn gives an extended view for occurrence number nnn;
CNnnn gives all the occurrences for a given item starting from occurrence number nnn;
CN+nnn gives all the occurrences for a given item starting from the nnn-th occurrence after the last occurrence displayed.
CN-nnn gives all the occurrences for a given item starting from the nnn-th occurrence before the last occurrence displayed.

Definition of complex search functions

A combination of items within an Entry can be searched throughout a dictionary. In this case, the items are grouped into a complex search function, sometimes called a "family", which can then be applied to the dictionary to search all occurrences satisfying the conditions. The logic operators & and / are used to determine the relations between the items to be searched and the round brackets ( ) are used to indicate the order of application. The items to be searched are called using their identification number, which can be seen on the list of previously searched items obtained by entering command LIST.

w1&w2 creates a search function which looks for entries in which both item 1 and item 2 appear;
w1/w2 creates a function which searches entries in which either item 1 or item 2 appear;
(w1/w2)&w3 creates a function which searches entries in which item 3 appears together with either item 1 or item 2.
f1&(w4&w5) creates a function which searches entries in which family 1 appears together with both item 4 and item 5.

Searches using pre-defined functions

Once a complex search function has been defined as described in the previous section, it can be used to look-up the dictionary being

analysed to find all those contexts which satisfy the conditions expressed in the function.
The command uses two parameters, both of which can be altered by the user: the first defines the length of the context and the second determines the maximum permissible distance between two items associated by the function: the default value for both parameters is thirty elements. The value for the maximum distance between two items in a family can be changed to search particular phenomena, e.g. it can be set to 1 to search adjacent items or set to 1000 to ensure that the "family" searched can be found for any entry in the LDB.

Displaying and Setting the Default Parameters

As we have seen, certain functions use parameters whose default values can be altered by the user according to his own needs. There are three such parameters: the first determines the length of the entry context created on request of the user, the value given to the parameter indicates the number of elements which are to be displayed before and after the item for which the entry context is being created; the second parameter determines the maximum number of elements which can separate the items searched in combination; the third parameter identifies the printing device to be used by the DBT search functions, the users can choose between on-line printing of his results during the work session or storing them on a file on disk in printing format ready for later use.

Displaying previously searched items

At any moment during a MLDB work session, a list of all the items (single items or "families") which have been searched so far can be displayed. Each item is listed together with its frequency of occurrence for each data field and is preceded by a number giving its order of appearance. This order number is used to identify the item and is needed for the definition of the complex search functions described above.

Restriction Rules

A function is now being implemented which will make it possible to set restriction conditions on any search function used in system. Each restriction condition consists of a number of rules; the rules are formed using the following elements: 1) logic operators - & and / - indicating the relationship between the rule being defined and the previous rules.; 2) Code for the Field for which the restriction rules will apply; 3) type of rule: B beginning with; E ending with; S containing the string; M mask; = equal to; 4) any string to be used for the value condition

4. A Procedure for the semi-automatic sense linking and merging of the LDBs

Although at the present moment, and not only within the context of ACQUILEX, there is considerable interest in the possibility of using dictionaries prepared by publishing houses in computer typesetting format as sources for the construction of computerized lexicons for NLP systems, this approach has been criticized on the basis of the incompleteness and lack of coherence and systematicness of the source material, prepared for human users and not machines. One of the main aims of the procedure presented here is to meet this criticism by proposing a tool which can be used to create a new type of lexical entry in which the data from each source can be examined, compared and verified, equivalent information can be merged and unified, and redundant information can be eliminated, while information which, for example, is only given in one of the sources will become an integral part of the merged entry. A merged lexical structure of this type will not only be more complete than that represented by the single source LDBs but can be continuously enriched as new LDBs or other data are added to the system. In addition, the lexical information presented in this form is easily accessible for a series of processing and analyses at a higher level.

We are thus now working on creating a new composite data structure in which all the information for each separate graphic form which has been given headword status in at least one of the LDBs could be merged in a single mega-structure. This "super-entry" is divided into a number of different levels. The first distinction is made at the homograph level, the entry is sub-divided for each grammatical category; at a second level, homonym distinctions given in the source dictionaries are represented; the third level is that of the sense divisions.

At a formal level there are a number of problems which must be addressed when mapping entries from different sources onto a common structure and we do not expect to be able to solve them all automatically. Different dictionaries do not necessarily use the same criteria when establishing their headword list. All the possible

variations in homograph and homonym distinctions must be allowed for when creating the composite entry. To a large extent, this is facilitated by the mapping of the lexical entries of the different LDBs onto the Common Lexical Entry which is organized on a "one-entry one-major-part-of-speech" basis and in which the particular homonymic distinctions made in the different source dictionaries are indicated in a special field. Secondly, the links must be established at the sense level and not at the headword level. Again, this is not a simple task because, as is known, the distinction between senses even in two monolingual dictionaries for the same language can differ considerably. Such variations can be caused by differences in size and scope between the dictionaries but may also depend on idiosyncratic decisions taken by different lexicographers. Furthermore, additional problems arise when senses are mapped between monolingual and bilingual LDBs due to different perspectives on the language: senses which may be grouped together in a monolingual have been distinguished in a bilingual on the basis of differences in the target language translations, vice versa, at times the bilingual for convenience lumps more than one sense together when translations of different senses are identical.

The procedures which create the composite entry operate in two steps. First, all the information contained in the separate source dictionaries for a given graphic form, entered by the user on the keyboard, is collected and mapped onto the schema of the "super-entry". In this phase, the information is mapped with respect to the homograph and homonym levels. In the second stage and working for one part of speech at a time, the procedures used for sense mapping will include algorithms which read as input the separate definitions, sense labels and examples from each source entry, identifying and matching identical character strings, word-forms of the same lemma, equivalent genus terms, and synonyms over the different LDBs. Strategies are also being developed to allow the system to make decisions when faced with situations in which, for example, a single sense in one LDB is represented by two or more senses in the others. When sense mapping is not possible because the different entries are not homogeneous then the separate senses from each LDB are listed in sequence.

```
ALBERO
D     Entry ALBERO
D     Homon 1
D      PoS1 SM
D     Sense °
D      Def1 OGNI PIANTA CON FUSTO ERETTO E LEGNOSO CHE SI RAMIFICA
D     Sense °
D      Def1 FUSTO DI LEGNO O DI METALLO PER SOSTENERE PENNONI COFFE VELE
D     Field 4Z
D     Sense °
D      Def1 FORMAZIONE ANATOMICA RICCA DI DIRAMAZIONI E RAMIFICAZIONI
D     Sense °
D      Def1 ALBERO GENEALOGICO
D       Loc L
D     Sense °
D      Def1 ELEMENTO DI FORMA ALLUNGATA ATTO A TRASMETTERE LA POTENZA
D     Field 6H
D     Sense °
D      Def1 INSIEME DEI COMPOSTI DERIVATI DA UNA SOSTANZA
D     Field 6D
D     Sense °
D      Def1 GRAFO PRIVO DI CIRCUITI
D     Field 6E
      Homon 2
G      PoS2 s.m.
G     Sense °
G      Def2 pianta con fusto alto, legnoso, con rami nella parte superiore
G     Examp ▪ da frutto
G     Idiom ▪ di Natale
G      Expl abete che a Natale si addobba con lumi e ornamenti diversi e al
            quale si appendono doni
G     Idiom ▪ genealogico
G      Expl rappresentazione grafica della discendenza di una famiglia
G     Sense °
G     SI fl {mar.}
G      Def2 antenna che regge i pennoni con le vele e tutta l'attrezzatura
G     Sense °
G     SI fl {mecc}.
G      Def2 parte rotante, generalmente cilindrica, che, in una macchina, ha
            la funzione di trasmettere potenza meccanica da un organo a un altro
G     Examp ▪ a gomito
G     Examp ▪ motore
G      Expl che serve a trasmetter moto da motore a ruote, eliche o simili
      Homon 3
C      PoS3 sm
C     Sense °
C       SIs pianta
C     Df/Tr tree;
C     Examp =**= da frutto
C       ExT fruit tree;
C     Examp =**= genealogico
C       ExT family tree;
C     Examp =**= di Natale
C       ExT Christmas tree;
C     Examp =**= del paradiso
C       ExT tree of heaven;
C     Examp =**= della vita
C       ExT white cedar.
C     Sense °
C        Fl Naut
C     Df/Tr mast;
C     Examp =**= maestro
C       ExT mainmast.
C     Sense °
C        Fl Tecn
C     Df/Tr shaft;
C     Examp =**= a camme o della distribuzione
C       ExT camshaft;
C     Examp =**= motore o a gomiti
C       ExT crankshaft.
```

Figure 8 First mapping of composite entry for "albero"

```
Dizionario Unificato MIS                                        ALBERO
-------------------------------------------------------------------------

D     Entry ALBERO2
D      PoS1 SM
      Sense °
D      Def1 OGNI PIANTA CON FUSTO ERETTO E LEGNOSO CHE SI RAMIFICA
G      Def2 pianta con fusto alto, legnoso, con rami nella parte superiore
C       SIs pianta
C     Df/Tr tree;
GC    Examp ▪ da frutto
C       ExT fruit tree;
GC    Idiom ▪ di Natale
G      Expl abete che a Natale si addobba con lumi e ornamenti diversi e al
            quale si appendono doni
C       ExT Christmas tree;
C     Examp ▪ del paradiso
C       ExT tree of heaven;
C     Examp ▪ della vita
C       ExT white cedar.
      Sense °
D      Def1 FUSTO DI LEGNO O DI METALLO PER SOSTENERE PENNONI COFFE VELE
D     Field 4Z
G     SI fl {mar.}
G      Def2 antenna che regge i pennoni con le vele e tutta l'attrezzatura
C        Fl Naut
C     Df/Tr mast;
C     Examp ▪ maestro
C       ExT mainmast.
      Sense °
D      Def1 FORMAZIONE  ANATOMICA RICCA DI DIRAMAZIONI E RAMIFICAZIONI
      Sense °
D      Def1 ALBERO GENEALOGICO
D       Loc L
GC    Idiom ▪ genealogico
G      Expl rappresentazione grafica della discendenza di una famiglia
C       ExT family tree;
      Sense °
D      Def1 ELEMENTO  DI FORMA ALLUNGATA ATTO A TRASMETTERE LA POTENZA
D     Field 6H
G     SI fl {mecc}.
G      Def2 parte rotante, generalmente cilindrica, che, in una macchina, ha
            la funzione di trasmettere potenza meccanica da un organo a un altro
C        Fl Tecn
C     Df/Tr shaft;
G     Examp ▪ a gomito
G     Examp ▪ motore
G      Expl che serve a trasmetter moto da motore a ruote, eliche o simili
C     Examp ▪ a camme o della distribuzione
C       ExT camshaft;
C     Examp ▪ motore o a gomiti
C       ExT crankshaft.
      Sense °
D      Def1 INSIEME DEI  COMPOSTI  DERIVATI DA UNA SOSTANZA
D     Field 6D
      Sense °
D      Def1 GRAFO PRIVO DI CIRCUITI
D     Field 6E
-------------------------------------------------------------------------
```

Figure 9 Automatic proposal of information merging between the DMI, Garzanti and Collins for "albero"

Figure 8 gives an example of the first stage of the mapping for the Italian lemma **albero** (tree) where all the information from the three dictionaries is grouped together under the same POS.

In Figure 9, we see how the hypothesis of information merging is proposed by the system. The first definition for the DMI has been matched with the first definition from Garzanti because of the cooccurrences of **pianta, fusto** and **rami**... The first sense of Collins has been matched to these definitions because of the presence of the Semantic Indicator **pianta** and also because of the examples **albero da frutto** and **albero di Natale** which match against the examples from Garzanti.

The second definition of the DMI has been matched with the second definition of Garzanti because of the cooccurrence of **pennoni** and **vele** in both definitions. The second sense of Collins will be matched here because the normalization of the sense labels of our dictionaries will give a standard subject code NAUT for both labels "mar" and "Naut". In addition, this definition in the DMI has a subject code 4Z which will also be normalized as NAUT.

The algorithms for the rest of the matching will continue in the same way, proceeding definition by definition. The starting point is always the DMI definition as the DMI is the largest and richest dictionary; when it is not possible to match anything to a particular definition the procedure will continue with the following one. The result will be a composite entry in which information from the three dictionaries has been linked and is ready for successive analyses.

At this point, the user can call for the composite entry to be displayed on the screen. If he is not satisfied with the sense mapping which is proposed by the system he can modify it using the user friendly editor developed for the MLDB system (see Section 2 above) which is being extended by the addition of functions to guide and assist him in his choices for the mapping between senses. During the session, the user can create as many composite entries as he likes. At the end of the session, he has the possibility of saving all or part of

his results on disk and/or printing them out ready for further studies and analyses. In addition, the new entries created can go to form part of a new merged LDB on which all the access and interrogation functions of the multi-lexical system can be used.

We now intend to improve the performance of the system by studying and developing more sophisticated algorithms to achieve a more efficient compacting and merging of the information contained in separate definitions referring to the same word sense.

APPENDIX

MLDB QUERY SYSTEM
USER MANUAL
(Preliminary Version)

The data query system has been designed with the non-expert user in mind. The commands have been implemented in a simple, user friendly fashion. The objective has been to permit the user to access any of the LDBs contained in the MLDB and to navigate freely throughout them in order to retrieve any lexical item or combination of items, to display and print his results or to store them on disk for future studies and analyses. The intention is to offer an exhaustive dictionary access and to give the user the possibility to search, browse and retrieve large quantities of lexical information for in-depth studies which would otherwise have been impossible.

1. Opening a QUERY Session

The MLDB system is accessed by issuing the command:

MLDB

The first step is to select the LDB to be queried. The system prompts the user to enter the appropriate LDB name:

COLLINS, GARZANTI, DMI, MISC

(where MISC stands for the derived LDB described in Section 4; the query system is not yet active on this LDB)

On access to the bilingual dictionary, the default language on which searches are to be operated is Italian. It is possible change the language selected at any moment during a bilingual dictionary query session using the LANGUAGE function under the SETUP command

I selects Italian as the language on which the search is to be operated;
E selects English

Throughout the LDB query session, the user is guided by a menu of commands which is displayed at the bottom of the screen and accompanied by a brief Help.

The main menu, which is displayed once the LDB has been selected is as follows:

SEARCH VIEW FAMILY LIST DISPLAY PREVIOUS NEXT SETUP EXIT

The scope of the commands has been described in detail in the previous section; this manual will guide the user in an LDB query session.

A command is selected either by moving the cursor along the menu line to the chosen command and then pressing the Enter key or by digiting the capital letter corresponding to the key letter of the command to be chosen. When the cursor is positioned on a function, a brief Help message describing the scope of the command selected appears below the Menu line.

During the query session, the commands which can be activated at any given moment and their key letters are evidenced. For instance, at the beginning of the session, when the main menu is displayed for the first time, the commands SEARCH and EXIT appear highlighted as they are the only possible choices.

2. Querying the MLDB

To begin a query session the SEARCH command must be selected.

2.1 SEARCH

When this command is issued, the system prompts the user to enter the value (item or character string) to be looked-up.

The value is entered on the keyboard; it can be entered in lower or upper case and without accents. In cases of ambiguity then the forms retrieved with and without accents are displayed, and the item or items on which to operate the search can be selected interactively

Searches are made on a Attribute-Value basis. This means that the item or character string entered as value is searched with reference to its particular function in the entry. Thus the search will be made through all the fields contained in the LDB entries and the results will be displayed Field by Field. Alternatively, the user can invoke the field code selection command under the SETUP function in order to select the particular fields on which he wishes to operate his search. See FIELD-SEARCH under SETUP below.

The following types of look-up can be made:

xxx searches all items matching the item xxx

s* searches all items beginning with string s (e.g. CONTR* searches all the items which begin with the string CONTR);

*s searches all items ending with string s (e.g. *TORE searches all items ending with the string TORE);

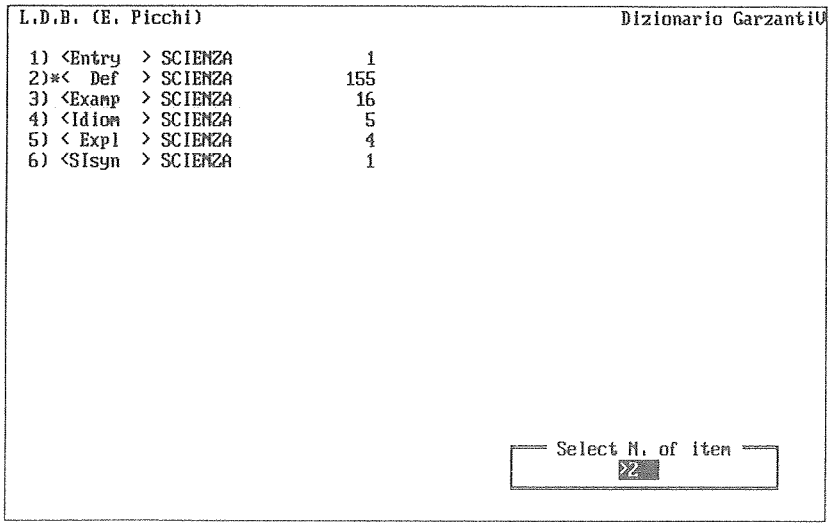
s searches all items containing the string s (e.g. *IGLI* searches all the items which contain the group of letters IGLI);

(s/t) searches all items containing the string s or the string t (e.g. *(GL/LG)* searches all items containing either GL or LG);

*(s/t) searches all items ending with the string s or the string t (e.g. *(TORE/TRICE) searches all items ending with the string TORE or the string TRICE, Italian suffixes typically used to denominate professions).

(s/t)* searches all items beginning with the string s or the string t (e.g. *(?/?) searches all items beginning with the string xxx or the string xxx, Italian prefixes typically used to denominate xxx).

Here below we show the display of the results for a search for the word "scienza" in the Garzanti LDB. It can be seen that the results are given with the number of occurrences of the value looked-up for each attribute in which it has been found.



The menu of functions available in the SEARCH environment are:

CONTINUE INTERRUPT SELECT

INTERRUPT the display of the results is interrupted and the system returns to the main menu.

CONTINUE during the display of the results, this command calls the next page; at the end it returns the system to the main menu

SELECT selects the results of a query which are to be saved for successive operations. The items chosen using SELECT are inserted in a list in the order in which they are selected. The order number given to each item is used to identify it subsequently. The user is prompted to enter the identification number for the result he wishes to select. For example, the user who wishes to select the occurrences of "scienza" for the Definition fields shown in the figure above will enter no. 2. The list of results selected can be viewed at any time by issuing the LIST command from the main menu environment.

When the fields to be viewed have been selected, the INTERRUPT or the CONTINUE command will return the user to the main menu:

SEARCH VIEW FAMILY LIST DISPLAY PREVIOUS NEXT SETUP EXIT

2.2 VIEW

This command is used to view and browse the results of dictionary searches. The occurrences of each item looked-up are viewed in alphabetic order, Entry by Entry, with an "extended text format", i.e. the Headword plus a section of the Entry containing the item searched. The length of the section of the Entry viewed can be changed using the ENTRY DISPLAY-SIZE command under SETUP. The default length is 30 tokens, where tokens are all those elements present in the Entry and include words, field names, punctuation marks, etc..

By default, all the fields contained in each entry are viewed. Alternatively, the user can invoke the field code selection command under the SETUP function in order to select the particular fields on which he wishes to operate his view. See FIELD-VIEW under SETUP below.

The VIEW command is always referred to the current item being queried. The current item can be displayed using the DISPLAY command or changed by using the SELECT, UP and DOWN commands

under LIST, or the PREVIOUS and NEXT commands in the main menu environment.

When the command is issued, the system prompts the user to set the Parameters which determine the results to be viewed. By default, i.e. if no parameter is given, all the occurrences for the entries in which the current item being searched is found are viewed in alphabetic order.

CEnnn	gives an extended view for occurrence number nnn;
CNnnn	gives all the occurrences for a given item starting from occurrence number nnn;
CN+nnn	gives all the occurrences for a given item starting from the nnn-th occurrence after the last occurrence displayed.
CN-nnn	gives all the occurrences for a given item starting from the nnn-th occurrence before the last occurrence displayed.

When the parameters have been entered, the following menu appears at the bottom of the screen.

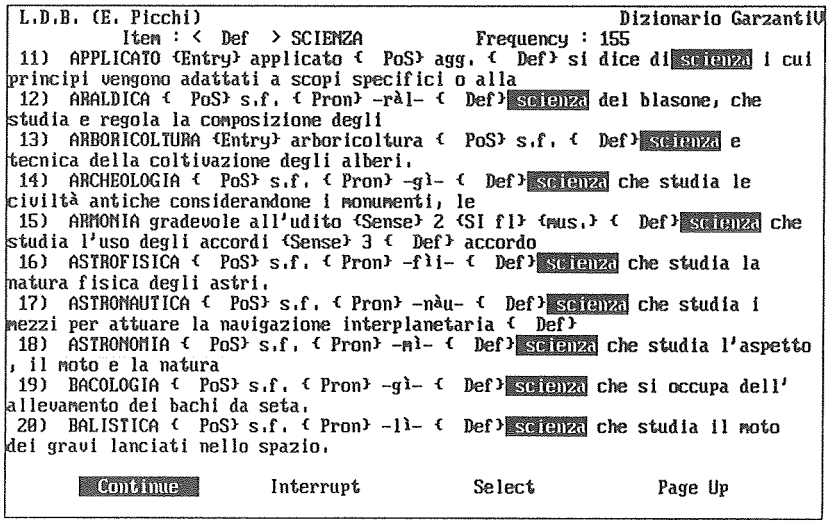
CONTINUE INTERRUPT SELECT

CONTINUE: display of results continues from current occurrence downwards

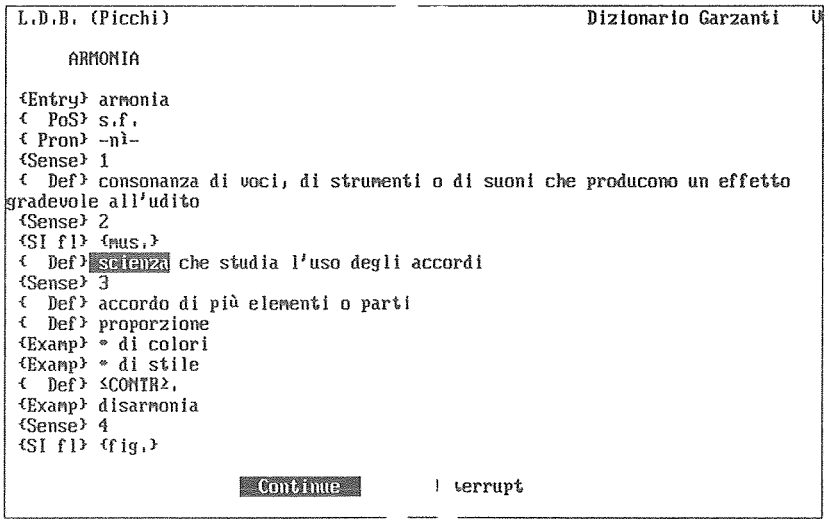
INTERRUPT: interrupt of VIEW and return to the main menu; the next VIEW will display the results beginning from the occurrence immediately following the last item viewed.

SELECT: the entire entry for the chosen occurrence is displayed in Dictionary Access Format. Again, by default, all the fields contained in each entry are viewed. Alternatively, the user can invoke the field code selection command under the SETUP function in order to select the particular fields on which he wishes to operate his view. See FIELD-VIEW under SETUP below.

Here below we show the display of the selected results for the previous query in extended text format.



Below, the entry for armonia has been selected using the SELECT function and is displayed in dictionary access format.



2.3 FAMILY

This command is used for the creation of complex search functions, known as "families", in which occurrences of a combination of two or more given items in the same entry are looked-up. When it is issued the system prompts the user to:

DEFINE FAMILY
or SEARCH FAMILY

DEFINE FAMILY

Families are defined using the codes w for item and f for family, and the AND and OR operators: & and /

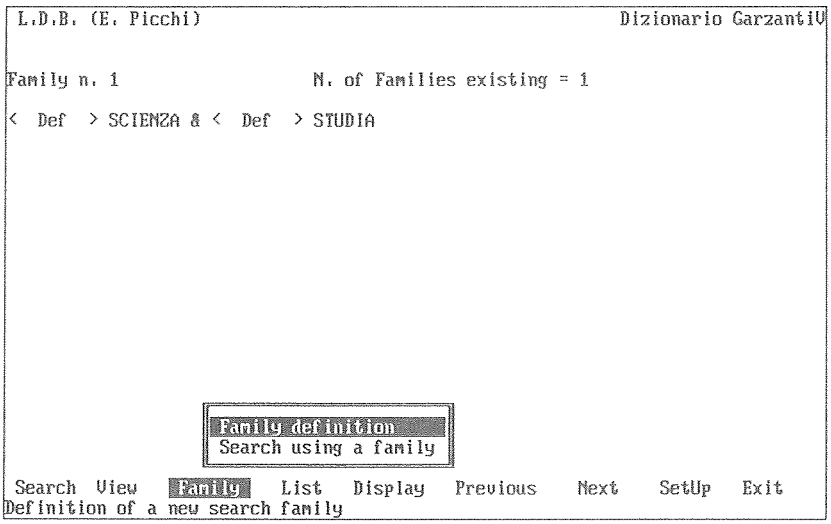
w1&w2 creates a search function which looks for entries in which both item 1 and item 2 appear;

w1/w2 creates a function which searches entries in which either item 1 or item 2 appear;

(w1/w2)&w3 creates a function which searches entries in which item 3 appears together with either item 1 or item 2.

f1&(w4&w5) creates a function which searches entries in which family 1 appears together with both item 4 and item 5.

In the next figure, we show a display of a complex search function which can be used to search all entries in which scienza and studia cooccur in the definition field.

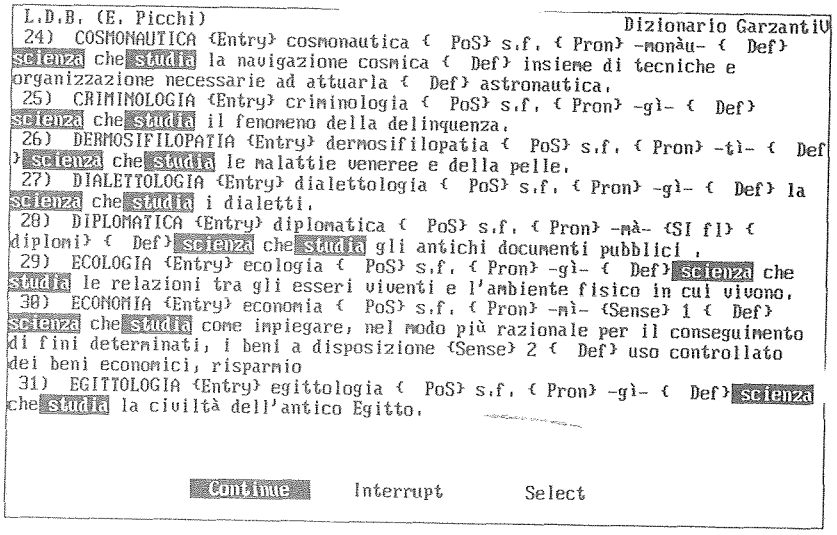


SEARCH FAMILY

This command is used to apply a previously created "family" in a dictionary look-up. When the command is issued, the user is prompted to enter the identification number of the family to be searched. This can be displayed by issuing the **DISPLAY-FAMILY** command under **LIST**.

The occurrence of a "family" is found in an entry when the distance between the items searched in combination is equal to or less than the "distance" parameter determined by the **COOCCURRENCE** command under **SETUP**. The default value for the maximum distance between two items in a family is thirty elements. This value can be changed to search particular phenomena, e.g. it can be set to 1 to search adjacent items or set to 1000 to ensure that the "family" searched can be found for any entry in the LDB.

The figure below shows the results of the search using the **scienza** and **studia** family defined above.



2.4 LIST

When this command is issued, a list of all the results which have been selected using the SELECT function described above under SEARCH is displayed. As results are selected, they are assigned an order number and this is used to identify them when they are to be used in a successive operation.

The following menu appears at the bottom of the screen:

CONTINUE SELECT UP DOWN DISPLAY-FAMILY

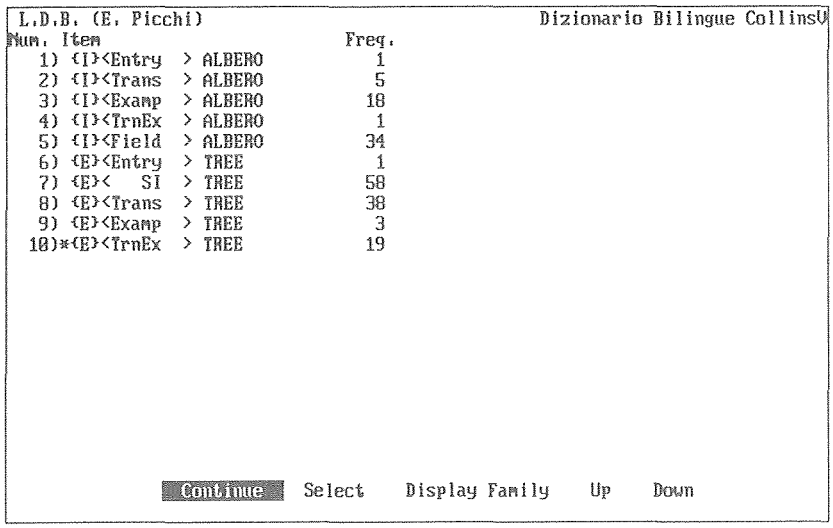
CONTINUE: if this command is issued the user returns to the main menu.

SELECT: the user is prompted to select the item searched which he wishes to use in a successive operation. The system then returns to the Main Menu.

UP: the previous item becomes the current item

DOWN: the following item becomes the current item

DISPLAY-FAMILY: the list of previously defined families is displayed The figure shows a List of all the searches made in Collins for albero and tree for both English-Italian and Italian-English.



2.5 DISPLAY

When this command is issued, the current item is displayed and the system returns the Main Menu.

2.6 PREVIOUS

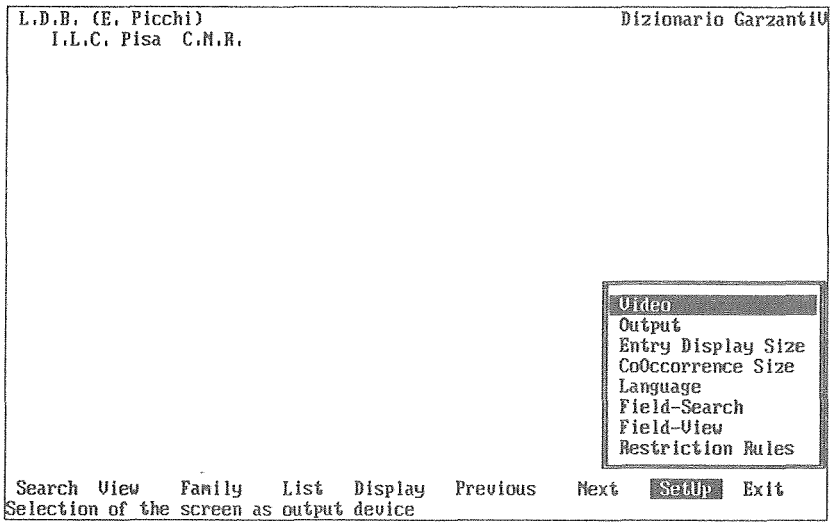
When this command is issued, the previous item becomes the current item

2.7 NEXT

When this command is issued, the successive item becomes the current item

2.8 SETUP

This command is used to set or reset some of the query system's parameters.
When the command is issued, the following menu is displayed:



VIDEO: the screen is selected as the output device for the display of results and the current parameters for the length of the sections of the entries to be displayed under VIEW and the maximum distances between items in a FAMILY are displayed. A V appears in the upper right corner of the screen.

OUTPUT: the system prompts the user to select and set his output device. The choice is PRINTER, FILE, CLOSE PRINT QUEUE

PRINTER will select the printer as the output device for the selected results; a P appears in the upper right corner of the screen.

FILE will select a file on disk as the output device for the selected results which are being printed; an F appears in the upper right corner of the screen. At the beginning of each MLDB query session, a file is opened and all results selected by issuing the FILE command are automatically memorized in this file and its name is displayed. A new file is assigned for each query session.

CLOSE PRINT QUEUE will close the output device selected.

ENTRY DISPLAY-SIZE: when this command issued, the parameter which governs the length of the length of the sections of the entries to be displayed under VIEW can be changed. The system prompts the user to enter the new value. The default value is 30 items.

COOCCURRENCE when this command is issued, the parameter which governs the maximum distance between items which are combined in complex search functions can be changed. The system prompts the user to enter the new value. The default value is 30 items.

LANGUAGE: this command is used to select the language on which the search functions must operate in the bilingual LDB.

I selects Italian
E selects English

For the Collins LDB, the default language is Italian.

FIELD-SEARCH: when this command is issued, the fields on which the SEARCH function is to operate can be selected. All the fields which have been indexed are listed:

SELECT ALL: selects all fields

UNSELECT ALL: unselects all fields so that the particular fields of interest can then be selected, one by one, by positioning the cursor on them and pressing the Enter key.

ESC: accepts the fields selected and returns to the main menu

FIELD-VIEW: similar to the above command, the fields on which the VIEW function is to operate can be selected. All the fields present in the LDB entry are listed:

SELECT ALL: selects all fields

UNSELECT ALL: unselects all fields so that the particular fields of interest can then be selected, one by one, by positioning the cursor on them and pressing the Enter key

ESC: accepts the fields selected and returns to the main menu

RESTRICTION-RULES: this function permits the setting of restriction rules on any search function used in the system; e.g. with reference to the normalization of the POSs of the different dictionaries, the following restriction rule can be imposed: NORM_POS = n thus permitting the search functions to operate only on entries with this grammatical category.

The restriction rules can consist of several conditions linked using the AND and OR operators: & and /

When the command is issued, the following menu appears:

ACCEPT ADD MODIFY DELETE UP DOWN DISABLE

ACCEPT: returns to main menu, activating the set of restriction rules which have been imposed.

ADD: a restriction condition is imposed. Each restriction condition consists of a number of rules; the rules are formed using the following elements: 1) logic operator; 2) Code for the Field for which the restriction conditions must apply; 3) type of rule; 4) any string to be used for the value condition
where

1. logic operators - & and /: the user is prompted to choose the operator to be used
2. field code: the set of fields which form the lexical entry are displayed in a window and the restrictions can be imposed by positioning on the chosen field and pressing the Enter key to select it.
3. type of condition - a choice of conditions is displayed:
   B beginning with
   E ending with
   S containing the string
   M mask
   = equal to
4. the string to be used with the condition type displayed above

MODIFY: an existing restriction condition is modified

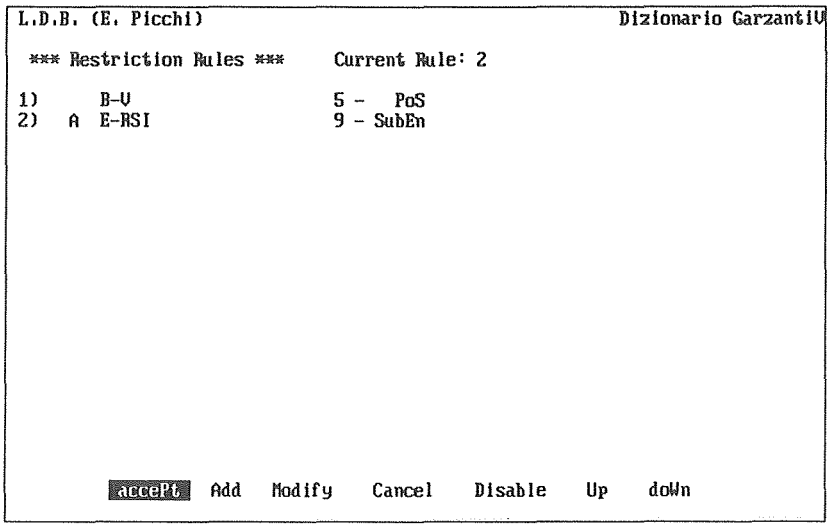
DELETE: an existing restriction condition is deleted

UP: the previous restriction condition becomes the current condition

DOWN: the following restriction condition becomes the current condition

DISABLE: the entire set of existing restrictions plus any filter on the lexical search functions are deleted

In the following figure we show an example of a Restriction condition which restricts a search to all verbs ending with "rsi" (indicating Italian reflexive or pronominal verbs).



2.9 EXIT

When this command is issued, the user is asked to confirm that he wishes to exit from the MLDB by entering Y or N.

References

Dictionaries

Il Nuovo Dizionario Italiano Garzanti, Garzanti, Milano, 1984.

Collins Concise English-Italian, Italian-English Dictionary, Collins, London and Glasgow, 1985.

Longman Dictionary of Contemporary English (LDOCE), P. Procter et al. (eds.), Longman, Harlow and London, 1978.

Zingarelli N. (1970), *Vocabolario della Lingua Italiana*, Zanichelli, Bologna, 1970.

Others

Boguraev B.K. (1986), "Machine-Readable Dictionaries and Research in Computational Linguistics", paper presented at the Workshop "On Automating the Lexicon", Grosseto, Italy, 1986.

Boguraev B., Briscoe E., Calzolari N., Cater A., Meijs W., Picchi E., Zampolli A. (1989), "ACQUILEX: Acquisition of Lexical Knowledge for Natural Language Processing Systems", Technical Annexe for Esprit Basic Research Action no. 3030, April 1989.

Boguraev B., Briscoe E, Carroll J., Copestake A. (1990), "Database Models for Computational Lexicography", accepted for the EURALEX Conference, Malaga, Spain, 28-31 August 1990.

Calzolari N., Picchi E. (1986), "A project for a bilingual lexical database system", Advances in Lexicology of the UW Centre for the New Oxford English Dictionary, Waterloo, Canada, November 1986.

Calzolari N., Peters C., Roventini A. (1990), "Computational Model of the Dictionary Entry: Preliminary Report", ACQUILEX ESPRIT B.R.A.3030, Six Month Deliverable, ILC-ACQ-1-90.

Gruppo di Pisa (1979), "Il Dizionario di Macchina dell'Italiano" in Linguaggi e Formalizzazioni, D. Gambarara, F. Lo Piparo, G. Ruggiero (eds.), Roma, Bulzoni, 1979, pp.683-707.

Picchi E. (1983), "Textual Database" in Proceedings of the International Conference on Data Bases in the Humanities and Social Sciences, Rutgers University Library, New Brunswick, New Jersey, USA,1983.

Picchi E. (1990), "D.B.T.: A Textual Data Base System" in Cignoni, L., Peters, C. (eds.), "Computational Lexicology and Lexicography. Special Issue dedicated to Bernard Quemada. I", Linguistica Computazionale, Vol VI, 1990., to appear.

Picchi E., Peters C., Calzolari N. (1990), "Implementing a Bilingual Lexical Database System", in BUDALEX '88 Proceedings, Papers from the EURALEX Third International Congress, Budapest, 4-9 September 1988, T. Magay and J.Zigány (eds.), 1990.

Zampolli A. (1983), "Lexicological and Lexicographical Activities at the Istituto di Linguistica Computazionale", in "The Possibilities and Limits of the Computer in Dictionary Producing and Publishing", Proceedings of the European Science Foundation Workshop, Pisa, 1981, A. Zampolli, A. Cappelli (eds.), Linguistica Computazionale, Vol III(1983), p.237-278.