WP3

Dictionary Browsing Tool Requirements

Carol Peters, Eugenio Picchi

internal use only

15 November 1991

Date	Version	Author	Comments
15/11/1991	1.0	C. Peters, E. Picchi	

MULTILEX

TITLE	: Dictionary Browsing Tool Requirements
PROJECT	: MULTILEX
AUTHOR	: Carol Peters*, Eugenio Picchi**
ESTABLISHMENT	: *IEI-CNR, **ILC-CNR, Pisa University
ISSUE DATE	: 15 November 1991
DOCUMENT ID	: MWP3
VERSION	: 1.0
STATUS	: final
WP NUMBER	: WP3
DISTRIBUTION	: MULTILEX Partners
LOCATION	: PSA
KEYWORDS	: Computational Lexicography, Lexical Database Tools

ABSTRACT

The main features of a dictionary browsing tool designed to satisfy the requirements of the linguist or lexicographer working on machine-readable dictionary data are discussed. Examples are given of the results which can be obtained using existing tools, and recommendations are made regarding the functions which should be included in a dictionary browsing tool developed within the MULTILEX project.

DICTIONARY BROWSING TOOL REQUIREMENTS

Carol Peters[1], Eugenio Picchi[2]

[1]Istituto di Elaborazione della Informazione, CNR, Pisa, Italy
[2]Istituto di Linguistica Computazionale, CNR, Pisa, Italy

1. INTRODUCTION

The scope of a dictionary browsing tool is to provide the user, i.e. linguist or lexicographer, with fast and flexible access to all the lexical information contained in a machine-readable dictionary: phonetic, morphological, syntactic, semantic and conceptual. He/she must be able to navigate through the dictionary data and within the different fields which constitute the entry in order to retrieve, view and manipulate information of interest in whatever part of the dictionary it is stored. Ideally, in order to optimize the consultation process, the user will also have access to morphological procedures, to modules which permit direct access from dictionary data to on-line text and language reference corpora and vice versa, and to procedures which permit linking between different mono- and/or bilingual electronic dictionaries so that data in equivalent entries, coming from different sources, can be easily analysed and compared.

The aim of the present report is to describe the type of access operations which the user needs to perform on lexical data and the type of tool which can be implemented to satisfy such requirements. The browsing tool is envisaged as part of a more complex dictionary query system which also maintains procedures for other operations on the lexical data. The report has been written mainly in consideration of the requirements of the lexicographer and therefore emphasis is placed on the design of a user-friendly tool which can be implemented on small, inexpensive desk-top systems, e.g. personal computers running the MS/DOS operating system or from the Apple Macintosh series. At the same time, the procedures should be easily transportable to more powerful systems such as SUN machines or systems running in the UNIX environment. Examples are given of typical browsing functions in operation on a lexical database system and the report concludes with recommendations regarding the functions to be included in a dictionary browsing tool developed within the MULTILEX project.

2. STRUCTURING DICTIONARY DATA

As stated, a dictionary browsing tool will supply the user with dynamic search procedures to freely access, search, view and display all types of machine-readable dictionary data according to his particular needs. It is assumed that the lexical data contained in the machine-readable dictionary has already been parsed and structured on a representation schema, according to a previously defined computational model.

The underlying structure is crucial as it will condition the kinds of access and browsing functions which are possible on the data. In this section we will briefly present the types of computational model and data representation schema which can be adopted to structure dictionary data in lexical database form and then mention problems which must be considered when the lexical information encoded in the dictionary entry is extracted and mapped onto the data model.

2.1 Computational Model

There is still no firm consensus on what would constitute a general computational model of a dictionary. Boguraev et al. (1990) identify four possible classes of dictionary models: the relational model; the tagged representation; the hierarchical model; and a "two-level hierarchical-tagged" model. The classical database relational model, in which the data is mapped directly to a set of tables, is little suited to handling dictionaries given the complex nature of the lexical entry where the attributes generally contain non-normalised text rather than simple scalar values. The tagged dictionary representation preserves all of the information contained in the entry, explicitly labelling each information field to facilitate successive retrieval, but on its own does not satisfactorily represent the structural relationships which were implicit in the source. The hierarchical model is considered far more appropriate to represent the lexical intuitions encoded in the typical dictionary entry as it permits the clustering and nesting of information at different levels; lexical entries can be characterised as shallow hierarchies of attribute-value pairs with a variable number of instances of specific nodes at each level, e.g. multiple homographs within an entry or multiple senses with a homograph (see Neff et al., 1987). The tagged representation is thus generally combined with the hierarchical model to provide an adequate representation of the dictionary. In the two-level hierarchical-tagged model adopted by the ACQUILEX project[1], the source dictionary is seen as the primary repository of lexical data and the database system automatically generates from it subsidiary sets of indices which encode all the information that it has been decided to represent in the LDB (see Carroll, 1990).

2.2 Representation Schema

Together with the discussion on the most suitable computational model for the dictionary, there has been much recent interest in the adoption of a standardized language to represent the information contained in the dictionary entry. The most important international activity currently under way in this field is the Text Encoding Initiative, co-sponsored by the Association for Computational Linguistics (ACL), Association for Literary and Linguistic Computing (ALLC), Association for Computing in the Humanities (ACH) and by the European Community. This is including a study on the definition of a common representation for the dictionary

[1]The aim of ACQUILEX, ESPRIT BRA 3030, is to develop techniques and methodologies for utilising both monolingual and bilingual machine-readable dictionary sources to construct lexical components for natural language processing systems (see Boguraev et al., 1987, for a description of the scopes and goals of the project).

entry as part of its general programme to provide guidelines and standards for the representation and exchange of texts in machine-readable format using the SGML mark-up language (see TEI, 1990, and Ide et al., 1991).

Much work has also been done in this area for the ACQUILEX project and is described in (Calzolari et al., 1990). These authors stress that the definition of an explicit and uniform representation language for machine-readable dictionaries is essential in order to permit the exchange of data between projects in a common format, to facilitate uniform analyses over different dictionaries, and to write generalized parsers for different dictionaries. A standard representation methodology and common lexical entry template has been designed for both mono- and bilingual entries for the ACQUILEX Project Database. The ACQUILEX template has been organized on a 'one-entry one-major-part-of-speech' basis in which the particular relations between lexical items given implicitly in the source dictionaries (such as lexical or grammatical homograph, variant or derivative form) are indicated in a special field.[2]

Whatever the representation schema used, three main categories of information are normally identified in the dictionary entry: orthographic/phonetic information; morpho-syntactic information; semantic information (which in the case of bilingual and multilingual dictionaries or lexicons will also include translation data). Examples of data representations taken from different lexical database projects are given in Figures 1a, 1b and 1c. The kind of lexical information which is encoded in the dictionary entry for each of these categories has been discussed in the papers of Work Package 2 and we will not go into detail here. What is important from our viewpoint is that all the information which is encoded, explicitly and implicitly, in the source dictionary entry and to which access paths are to be constructed, is extracted and mapped onto the chosen data model. Lookup on structured dictionary data is made on an attribute-value basis and the browsing tool query processor will operate on the attribute-value pairs which are recognized during the data structuring and parsing stages. The attributes are defined in the representation schema; the values will consist of the (textual and coded) data derived from analyses of the dictionary entry.

2.3 Parsing and Analysing the Lexical Entry

The dictionary parsing and analysis procedures will be complex, firstly because the lexical information contained on the source tape is mixed with other data (e.g. formatting commands if a type-setting tape is being used), and secondly because much of the information stored in the entry is given only implicitly. Parsing of the source tape thus normally consists of more than one phase. The goal of the first stage is to segment the information on the tapes into separate entries and to identify the main information fields in the entry, storing the results on a direct access file in which the access key is the dictionary headword and the data is simply the body of the entry. The second stage will then consist of a more painstaking analysis of each

[2]Other studies on the entry structures of mono- and bilingual dictionaries can be found in Amsler and Tompa, 1988, and Fought and Van Ess-Dykema, 1990.

```
entry                   /*root of the tree*/
+-hdw                   /*English headword; key to the DAM file*/
|
+-superhom              /*superscript homograph*/
  +-hum                 /*superscript number from printed dictionary*/
  +-pron                /*encoded pronunciations*/
  |
  +-altspel             /*alternate spelling*/
  | +-note              /*e.g. "U.S. English"*/
  | +-spel              /*the alternate spelling string*/
  |
  +-hom                 /*homograph*/
    +-hnom              /*homograph number*/
    +-pos               /*part-of-speech*/
    +-morph             /*irregular inflections, etc.*/
    |
    +-sens              /*sense*/
      +-snum            /*sense number*/
      |
      +-xlat            /*translation information for headword*/
      | +-note          /*usage note for the English term*/
      | +-spel          /*the Italian translation*/
      | +-gnd           /*grammatical information about the Italian*/
      |
      +-xmp             /*"example" phrases containing the headword*/
      | +-note          /*usage note for the English phrase*/
      | +-gloss         /*the English phrase*/
      | +-pos           /*part-of-speech*/
      | +-expl          /*more usage information*/
      | +-tran          /*Italian translation of phrase*/
      | +-gnd           /*grammatical information about the Italian*/
```

Figure 1a) LDB Design for the Collins English-Italian dictionary at IBM (from Neff et al., 1987)

syn

cat - syntactic category
c1 - first compound field
c2 - second compound field
gcode - grammar code
label - label field
multiple - number of words in headword field, and whether hyphenated

sem

antonym - from the definition field (after---opposite)(lowercased)
box - box codes
class - semantic head of definition, as determined by Alshawi parser
defn - 'implicit' x-refs in definition (after ---see etc.)(uppercased)

pron

nsylls - number of syllables
s1, s2, s3, etc. where s1 is the first syllable etc.
stress - stress for this syllable
onset - phonemes at the syllable onset
peak - phonemes at the syllable peak
coda - phonemes at the syllable coda

Figure 1b) Attribute name hierarchy and meanings of terminal attributes for LDOCE (from Carroll, 1990)

ENTRY_GROUP

- HEADWORD_GROUP
  - RELATED_ENTRY_GROUP
  - PHONETIC_GROUP
  - ETYMOLOGY_GROUP
  - SENSE_GROUP
    - CROSS_REFERENCE_GROUP
    - GRAM_INF_GROUP
      - INFLECTION_GROUP
    - SENSE_LABEL_GROUP
      - SEMANTIC_FEATURES_GROUP
      - SEMANTIC_LABEL_GROUP
      - SEMANTIC_INDICATOR_GROUP
    - DEFINITION_GROUP
      - TAXONOMY_GROUP
    - TRANSLATION_GROUP
    - EXAMPLE_GROUP
      - EXAMPLE_TRANS_GROUP
    - MULTIWORD_GROUP
      - PROVERB_GROUP
    - SEMANTIC_RELATIONS_GROUP
      - SUPERORDINATE_GROUP
      - SYNONYM_GROUP
      - ANTONYM_GROUP
      - ALTERATE_GROUP

Figure 1c **Abbreviated Schema of ACQUILEX Common Lexical Entry (only main node tags are given without dependent attributes)** (see Calzolari et al., 1990, for the complete representation schema)

field in order to extract the lexical information as specified by the data model. For instance, in the first stage all grammatical information on the headword may have been stored directly under a general label; this information will then be re-analysed to extract, for a verb for example, information on category, morphology, subcategorization, selectional restrictions on the arguments, and so on. During this stage, a number of problems must be tackled; these are problems which arise typically when data which has been prepared for human consultation is subjected to electronic processing. They include the *reconstruction of compressed information,* e.g. a headword rendered as 'curieux, -ieuse" in the source dictionary must be expanded in the database into two distinct items, "curieux" and curieuse"; the *disambiguation of implicit information*, e.g. given a headword such as "defogliante, defoliante", the second form must be recognized as a less common variant form and mapped to a specific field on the data schema; *the factoring out of compacted information*, e.g. an example such as "brilla per la sua bellezza/intelligenza" could become two separate examples: "brilla per la sua bellezza" and "brilla per la sua intelligenza"

Successive stages of parsing usually regard more intensive analyses of the definition data to extract semantic information. Dictionaries are rich sources of such information and extraction procedures which analyse the structure of the definitions in terms of their genus and differentiae can be used to extract data on semantic relations, in particular taxonomies, and on other conceptual categories (see Calzolari and Picchi, 1988). Such information can be stored directly in a lexical data base and is available for subsequent formalisation in a lexical knowledge base (see Section 2.4 below). An example of a parsed and structured lexical database entry is given in Figure 2.

Hdwd	1	cucina
PoS	2	s.f.
NPoS	3	N
Sen_No	4	1
Def	5	stanza in cui si preparano e si cuocino i cibi
Genus	6	STANZA
Sen_No	7	2
Def	8	azione del cucinare
Genus	9	AZIONE
Ex	10	far da
Explan	11	cucinare
Sen_No	12	3
Def	13	modo di cucinare, e anche i cibi cucinati
Genus	14	MODO
DifRel	15	CUCINARE MODE_OF
Ex	16	francese
Ex	17	grassa
Ex	18	magra
Sen_No	19	4
Def	20	apparecchio per la cottura dei cibi
Genus	21	APPARECCHIO
Ex	22	a gas
Ex	23	a legna
Ex	24	elettrica

Figure 2 Example of Parsing and Mapping onto a Database Representation Schema for the Italian lemma "cucina" from the Garzanti dictionary

2.4 Typed Feature Structure Systems.

Currently much attention is being given to the advantages of representing the properties of entries extracted from machine-readable dictionaries using typed feature structure systems. A system of this type is now being adopted by the ACQUILEX project to map its lexical information from the project lexical databases to a multilingual, hierarchically organised, lexical knowledge base. The LKB uses a unification-based representation language which incorporates minimal default inheritance and typed feature structures. The types, which are defined manually, provide a framework for defining "templates" for the extraction of lexical information from the dictionary definitions (see Calzolari, 1991, for a description of such templates). Each different type consists of a pre-defined list of relevant features or 'feature structure'. The values to be assigned to the features are filled in (automatically or semi-automatically) at the lexical entry level. This approach to word specification provides a highly structured organization of the lexicon according to which the properties of related word types as well as the relations between word types and specific words are expressed in terms of structure sharing and inheritance. This makes it possible to avoid redefining the same information structures across lexical templates, thus reducing a great deal of redundancy in the specification of entries (see Sanfilippo, 1991, Copestake, 1991, for a more detailed description of this system). An example of a feature structure for the physical qualities of food is given in Figure 3. Further examples of this type of representation will be given in Section 5 when discussing the ways in which several dictionaries can be linked to make it possible for the lexicographer to browse over material from different sources.

```
phys-qual
Parents = top

[phys-qual
 COLOUR: colour
 SMELL: [smell
        CONST: [constituency
                UNIFORM: boolean
                SPEC: string
                CONSTITUENTS: [list_of_preds_and_degrees]]]
 TASTE: taste
 TEMPERATURE: temperature
 TEXTURE: texture
 SIZE: size
 AGEING: ageing]
```

Figure 3 Feature structure for the physical qualities of FOOD in the ACQUILEX type system

3. BROWSING FUNCTIONS

In this section, the different kinds of access functions which should be implemented by a dictionary browsing tool are described in detail.

The browser query system must provide the user with a series of functions which can be used to access the dictionary data and lookup various lexical elements or combinations of elements via any of the information categories present in the entry and identified explicitly by the data parsing procedures according to the database model. Data indexing procedures must thus take the structured data fields as input and create sets of indices on the values for all the selected attributes so that the lexical entry can be accessed on the basis of the contents of each information field, and not just the headword as in traditional dictionary lookup. In addition, for multilingual dictionary data, each data field must be 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.

Any data query language used must be easy to formulate and as close as possible to natural language; it must enable the user to formulate an unrestricted variety of queries against an LDB and it should be possible to flexibly format answers to queries, either for display or for storage and use by other procedures. The user must be able to specify conditions on the attributes of LDB entries. Only entries satisfying conditions will become part of the query answer. The user can specify which attributes of successful entries are part of the answer and what the output format will be.

Thus, the user must be able to lookup given lexical items (single word forms or all the forms of a given lemma) 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 all or a selected part of the results, define conditions to restrict his search and, if operating on bilingual or multilingual dictionaries, select the language on which the search is to operate.

3.1 User Interfaces

Queries are formulated and represented in different modes by different systems. For example, Neff at al (1987) use a Query-By-Example type syntax and representation. The query is stated as entries in the nodes of a two-dimensional representation of the LDB. Example elements, denoted by a leading underscore are used to relate values of attributes in a query tree to conditions in a condition box. An example of a query using this system is given in Figure 4a.

In the Acquilex LDB system, implemented at Cambridge, queries can be represented in two forms: textual and graphical. In both forms they can be seen as trees whose branches are the attribute names and whose leaves are the values. Queries are processed in two stages: the system first determines which values for

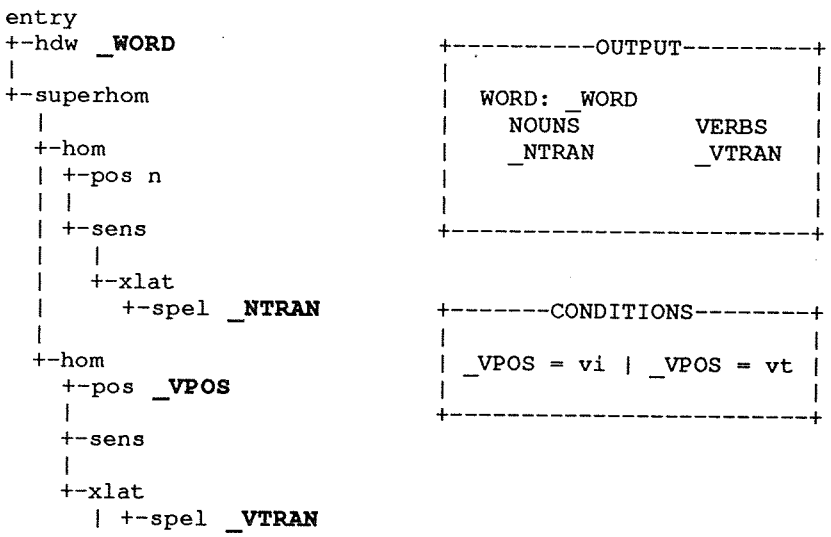


Figure 4a) Representation of a query on the Collins English-Italian dictionary. The query searches all words which are both nouns and verbs in English together with their translations. (from Neff et al, 1987)

(i)

```
[[syn
    [gcode T5a]]
[pron
[nsylls 2]
[s1
        [onset p]]]]
```

(ii)

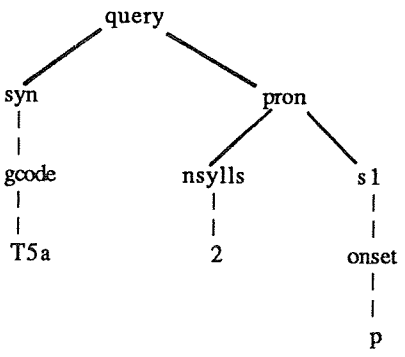


Figure 4b) Representation of a query in both (i) textual and (ii) graphical form on LDOCE. The query searches entries in which the grammar code is T5a, there are 2 syllables and the first begins with p. (from Carroll, 1990)

attributes in the query are most discriminating (on the basis of statistics acquired during the creation of the index files), and finds an initial set of entries which satisfy these values. It then reads these entries from the dictionary, checks which ones satisfy the rest of the non-discriminating attribute values, and returns the ones which do as the result. Examples of a query in this system, in both text and graphical mode, are given in Figure 4b.

The graphical interface is generally considered as the most convenient; queries are formulated in query windows by clicking a mouse button on commands which are presented as items in menus. Results are displayed in separate windows. The graphical interface is implemented on a Macintosh and thus the generic 'Cut', 'Copy' and 'Paste' clipboard commands can be used in query windows to manipulate queries and results. The text mode is more cumbersome, the user is obliged to enter much more information manually, on the keyboard, as commands are issued by typing them. He also needs a more complete knowledge of the command syntax.

In fact, nowadays, most user-friendly browsing systems are menu-driven. A query session will begin by presenting the user with a main Menu of all the commands which can be operated on the data. A context sensitive system Help is normally associated with the command menu and can be referred to at any moment during the session to enable the user to view detailed instructions on the scope and modalities of use of each command. During a query session, whenever the menu is displayed, the commands which can be activated at that particular moment should be evidenced.

3.2 Querying a Lexical Database: the MLDB

In order to give an idea of typical browsing functions in operation over lexical data, in the following section we give an example of a query session on the Pisa Multilingual Database System. The lexical components of the MLDB include the Italian Machine Dictionary - mainly based on the Zingarelli Italian Dictionary -, the Garzanti 'Nuovo Dizionario Italiano', and the Collins Concise Italian/English, English/Italian Dictionary. (See Marinai et al., 1990, for a detailed description of the design and implementation of this system.)

The MLDB is a menu-driven system in which commands are issued by selecting them from menus. A particular advantage of this system is that the user needs no prior knowledge before using it; he is guided through a query session by the command menu which is displayed at the bottom of the screen and a context sensitive Help can always be invoked. Throughout the session, the commands which can be activated at any given moment and their key letters are evidenced.

The following command types from the MLDB will be illustrated as being fundamental for any dictionary browsing system:

Search: simple and complex queries are formulated on the lexical data
View: lexical entries satisfying the search conditions are displayed in various formats
Setup: system default parameters are modified
Restriction rules: particular conditions are imposed on search and view functions

3.2.1 SEARCH

When this command is issued, the MLDB 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.

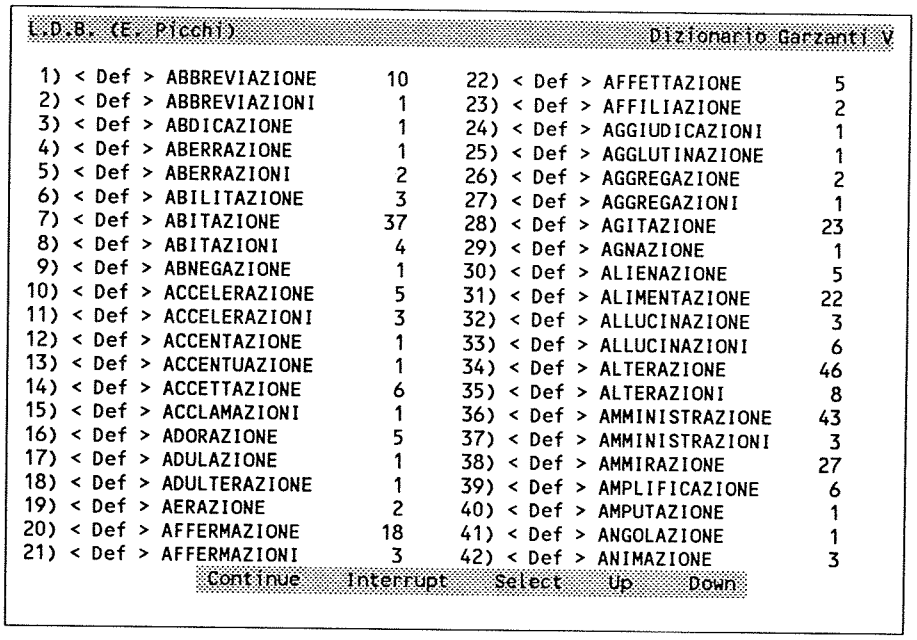
Simple and complex queries can be formulated: single lexical items, sets of lexical items with common features expressed using wildcard characters or groups of words correlated by logical operators can be searched.Searches are made on an Attribute-Value basis. This means that the item or character string 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.

The following types of simple look-up can be made:
xxx searches all items matching the item **xxx**
s* searches all items beginning with string **s**;
***s** searches all items ending with string **s**;
s searches all items containing string **s**;
(s/t) searches all items containing string **s** or string **t**;
***(s/t)** searches all items ending with string **s** or string **t**;
(s/t)* searches all items beginning with string **s** or string **t**.

Here below we show a display of the results for a search for the word "**sapore**" 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.

```
L.D.B. (E. Picchi)                                  Dizionario Garzanti V

1) <Entry> SAPORE              1
2) <  Def> SAPORE             80
3) <Examp> SAPORE              4
4) < Expl> SAPORE              5
5) <SIsyn> SAPORE              1
6) < Is_A> SAPORE             14

                    Continue              Select
```

In the next figure, we show the display of the results of a search made using wild cards in order to find all items ending with "**azione/azioni**" in the definition fields.



The results which are of interest for subsequent operations can be selected. The list of the results selected can be viewed at any time by issuing the **LIST** command from the main menu environment.

COMPLEX SEARCH FUNCTIONS

Complex search functions can be defined in which occurrences of a combination of two or more given items in the same entry are looked-up. In the MLDB system, such functions are known as "families". When the command **FAMILY** 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, OR and NOT operators: implemented as &, / 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 creates a function which searches entries in which item 1 but not 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.

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 the **LIST** function.

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. Alternatively, it can be set to specify that items searched in combination must be found in the same (i) entry, (ii) sense group, or (iii) data field.

SEARCHES USING MORPHOLOGY

The MLDB also maintains morphological procedures for Italian and English. The morphological generator can be used expand any given lemma by producing the set of all its forms; the whole paradigm for the lemma can then be searched throughout the mono- or bilingual LDBs in the system by entering a single command.

In the next figure, we show the results of a search using the morphology procedure on the Italian verb **vendere**. All the forms of the verb were generated and then searched throughout the Garzanti LDB, in the definition fields.

```
L.D.B. (E. Picchi)                          Dizionario Garzanti V

Num. Item                          Freq.
 1) < Def > VENDERE                 24
 2) < Def > VENDUTO                  6
 3) < Def > VENDUTA                  6
 4) < Def > VENDUTI                  3
 5) < Def > VENDUTE                  1
 6) < Def > VENDE                   97
 7) < Def > VENDONO                 20
 8) < Def > VENDEVA                  1
 9)*< Def > VENDENDO                 1

   Continue   Select   Up   Down
```

3.2.2 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, in 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 the **SETUP** function.

The **VIEW** command is always referred to the current item being queried (single lexical items or combinations of items). 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. The entire entry for a chosen occurrence can also be viewed in Dictionary Access Format (a format which resembles the data representation schema defined for the LDB).

Here below we show the display, in extended text format, of results for the previous query on **sapore** in which the item is found in the Definition field.

```
L.D.B. (Picchi)                                    Dizionario Garzanti V
        Item : <Def> SAPORE          Frequency : 80
1)  ABBOCCATO dolce {PoS} s.m. {NPoS} N {Def} il sapore abboccato del vino
    {Is_A} SAPORE {Examp} questo vino ha dell'
2)  ACETOSA Poligonacee {Def} pianta erbacea con foglie a saetta di sapore
    acido, fiori verdi e rossi raccolti in pannocchia.
3)  ACETOSELLA erbacea con foglie simili a quelle del trifoglio, di sapore
    acido, da cui si estrae il sale di acetosella
4)  ACIDO {NPoS} A {Pron} à- {Sense} 1 {Def} di sapore aspro, pungente.
    {SIsyn} agro {SIsyn} acre {SIsyn} brusco
5)  ACIDO {Def} sostanza solida, liquida o gassosa, di sapore pungente,
    che in soluzione libera cationi idrogeno e che
6)  ACRE {Morph} superl. {acerrimo} {Sense} 1 {Def} di sapore piccante,
    pungente. {SIsyn} aspro {SIsyn} agro {SIsyn} acido
7)  AFFUMICARE, a scopo di conservazione o per dar loro un sapore
    particolare.
8)  AGLIACEO A {Pron} -glià {Def} si dice di odore e sapore simile a
    quello dell'aglio.
9)  AGRETTO s.m. {NPoS} N { Pron} -grét {Def} sapore agro. {Is_A} SAPORE
10) AGREZZA {Def} l'essere agro {Is_A} PROPERTY {AlRel} AGRO ADJ2NOUN
    {Def} sapore agro. {Is_A} SAPORE {SIsyn} acidità {SIsyn} acredine
11) AGRO.1 {PoS} agg. {NPoS} A {Sense} 1 {Def} di sapore aspro.
    {SIsyn} acido {SIsyn} acre {Sense} 2 {Def} acerbo

          Continue          Interrupt          Select        F1 Help |
```

The entry for **agro**[1] has been selected and is displayed in dictionary access format.

```
L.D.B. (Picchi)                                    Dizionario Garzanti V

     AGRO.1*

{Entry} agro.1*
{  PoS} agg.
{ NPoS} A
{Sense} 1
{  Def} di sapore aspro.
{SIsyn} acido
{SIsyn} acre
{Sense} 2
{  Def} acerbo
{Examp} quest'arancia è agra
{Sense} 3
{SI fl} {fig.}
{  Def} pungente, severo, malevolo
{Examp} le sue parole furono agre
{  PoS} s.m.
{ NPoS} N
{Sense} 1
{  Def} sapore agro

                  Continue                    Interrupt
```

Here below, we show a family which has been created using a complex search function. The family can be used to find definitions in which either the word **strumento** or the word **attrezzo** (instrument, tool or implement) is found together with different forms (inflected and derived) of the Italian lemma **misurare** (to measure). Complex search functions of this type are very useful in order to search dictionaries for all headwords which are defined in a particular way. Here we are looking for all types of tools which are used for some sort of measuring purpose.

```
L.D.B. (Picchi)                                    Dizionario Garzanti V


Family n. 1 : ( <Def>STRUMENTO ¦ <Def>ATTREZZO )  &  ( <Def>MISURA ¦
                <Def>MISURARE ¦ <Def>MISURABILI ¦ <Def>MISURATORE ¦
                <Def>MISURAVA ¦ <Def>MISURAZIONE ¦ <Def>MISURE )
```

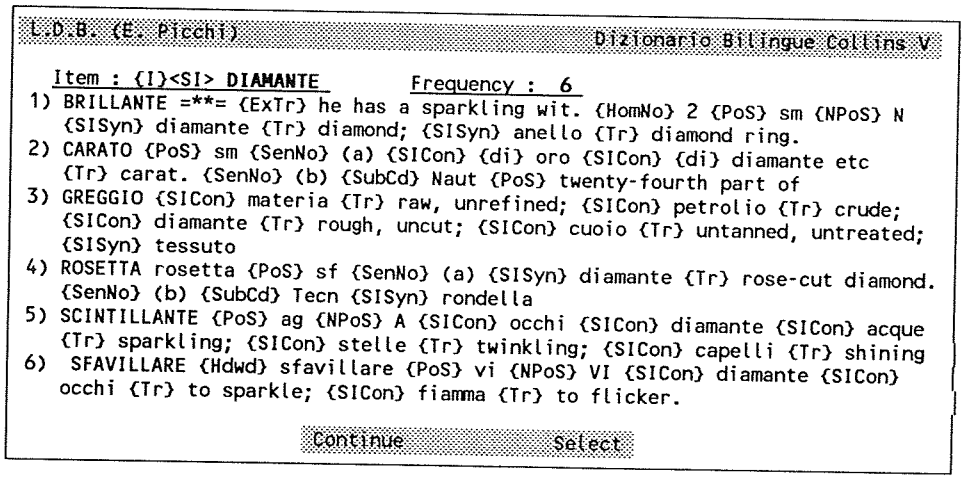
This figure shows a display of the first screenful of results obtained applying the previously defined search function.

```
L.D.B. (Picchi)                                    Dizionario Garzanti V

1)  ACETIMETRO s.m. {Def} strumento per misurare il grado di concentrazione
    acida dell'aceto.
2)  AEROMETRO {Def} strumento atto a misurare la densità dell'aria.
3)  ALCOLOMETRO s.m. {Def} strumento che misura la quantità di alcool
    contenuta in un liquido
4)  AMPEROMETRO s.m. {Def} strumento per misurare l'intensità di una
    corrente elettrica.
5)  ANEMOMETRO s.m. {Def} strumento che misura la velocità del vento.
6)  BAROMETRO s.m. {Def} strumento che serve a misurare la pressione
    atmosferica.
7)  CALIBRO {Def} strumento di forme diverse, per misurare lo spessore di
    lamiere
8)  CALORIMETRO s.m. {Def} strumento destinato alla misura della quantità
    di calore dei corpi.
9)  CLINOMETRO s.m. {Def} strumento usato in marina e in topografia per
    misurare l'inclinazione
10) CONTAGOCCE invar. {Def} piccolo strumento usato per misurare a gocce
    liquidi, spec. medicinali
11) CURVIMETRO s.m. {Def} strumento per misurare la lunghezza di un arco
    di curva.

    Continue        Interrupt        Select        F1 Help
```

Here we see the results of a search on the Collins English/Italian bilingual database. The items **diamond/diamante** were searched throughout the LDB in all fields.

```
L.D.B. (E. Picchi)                          Dizionario Bilingue Collins V
Num. Item                          Freq.
 1) {I}< Hdwd> DIAMANTE             1
 2) {I}<   SI> DIAMANTE             6
 3) {I}<Trans> DIAMANTE             1
 4) {I}< ExTr> DIAMANTE             2
 5) {E}< Hdwd> DIAMOND              2
 6) {E}<   SI> DIAMOND              3
 7) {E}<Trans> DIAMOND              5
 8) {E}<   Ex> DIAMOND              2
 9)*{E}< ExTr> DIAMOND              3

          Continue   Select   Display Family   Up   Down
```

Here we have the display of the results when **diamante** was found in the Semantic Indicator field. It can be seen that in two cases, for **brillante** and **rosetta**, as a result of previous analyses on the data, the semantic indicator **diamante** is indicated as a synonym of the headword.



3.2.3 SETUP

A number of the query system's default parameters can be modified by the user.
OUTPUT: the output device can be selected; the user can choose to display his selected results on the screen, print them or store them in a file on disk.
ENTRY DISPLAY: the parameters which govern the length of the sections of the entries to be displayed when results are viewed can also be changed.
COOCCURRENCE: The domain over which complex search functions are to apply can be set, i.e. the area in which items combined in the function are to be searched; the user can choose between applying the complex function over the entire entry, over separate senses, or single fields
LANGUAGE: for bilingual or multilingual lexicons, the MLDB system automatically indexes each part of the entry according to the appropriate language; the user can then select the language on which the search functions must operate.
FIELD-SEARCH: the particular fields on which the **SEARCH** function is to operate can be selected.
FIELD-VIEW: similar to the above command, the fields on which the **VIEW** function is to operate can be selected.

3.2.4 RESTRICTION-RULES

This function permits the setting of restriction rules on any view function used in the system, for instance a restriction can be imposed on the POS of the entries satisfying the search conditions, thus permitting only entries with this grammatical category to be viewed.

The restriction rules can consist of several conditions linked using AND, OR and NOT operators: &, / and \

A restriction condition is formed using the following elements: 1) logic operator; 2) Code for the Field for which the restriction conditions must apply; 3) type of condition (**B** beginning with, **E** ending with, **S** containing the string, **M** mask, = equal to, \ not); 4) any string to be used for the value condition.

In the following figure we show an example in which the item **fare** (to do) was searched throughout the definition fields of the Grazanti LDB and the results are displayed only for those entries with verb as the grammatical category. In order to impose this restriction the normalized POS field was used so that all verb subcategorizations could be identified with one command, however they had been indicated in the source dictionary.

```
L.D.B. (E. Picchi)                                    Dizionario Garzanti V
  Item : < Def > FARE                Frequency :  570      *Restr.Active*
1)  ABBONARE.1* {Entry} abbonare.1* {PoS} v.tr. {NPoS} VT {Morph} {io abbòno
    ecc.} {Def} fare un abbonamento per conto di qlcu. {SubEn} ABBONARE.1*
    -arsi {PoS} v.rifl.
2)  ABBORRACCIARE {Entry} abborracciare {PoS} v.tr. {NPoS} VT {Morph} {io
    abborràccio ecc.} {Def} fare qlco. male, in fretta {Examp} * un discorso
3)  ABBOZZARE.1* {PoS} v.tr. {NPoS} VT {Morph} {io abbòzzo ecc.} {Sense} 1
    {Def} fare un abbozzo {Examp} * un disegno {Examp} * un racconto
4)  ACCAREZZARE {Entry} accarezzare {PoS} v.tr. {NPoS} VT {Morph} {io
    accarézzo ecc.} {Sense} 1 {Def} fare carezze, lisciare con la mano,
    spesso in segno d'affetto {Examp} * un cane
5)  ACCATASTARE {Entry} accatastare {PoS} v.tr. {NPoS} VT {Sense} 1 {Def}
    fare una catasta {Sense} 2 {SI fl} anche {fig.} {Def} disporre
    disordinatamente le cose una sull'altra {Examp} * libri
6)  ACCENNARE {Entry} accennare {PoS} v.intr. {NPoS} VI {Morph} {io accénno
    ecc.} {Sense} 1 {Def} fare un cenno, indicare con cenni {Examp} * di sì
    col capo {Examp} mi accennò di seguirlo
7)  ACCIABATTARE {Entry} acciabattare {PoS} v.tr. {NPoS} VT {SI fl} {rar.}
    {Def} fare un lavoro in fretta e con trascuratezza.
8)  ACCINGERSI io mi accinsi}, {tu ti accingésti ecc.}; p.pass. {accinto}
    {Def} prepararsi a fare una cosa {Examp} si accingeva a parlare
                  Continue      Interrupt      Select
```

4. FROM DICTIONARY TO TEXT AND LANGUAGE CORPORA

Ideally, a dictionary browsing system should also offer the user facilities which permit him to access and query texts and language reference corpora directly from the dictionary environment. This is important for the lexicographer employed in the compilation of dictionary entries so that he can access and retrieve material from the corpus in order to be able to analyze the behaviour of the particular term on which he is working, obtain indications on the best way to represent it in the dictionary entry, and extract examples directly. It should be possible to use all the browsing functions implemented in an LDB system to search groups of words throughout a

reference corpus. For example, the inflectional morphological procedure can be invoked to search all words in the corpus belonging to the same lexical paradigm. In particular, complex searches can be made on a corpus using information which has been derived from the dictionary data, e.g data on taxonomic and other semantic relations that has been extracted by procedures which analyze the definitions can be used to search groups of semantically related words throughout a set of texts.

5. BROWSING OVER MORE THAN ONE DICTIONARY

An important requirement of the lexicographer consulting dictionary data is to be able to access and query data not just in a single LDB but over different LDB systems. A dictionary browsing system should thus include functions to link and compare lexical data from different sources. Clearly, the source dictionaries must be structured using compatible data models; the main problem to be solved is that links must be created at the sense and not the headword level.

A procedure which has been studied to permit semi-automatic mapping between the monolingual and bilingual LDBs contained in the Pisa integrated MLDB system is described in (Marinai et al., 1991). Equivalent entries from separate electronic dictionaries are combined and links are created between their senses, mainly on the basis of information which can be extracted from definitions, examples and any semantic labels, in order to create a new more complete composite entry which represents the sum of the information contained in the individual dictionaries. The lexicographer can use the system to compare and study the lexical information given for the same item in different sources. This system has been experimented on two Italian monolingual LDBs and an Italian/English bilingual LDB. Figure 5 shows the sense linking proposed by the procedure of the information contained in the three source dictionaries for the Italian lemma "diamante". The composite entries created can go to form part of a new merged LDB on which all the query and analysis functions of the MLDB system can be used.

Another way of comparing information derived from entries of different dictionaries is offered by the typed feature structure system which has been implemented in the ACQUILEX Lexical Knowledge Base. Data from several monolingual dictionaries for four languages (English, Dutch, Italian and Spanish) are stored in the same type system. Lexical entries from different source LKBs adopting a common type mechanism can be compared using a general feature structure matching utility. In a monolingual environment, this routine can be performed to implement classifications and cross-classifications of lexical entries into taxonomies, for example. In a multilingual environment, the relationship between word links is represented by tlinks (for translation links). In general, there may be a many-many equivalence between word senses, but each possibility is represented by a single tlink. A tlink is a structure which states that two feature structures are to be regarded as translation equivalents. The tlink is stored and indexed in such a way as to facilitate finding translation equivalents of particular words or word senses. It contains paths which should be instantiated by complete

```
      Hdwd DIAMANTE
      NPoS N
     Sense 1
G      Def forma cristallina del carbonio, durissima, trasparente, incolora
G      Def prezioso per la sua rarità, è usato in gioielleria e come abrasivo
           di massima durezza
G    Idiom nozze di ▪
G     Expl il sessantesimo anniversario di matrimonio
D      Def CARBONIO CRISTALLIZZATO DURISSIMO TRASPARENTE
C    SLCOL gen
C    Trans diamond;
C       Ex di ▪, di ▪i
C     ExTr diamond {attr.}
     Sense 2
G      Def strumento con punta di diamante usato per tagliare il vetro
D      Def ARNESE CON PUNTA DI DIAMANTE PER TAGLIARE IL VETRO
     Sense 3
G      Def punto del fuso di un'ancora dal quale si diramano i due bracci
           uncinati ({marre})
D      Def ESTREMITA' DEL FUSO DELL' ANCORA A CEPPO
D    SubCd 4Z
C    SubCd Naut
C    SIcon {di} ancora
C    Trans crown.
     Sense 4
GD     Def carattere tipografico molto minuto
G    Idiom edizioni ▪
G     Expl di formato molto piccolo, composte con i suddetti caratteri.
     Sense 5
      Def TRACCIATO INTERNO DEL CAMPO DI GIOCO NEL BASE-BALL
     Sense 6
D      Def FOSSO STRETTO SCAVATO NEL FOSSATO DI ANTICHE FORTIFICAZIONI
D    SubCd 4G
     Sense 7
D      Def MONETA D' ARGENTO FERRARESE COL DIAMANTE SUL ROVESCIO
D    SubCd 5P
```

G = Garzanti dictionary data
D = Italian Machine Dictionary (DMI) data
C = Collins English/Italian dictionary data

Figure 5 MLDB system proposal for sense mapping between the DMI, Garzanti and Collins LDBs for the Italian lemma "diamante". (from Marinai et al., 1990)

```
(TEACHER 1)
[simple-tlink
FSO: [rule
     0: <0> = [lex-noun-sign
              ORTH: teacher
              CAT: noun-cat
              SEM: [unary-formula-entity-arg1
                    IND: <1> = entity
                    PRED: <2> = teacher_1
                    ARG1: <1>]
              SENSE-ID: sense-id
              RQS: [human
              ORIGIN-AREA: string
              TELIC: verb-sem
              PHYSICAL: true
              OBJECT-INDEX: dummy-or-obj
              ORIGIN: (string basic)
              ANIMACY: boolean
              PHYSICAL_STATE: solid_a
              QUAL: phys_qual
              QUANT: quantity
              SIMILAR: string
              APPEARANCE: appearance
              FORM: physform
              CONSTITUENCY: constutuency
              AGE: scalar
              SEX: male
              WORK-FUNCTION: verb-sem]]
     1: <0>]
FS1: [rule
     0: <3> = [lex-noun-sign
              ORTH: maestro
              CAT: noun-cat
              SEM:     [unary-formula-entity-arg1
                    IND: <1>
                    PRED: <4> = maestro_1
                    ARG1: <1>]
              SENSE-ID: sense-id
              RQS: [human
              ORIGIN-AREA: string
              TELIC: verb-sem
              PHYSICAL: true
              OBJECT-INDEX: dummy-or-obj
              ORIGIN: (string basic)
              ANIMACY: boolean
              PHYSICAL_STATE: solid_a
              QUAL: phys_qual
              QUANT: quantity
              SIMILAR: string
              APPEARANCE: appearance
              FORM: physform
              CONSTITUENCY: constutuency
              AGE: scalar
              SEX: male
              WORK-FUNCTION: verb-sem]]
```

Figure 6 Simple tlink link encoding restriction that male teacher in English is translated as maestro in Spanish
(from Copestake and Jones, 1991)

lexical entries in the two languages being linked and paths which will contain equivalent feature structures when this is done. The matching program will compare feature structures and return the sense or senses which are most similar to the source sense. The implementation of the tlink mechanism in the LKB is described in detail in Copestake and Jones (1991). Figure 6 gives an example of the tlink mechanism used to link the feature structures derived from the Vox Spanish monolingual dictionary and LDOCE for 'maestro' and 'teacher', respectively.

6. RECOMMENDATIONS

In conclusion we make the following recommendations:

A Dictionary Browsing Tool for the linguist or lexicographer should be implemented in a user-friendly fashion which implies:
- adopting easy-to-formulate Data Query Languages;
- using command menus to guide the user throughout a query session;
- including a context-sensitive system Help, accessible at all times during the query session.

The Browser should include functions to perform the following operations:
- Lookup on single lexical items;
- Lookup on items containing one (or more) strings of characters;
- Lookup using a morphological generator;
- Definition of complex search functions;
- Lookup using pre-defined search functions;
- Display of results for the item(s) searched in different modes;
- Display and change of the system default parameters;
- Select and change language on which searches operate (for multilingual systems);
- Display of list of all items already searched;
- Recall of already searched items;
- Display of the currently active item

Ideally, a tool of this type will also be able to:
- permit browsing over more than one dictionary using the same query language;
- offer direct access to text and language reference corpora;
- allow the results of a query on the lexical data to be used when performing a search on text data.

REFERENCES

Amsler R., Tompa F. (1988), An SGML-based standard for English monolingual dictionaries, in *Proceedings of the Fourth Annual Conference of the UW Center for the New OED: Information in Text*, Waterloo, 61-79.

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.

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

Calzolari N. (1991), Acquiring and Representing Semantic Information in a Lexical Knowledge Base, *Proceedings of the ACL SIGLEX Workshop on Lexical Semantic and Knowledge Representation*, Berkeley, California, pp.188-196.

Calzolari N., Peters C., Roventini A. (1990), Computational Model of the Dictionary Entry: Preliminary Report, ACQUILEX, Esprit BRA 3030, Six Month Deliverable, ILC-ACQ-1-90, Pisa, 90p.

Calzolari N., Picchi E. (1988), Acquisition of Semantic Information from an On-Line Dictionary, Proceedings of the 12th International Conference on Computational Linguistics, Budapest, 1988, 87-92.

Carroll J. (1990), Lexical Database System: User Manual, ACQUILEX, Esprit BRA 3030,Deliverable No. 2.3.3 (a).

Copestake A. (1991), The LKB: A System for Representing Lexical Information extracted from Machine-Readable Dictionaries, *Proceedings of the Acquilex Workshop on Default Inheritance in the Lexicon,* Cambridge.

Copestake A. Jones B. (1991), Support for multi-lingual lexicons in the LKB system, (Draft version, August 1991), Esprit BRA 3030.

Fought J., Van Ess-Dykema C. (1990), Toward an SGML Document Type Definition for Bilingual Dictionaries, TEI Internal Report.

Ide N., Veronis J., Warwick-Armstrong S., Calzolari N. (1991), Principles for Encoding Machine Readable Dictionaries, TEI WP, A15W6.

Marinai E., Peters C., Picchi E. (1990), The Pisa Multilingual Lexical Database System, Esprit BRA 3030, Twelve Month Deliverable, ILC-ACQ-2-90, Pisa, 61p.

Marinai E., Peters C., Picchi, E. (1991), A prototype system for the semi-automatic sense linking and merging of mono- and bilingual LDBs, in N.Ide and S. Hockey (eds.), *Research in Humanities Computing*, OUP, (forthcoming).

Neff M., Byrd R., Rizk O. (1987), Creating and querying hierarchical lexical data bases, in *Proceedings of the Second ACL Conference on Applied Natural Language Processing*, Austin, Texas, 84-93.

Picchi E. (1991), D.B.T.: A Textual Data Base System, in L. Cignoni and C. Peters (eds.), *Computational Lexicology and Lexicography. Special Issue dedicated to Bernard Quemada. I., Linguistica Computazionale,* Vol VII, 177-205.

Sanfilippo A. (1991), LKB Encoding of Lexical Knowledge from Machine-Readable Dictionaries, *Proceedings of the Acquilex Workshop on Default Inheritance in the Lexicon,* Cambridge.

TEI (1990), Text Encoding Initiative: Guidelines. Draft Version 1.0, Technical Report TEI P1, ACH, ACL, ALLC.

Dictionaries Cited

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.

Vox: Diccionario General Ilustrado de la Lengua Española, Bibliograf S.A., 1987.

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