4 The contribution of computational lexicography

BRANIMIR K. BOGURAEV

4.1 Introduction

This chapter presents an operational definition of *computational lexicography*, which is emerging as a discipline in its own right. In the context of one of its primary goals – facilitation of (semi-)automatic construction of lexical knowledge bases (aka computational lexicons) by extracting lexical data from on-line dictionaries – the concerns of dictionary analysis are related to those of lexical semantics. The chapter argues for a particular paradigm of lexicon construction, which relies crucially on having flexible access to fine-grained structural analyses of multiple dictionary sources. To this end, several related issues in computational lexicography are discussed in some detail.

In particular, the notion of structured dictionary representation is exemplified by looking at the wide range of functions encoded, both explicitly and implicitly, in the notations for dictionary entries. This allows the formulation of a framework for exploiting the lexical content of dictionary structure, in part encoded configurationally, for the purpose of streamlining the process of lexical acquisition.

A methodology for populating a lexical knowledge base with knowledge derived from existing lexical resources should not be in isolation from a theory of lexical semantics. Rather than promote any particular theory, however, we argue that without a theoretical framework the traditional methods of computational lexicography can hardly go further than highlighting the inadequacies of current dictionaries. We further argue that by reference to a theory that assumes a formal and rich model of the lexicon, dictionaries can be made to reveal – through guided analysis of highly structured isomorphs – a number of lexical semantic relations of relevance to natural language processing, which are only encoded implicitly and are distributed across the entire source.

This paper was originally presented at a *Symposium on Natural Language – Language and Action in the World,* held in December 1989 at Bolt, Beranek and Neumann Laboratories, Cambridge, Mass. Some preliminary results were reported at the First International Workshop on Lexical Acquisition, held during the 11th International Conference on Artificial Intelligence in Detroit. I have benefited greatly from discussions with Sue Atkins, Ted Briscoe, Beth Levin, and James Pustejovsky. Most of the results in this chapter were obtained from the machine-readable versions of the *Longman Dictionary of Contemporary English* and the *Collins Thesaurus:* thanks are due to the publishers for granting access to the sources for research purposes.

One approach to scaling up the lexical components of natural language systems prototypes to enable them to handle realistic texts has been to turn to existing machine-readable forms of published dictionaries. On the assumption that they not only represent (trivially) a convenient source of words, but also contain (in a less obvious, and more interesting way) a significant amount of lexical data, recent research efforts have shown that automated procedures can be developed for extracting and formalizing explicitly available, as well as implicitly encoded, information – phonological, syntactic, and semantic – from machine-readable dictionaries (MRDs).

The appeal of using on-line dictionaries in the construction of formal computational lexicons is intuitively obvious: dictionaries contain information about words, and lexicons need such information. If automated procedures could be developed for extracting and formalizing lexical data, on a large scale, from existing on-line resources, natural language processing (NLP) systems would have ways of capitalizing on much of the lexicographic effort embodied in the production of reference materials for human consumption. On the other hand, there are at least two classes of disadvantages to the use of MRDs in natural language processing. First, because these are produced with the human user in mind, there is a strong assumption about the nature of understanding and interpretation required to make use of a dictionary entry; second, due to the very nature of the process of (human) lexicography, present-day dictionaries are far from complete, consistent, and coherent, certainly with respect to virtually any of the numerous kinds of lexical data they choose to represent and encode. An important question then becomes: where is the line between useful and relevant data to be extracted from existing machine-readable sources, on the one hand, and the level of 'noise' (inconsistencies, mis-representations, omissions) inherent in such sources and detrimental to the enterprise of deriving computational lexicons by (semi-)automatic means, on the other?

A number of arguments have been put forward in support of a claim that, in effect, a dictionary is only as good as its worst (or least experienced) lexicographer – and by that token, it is not much good for developing systematic procedures for extraction of lexical data. For instance, in the process of giving a descriptive introduction to the discipline of *computational lexicography*,[1] Atkins (1990) not only summarizes the process of building a large-scale lexicon[2] as

[1]There is still no widely accepted term covering the kinds of activities discussed here. A common practice is to use *computational lexicology*. In recognition of the fact that the ultimate goal of this, and related, research is to produce dictionaries – albeit by means different from the traditional ones (computer-based semi-automatic analysis of existing human dictionaries) and intended for a different kind of 'user' (natural language processing programs) – we prefer *computational lexicography*.

[2]In the rest of this paper, *dictionary* is going to be systematically used to refer to a (published) dictionary, or its machine-readable equivalent, compiled by humans for human use. In contrast, in order to emphasize the (semi-)automatic nature of compiling a formal repository of lexical data for use by a computer program for any natural language processing task, we call such a structure a *lexicon* (or *computational lexicon*).

"trawling" a machine-readable dictionary in search for lexical facts, but points out an imbalance between the kinds of syntactic and semantic information that can be identified by "minutely examining" existing dictionaries: "the useful semantic information which may be extracted at present is more restricted in scope, and virtually limited to the construction of semantic taxonomies".

Although we agree with Atkins' assessment of the state of the field, we ascribe this to the predominant paradigm of computational lexicography. More specifically, several factors are instrumental to the relative inadequacy of the semantic information derived from dictionaries.

First, from the perspective of building formal systems capable of processing natural language texts, there is (currently) a much better understanding of the nature of the syntactic information required for implementing such systems than of its semantic counterpart. In other words, the state of the art of (applied) computational linguistics is such that syntactic analyzers are much better understood than semantic interpreters; consequently, there is a fairly concrete notion of what would constitute necessary, useful, and formalizable syntactic information of general linguistic nature. Consequently, given the well-defined lexical requirements at syntactic level, there is that much more leverage in searching for (and finding) specific data to populate a lexicon at the syntactic level (see, for instance, Boguraev and Briscoe, 1989, for an elaboration of this point).

Second, most of the investigations aimed at recovery of lexical data from dictionaries fall in the category of 'localist' approaches. The notion is that if our goal is to construct an entry for a given word, then all (and the only) relevant information as far as the lexical properties of this word are concerned is to be found, locally, in the source dictionary entry for that word. This observation explains why constructing taxonomic networks on the basis of the general genus-differentiae model of dictionary definitions (as exemplified by the work of e.g., Amsler, 1981; Calzolari, 1984; and Alshawi, 1989) is essentially the extent to which identification of semantic information has been developed. It also underlies the pessimism (expressed by, e.g., Atkins, 1990) concerning the useful semantic information extractable from a dictionary. Most dictionary entries are, indeed, impoverished when viewed in isolation; therefore, the lexical structures derived from them would be similarly under-representative.

Third, it is important to take into account the relationship between the expressive power of on-line dictionary models and the scope of lexical information available via the access methods such models support. In particular, mounting a dictionary on-line only partially (as when leaving out certain fields and segments of entries) and/or ignoring components of an entry whose function is apparently only of typographical or aesthetic nature (such as typesetter control codes) tends to impose certain limitations on the kinds of lexical relationships that can be observed and recovered from a dictionary. Although, in principle, computational lexicography is concerned not only with developing techniques and methods for extraction of lexical data but also with building tools for making lexical resources

available to such techniques and methods, in reality often the on-line dictionary model is not an adequate representation of lexical information on a large scale. (Boguraev et al., 1990a, discuss this issue at some length.)

Finally, there is an alternative view emerging concerning a more 'realistic' definition of computational lexicography. Hoping to derive, by fully automatic means, a computational lexicon – from one, or several, dictionary sources – is overly optimistic, and provably unrealistic. On the other hand, discarding the potential utility of such sources on the grounds that they have not yielded enough consistent and comprehensive information is unduly pessimistic. Between these two extremes there is an opinion that the potential of on-line dictionaries is in using them to facilitate and assist in the construction of large-scale lexicons (see, for instance, Levin, this volume). The image is not that of 'cranking the handle' and getting a lexicon overnight, but that of carefully designing a lexicon and then, for each aspect of lexical data deemed to be relevant for (semantic) processing of language, using the dictionary sources – *in their entirety* – to find instances of, and evidence for, such data. This paradigm relies on directed search for a number of specific lexical properties, and requires a much stronger notion of a theory of lexical semantics than assumed by computational lexicography to date.

The remainder of this chapter addresses these issues in some detail. Section 4.2 presents the highlights of a particular model for an on-line dictionary, which promotes fine-grained analysis as an important prerequisite for fully exploiting the semantic content of dictionaries. Section 4.3 introduces the concept of distributed lexical knowledge and demonstrates the relationship between configurational patterns occurring regularly across the entire dictionary source, and lexical semantic relations that underlie – and hence can be recovered by exploiting – these patterns. Section 4.4 discusses the importance of lexical semantic theories. The emphasis, however, is not on promoting a particular theory; rather, we show how the model of lexical data extraction developed in the preceding sections can be put to use to populate the semantic component of lexical entries as stipulated by the theory.

Overall, the chapter argues that just as in the case of building a syntactic lexicon from a machine-readable source, there is far more in a dictionary than meets the eye; however, this wealth of information typically cannot be observed, nor extracted, without reference to a formal linguistic theory with very precise lexical requirements, and without a set of tools capable of making very explicit the highly compacted and codified information at source.

4.2 Structure and analysis of machine-readable dictionaries

Prior to seeking interesting and meaningful generalizations concerning lexical information, repositories of such information – and more specifically, machine-readable dictionaries – should be suitably analyzed and converted to lexical

databases (LDBs). We use the term "lexical database" to refer to a highly structured isomorph of a published dictionary, which, by virtue of having both its *data* and *structure* made fully explicit, lends itself to flexible querying. One of the arguments in this chapter is that only such a general scheme for dictionary utilization would make it possible to make maximal use of the information contained in an MRD.

4.2.1 Machine-readable dictionaries and lexical databases

Dictionary sources are typically made available in the form of publishers' typesetting tapes. A tape carries a flat character stream where lexical data proper is heavily interspersed with special (control) characters. The particular denotation of typesetter control characters as font changes and other notational conventions used in the printed form of the dictionary is typically highly idiosyncratic and usually regarded as 'noise' when it comes to mounting a typesetting tape on-line for the purposes of computational lexicography.

None of the lexical database creation efforts to date addresses, explicitly, the question of fully utilizing the structural information in a dictionary, encoded in the control characters at source. Consequently, little attention has been paid to developing a general framework for processing the wide range of dictionary resources available in machine-readable form.

In situations where the conversion of an MRD into an LDB is carried out by a 'one-off' program (such as, for instance, described by Alshawi et al., 1989 and Vossen et al., 1989 in Boguraev and Briscoe, 1989), typesetter information is treated mostly as 'noise' and consequently discarded. More modular (and, by design, customizable) MRD-to-LDB conversion schemes consisting of a parser and a grammar – best exemplified by Kazman's (1986) analysis of the *Oxford English Dictionary* (OED) – appear to retain this information; however, they assign only minimal interpretation to the 'semantics' of control codes. As a result, such efforts so far have not delivered the structurally rich and explicit LDB ideally required for easy and unconstrained access to the source data, as they have been driven by processing demands of a different nature from ours.[3]

The majority of computational lexicography projects to date fall in the first of the above categories, in that they typically concentrate on the conversion of a single dictionary into an LDB. Even work based on more than one dictionary (e.g., in bilingual context: see Calzolari and Picchi, 1986) tends to use spe-

[3]The computerization of the OED had, as its primary goal, setting up a dictionary database to be used by lexicographers in the process of (re)compiling a dictionary for human, and human only, use. As a particular consequence, mapping from database representation to visual form of dictionary entries was a central concern of the design; so was efficiency in access. Another consequence of the same design was a highly idiosyncratic query language, making the kind of structure analysis discussed below difficult and unintuitive (see Neff and Boguraev, 1989, 1990; and Boguraev et al., 1989, for more details).

cialized programs for each dictionary source. In addition, not an uncommon property of existing LDBs is their completeness with respect to the original source: there is a tendency to extract, in a pre-processing phase, only some fragments (e.g., part of speech information or definition fields) while ignoring others (e.g., etymology, pronunciation, or usage notes).

This reflects a particular paradigm for deriving computational lexicons from MRDs: on the assumption that only a limited number of fields in a dictionary entry are relevant to the contents of the target lexicon, these fields are extracted by arbitrary means; the original source is then discarded, and with it the lexical relationships implicit in the overall structure of an entry are lost. Such a strategy may be justified in some cases; in particular, it saves time and effort when a very precise notion exists of what information is sought from a dictionary *and* from where and how this is to be identified and extracted. In the general case, however, when a dictionary is to be regarded as a representative 'snapshot' of a language, containing a substantial amount of explicit and implicit information about words, selective analysis and partial load inevitably loses information. Although this process of 'pre-locating' lexical data in the complete raw source is occasionally referred to as "parsing" a typesetting tape, it is substantially different from the use of the same term below, where a parser is essentially a convertor of the flat character stream into an arbitrarily complex structured representation, and parsing is both constrained never to discard any of the source content, and augmented with interpretations of the typesetter control codes in context.

Partial LDBs may be justified by the narrower, short-term requirements of specific projects; however, they are ultimately incapable of offering insights into the complex nature of lexical relations. The same is true of computerized dictionaries, which are available on-line, but only via a very limited, narrow bandwidth interface (typically allowing access exclusively by orthography). Even a functionally complete system for accessing an analyzed dictionary rapidly becomes unintuitive and cumbersome, if it is not based on fine-grained structural analysis of the source. For instance, the query processor designed to interact with the fully parsed version of the OED (Gonnet, 1987; Raymond and Blake, 1987) and capable of supporting a fairly comprehensive set of lexical queries, still faces problems of formulation and expressive power when it comes to asking questions concerning complex structural relationships between fields and components of dictionary entries. We argue this point in some detail in the next section.

4.2.2 Parsing dictionaries into LDBs

An example of the functionality require for converting to a common LDB format a range of MRDs exhibiting a range of phenomena, is provided by the general mechanism embodied in the design of a Dictionary Entry Parser (DEP). A specific implementation, described in detail by Neff and Boguraev (1989, 1990) has been applied to the analysis of several different dictionaries.

DEP functions as a stand-alone parsing engine, capable of interpreting a dictionary tape character stream with respect to a grammar of that particular dictionary, and building an explicit parse tree of the entries in the MRD. In particular, rather than just tagging the data in the dictionary to indicate its structural characteristics, the grammar explicitly controls the construction of rather elaborate tree representations denoting deeper configurational relationships between individual records and fields within an entry. Two processes are crucial for 'unfolding', or making explicit, the structure of an MRD: identification of the structural markers, and their interpretation in context resulting in detailed parse trees for entries.

Neff and Boguraev (1989, 1990) present at some length a detailed motivation for the overall system architecture, give considerations leading to the design of a dictionary grammar formalism, and discuss analyses of typical dictionary configurations across a range of different MRD sources within the DEP framework. Here we only illustrate the kind of structure assignment carried out by one of our grammars to a (fragment) of a sample dictionary entry (this, and the majority of the examples in the rest of the paper, are taken from the *Longman Dictionary of Contemporary English* – see Procter, 1978).

> **book**[1] / . . . / *n* **1** a collection of sheets of paper fastened together as a thing to be read, or to be written in . . . **3** the words of a light musical play: *Oscar Hammerstein II wrote the book of "Oklahoma", and Richard Rodgers wrote the music* – compare LIBRETTO . . .

```
entry
+ −hdw: book
|
+ −homograph
   + −print_form: book
   + −hom_number: 1
   + −syncat: n
   |
   + −pronunciation
   |   + −primary
   |       + −pron_string: bUk
   |
   + −sense_def
   |   + −sense_no: 1
   |   + −defn
   |       + −def_string: a collection of sheets of paper
   |                      fastened together as a thing to
   |                      be read, or to be written in
   + − . . . . . .
   |
   + −sense_def
   |   + −sense_no: 3
```

```
|   + −defn
|   |  + −def_string: the words of a light musical play
|   |
|   + −example
|   |  + −ex_string:  Oscar Hammerstein II wrote the
|   |                 book of "Oklahoma", and Richard
|   |                 Rodgers wrote the music
|   |
|   + −explicit_xref
|     + −how: compare
|     |
|     + −implicit_xref
|     |  + −to: libretto
|     |
|     + −exrf_string: libretto
|
. . . . . .
```

In this representation, lexical data are encoded as a set of values associated with terminal nodes in the entry parse tree. In addition to the usual observations relating to structured representations of this kind, there are two important points to be made for dictionary entry parsing in particular.

First, it is not *only* the set of terminal values that fully represents the complete lexical content of the original entry, even though by a process conceptually equivalent to 'tree walking' we would be able to recover virtually everything that is presented visually at source. On the contrary: the global picture of the structure of a dictionary entry, only intuitively inducible from the typographic properties of a dictionary, now becomes visibly marked by the system of embedded labels.

Furthermore, this embedding of labels is systematic (it is ultimately defined by the particular grammar used by DEP to assign structure to the dictionary) and maps to the notion of a *path* – namely, a step-wise refinement of the function performed by a particular terminal value.

Take, for instance, the string "LIBRETTO" in the entry for "book" above. On the face of it, the change of typeface to small capitals indicates the use of a word outside of the controlled core vocabulary employed by the lexicographers during dictionary compilation. This is the minimal analysis that might be assigned to the particular font-controlled character, and carried over to the dictionary representation. At a deeper level of interpretation, however, is the realization that the typographical convention here is used to signal an (implicit) cross-reference to another entry. In terms of representation, we are faced with several possibilities. Discarding the 'noisy' typesetter control might result in a data structure that represents the fact that "libretto" is an (explicit) cross-reference of the straightforward *see* category (as opposed to, for instance, *compare, opposite,* or *see*

picture at). Alternatively, we might follow the minimal analysis and retain a trace of the font change:

. . . [begin[small_caps]]libretto[end[small_caps]] . . .

Both of these representations are clearly impoverished; of special interest here, however, is the fact that an alternative

. . . implicit_xref = "libretto"

is equally lacking: it fails to capture the fact that the string in question is an implicit cross-reference within an explicit cross-reference applying to the third sense definition of the first homograph for "book". This knowledge is encoded in the path from the root of the entry tree to this particular terminal node:

```
LDOCE:
  entry
    .homograph
      .sense_def
        .explicit_xref
          .implicit_xref
            .to: "libretto" ;
```

It is this notion of context-driven decoding of a simple font change code, such as small_caps, that assigns non-atomic 'labels' (i.e., composite paths) to pieces of text within an entry. Having functional properties of entry components defined decompositionally is important, because now fragments of dictionary entries can contribute to the lexical content of our target computational lexicon (or lexical knowledge base) not only by an interpretation of the immediate tag (terminal label) they carry, but by considering a complete or partial path, which indicates their overall participation in the lexical make-up of a language. We come back to this point in the next section, when we discuss in some detail the interpretation of the 'semantics' of path specifications, crucial to our notion of distributed lexical knowledge.

An immediate application of the path concept is its use in the design of an access mechanism for browsing through, and extracting data from, lexical databases. More specifically, if we assume a database model that holds instances of the structured representations produced by a mechanism similar to DEP, it is possible to specify, to arbitrary depth of detail, entries to be retrieved from the database by composing any number of paths into a declarative specification of a set of properties required of the entries sought. Paths can be fully instantiated, by 'quoting' literal strings to be found at terminal positions; alternatively, partially instantiated paths can be defined by assigning variables to terminal nodes or by leaving certain intermediate nodes in the path unspecified. A mechanism essentially equivalent to string calculus allows the specification of restrictions on, and constraints among, variables. Paths can be viewed as projections onto sets of

entries that fulfill such constraints; composing more than one path to a query and suitably interpreting the constraints associated with them makes it possible to combine statements concerning both content and structure of entries into arbitrarily complex search expressions. The process of search, or interpretation of the declarative specification of target entries, can be viewed as driven by unification. Neff et al. (1988) and Byrd (1989a) discuss a particular implementation of a query processor in some detail.

As an example, consider the following simple query designed to extract all (and only) nouns from a database constructed from a dictionary source.

```
LDOCE:
  entry
    ( .hdw: _word ;
      .homograph
        .syncat: "n" ;)
```

Without going into a detailed description of its syntax, this expression effectively specifies that all entries in the database (uniquely identified by the specified "**LDOCE**"), which contain an **entry . homograph . syncat** path with a terminal value of noun ("**n**") are valid search targets. Using a variable (**_pos**), the same effect – namely, extracting all noun entries – can be achieved by the following query.

```
LDOCE:
  entry
    ( .hdw: _word ;
      .homograph
        .syncta: _pos ; )
CONDITION ( _pos = "n" )
```

As an example of a more complicated query, designed to find all noun entries whose explicit cross-reference fields themselves include a non-empty implicit cross-reference (such as the "book" entry exemplified earlier), the query specification below merges the two paths leading to, respectively, **syncat** and to **implicit_xref . to.**

```
LDOCE:
  entry
    ( .hdw: _word ;
      .homograph
        ( .syncat: "n" ;
          .sense_def
            .explicit_xref
              .implicit_xref
                .to: _ixref ; ) )
CONDITION ( _ixref \= "" )
FORMAT    ( _word )
```

https://doi.org/10.1017/CBO9780511659478.006 Published online by Cambridge University Press

4.2.3 Structural properties of on-line MRD representations

The kind of structural analysis of dictionaries argued for here seeks to unfold *all* the functional implications of the font codes and other special characters controlling the layout of an entry on the printed page. As data is typically compacted to save space in print, and as it is common for different fields within an entry to employ radically different compaction schemes and abbreviatory devices, dictionary analysis faces non-trivial decompaction tasks. Furthermore, it is not uncommon for text fragments in a dictionary entry to serve more than one function; in such cases the analysis process should both identify the nature of the function and assign it proper structural representation. Neff and Boguraev (1989, 1990) present a detailed account of a comprehensive set of lexicographic conventions implemented via topography and carrying largely a semantic load; for illustrative purposes, as well as for reference in the next section, below are some examples of particular phenomena, together with the kind of fine-grained structural presentations assigned to them.

A particularly pervasive space-saving device in dictionary entries is the factoring out of common substrings in data fields. A definition-initial common fragment can be routinely shared by more than one sub-definition, as in "**incubator** . . . a machine for **a** keeping eggs warm until they HATCH **b** keeping alive babies that are too small to live and breathe in ordinary air". Similarly, a translation final fragment is not uncommon in bilingual dictionaries: "**Bankrott** . . . ~ **machen** to become *or* go bankrupt". A dictionary database should reflect this by duplicating the shared segments and migrating the copies as appropriate.

A more complex example of structure duplication is illustrated by a possible treatment of implicit cross-references, discussed earlier and exemplified here by a fragment of the entry for "nuisance".

> **nui.sance** /ˈnju:səns ‖ ˈnu:-/ *n* **1** a person or animal that annoys or causes trouble, PEST: *Don't make a nuisance of yourself: sit down and be quiet!* **2** an action or state of affairs which causes trouble, offence, or unpleasantness: *What a nuisance! I've forgotten my ticket* **3 Commit no nuisance** (as a notice in a public place) Do not use this place as **a** a lavatory **b** a TIP[4]

The dual purpose served by e.g., "TIP" requires its appearance on at least two different nodes in the structured representation of the entry, **def_string** and **implicit_xfr . to,** as shown in the figure below.

entry
+ − hdw: *nuisance*
+ − homograph
+ − print_form: *nuisance*
+ − pronunc . . . . . .
+ − syncat: n
+ − sense_def
| **+ − sense_no:** *1*

```
|   + − defn
|   |  + − implicit_xrf
|   |  |  + − to: pest
|   |  + − def_string: a person or animal that annoys
|   |                  or causes trouble: pest
|   + − example
|   + − example
|       + − ex_string: Don't make a nuisance of your-
|                      self sit down and be quiet!
. . . . . .
+ − sense_def
    + − sense_no: 3
    + − defn
        + − hdw_phrase: Commit no nuisance
        + − qualifier:  as a notice in a public
                        place
    + − sub_defn
    |  + − seq_no:  a
    |  + − defn
    |      + − def_string: Do not use this place as a
    |                      lavatory
    + − sub_defn
        + − seq_no:  b
        + − defn
            + − implicit_xrf
            |  + − to: tip
            |  + − hom_no: 4
            + − def_string: Do not use this place as
                            a tip
```

The sub-tree associated with the third sense definition of "nuisance" illustrates certain aspects of our analysis.

For instance, common substrings get replicated as many times as necessary and propagated back to their conceptually original places, as with "*Do not use this place as*". Multi-function entry components, such as implicit cross-reference kernels, now participate in more than one structure representation: the definition string itself, as well as the **implicit_xref cluster.** Note that an implicit cross-reference may itself be a structurally complex unit: in the example above, it consists of a kernel, "*tip*", and an annotation for the homograph number, "*4*", under which the relevant definition for the reference lexical item is to be found; in general, cross-references may also be annotated by sense number and additional morphological information.

Parenthetical strings are assigned functional labels: thus the string that ap-

pears in italics at source, "**as a notice in a public place**", is tagged as a **qualifier**. In general, parentheticals are commonly used by lexicographers to specify domains of use, selectional restrictions, typical collocations, and so forth; they also help in conflating more than one (related) definition under a single sense number (see Hanks, 1987, for an account of the use of parentheses in dictionary definitions). Additionally, or alternatively, a parenthesized fragment of a definition field can also be genuinely part of the definition string (for instance, "**clamour** . . . to express (a demand) continually, loudly and strongly"), in which case our grammar performs similar analysis to the one for implicit cross-references: the fragment is retained as part of the definition, as well as assigned a separate structural slot. The analysis of the entry for "accordion" illustrates this (note the replication of the string **key** as an implicit cross-reference and a parenthetical expression, in addition to its being part of the definition proper).

ac.cor.di.on / . . . / *n* a musical instrument that may be carried and whose music is made by pressing the middle part together and so causing air to pass through holes opened and closed by instruments (KEYs[1] (2)) worked by the fingers – compare CONCERTINA[1] – see picture at KEYBOARD[1]

entry
hdw: *accordion*
homograph
print_form: *ac.cor.di.on*
pronunciation . . . . . .
syncat: *n*
sense_def
sense_no: *1*
defn
implicit_xref
to: *key*
x_morph: *s*
how_no: *1*
s_no: *2*
par_string: *keys*
def_string: *a musical instrument that may be carried and whose music is made by pressing the middle part together and so causing air to pass through holes opened and closed by instruments (keys) worked by the fingers*
explicit_xref
how: *compare*
implicit_xref
to: *concertina*
hom_no: *1*

https://doi.org/10.1017/CBO9780511659478.006 Published online by Cambridge University Press

```
|  + −exrf_strong: concertina
|
+ −explicit_xref
    + −how: see picture at
    |
    + −implicit_xref
    |  + −to: keyboard
    |  + −hom_no: 1
    |
    + −exrf_string: keyboard
```

To summarize, the approach to dictionary analysis illustrated above expresses the crucial difference between our definition of *parsing*, and that of *tagging:* the latter involves, in principle, no more than identification of entry-internal field delimiters, their interpretation in context and markup of individual components by 'begin-end' brackets. It does not, however, extend to recovery of elided information; nor does it imply explicit structure manipulation (Boguraev et al., 1990, discuss the relationship between decompaction processes in dictionary analysis and the representational frameworks encoding the results of these processes).

The next section looks at the kind of generalizations of a semantic nature that can be made precisely because of the insights offered by an analysis of structural (and specifically, configurational) regularities of entries across the entire LDB representation of a dictionary.

4.3 Lexical knowledge in MRDs

Most of the work on deriving computational lexicons from machine-readable sources to date focuses on the individual lexical item. This particular perspective is especially visible in the context of providing phonological or syntactic information on a word-by-word basis. For instance, techniques have been developed for extracting from the pronunciation fields in a dictionary annotations suitable for driving speech recognition and synthesis systems; for mapping part of speech information to feature lists used for syntactic analysis; and even for constructing fully instantiated feature clusters, of the type posited by contemporary formal theories of grammar, from certain kinds of encodings of syntactic idiosyncrasies of words. Using such techniques, lexicons of non-trivial size have been constructed, thus providing 'proof of concept': fragments of dictionary entries can be formalized for the purposes of automated natural language processing.

4.3.1 Structure and organization of the lexicon

Most of the current work on fleshing out the semantic component of computational lexicons mimics the localist approach outlined above, by seeking to extract and formalize the information in certain fragments of dictionary definitions.

Typically, the target lexical entries encode, in a variety of ways, notions like category (type) information, selectional restrictions, polyadicity, case roles of arguments, and so forth. Although the utility of this kind of information for natural language processing is beyond doubt, the emphasis on the individual entry in separation misses out on the issue of global lexicon organization.

This is not to dismiss ongoing work that does focus precisely on this issue, for instance the attempts to relate grammatical nature with diathesis (e.g., Levin, 1985, 1990a, 1990b). Whether aspects of a verb's subcategorization, and specifically the range of alternative complement structures it can take, can be predicted from the semantic class of the verb and its predicate argument structure is not at issue here; rather, this is the kind of a question that can only be answered on the basis of applying strong methods of computational lexicography for analyzing data *across* entire dictionary sources.

Questions concerning the structure and organization of the lexicon are not uncommonly brought up in the context of studying linguistic and/or cognitive phenomena. Much of contemporary psycholinguistic research, in fact, exploits the assumption that the lexical component of language strongly interacts with the machinery (strategies and processes) underlying both language comprehension and generation. By that token, machine-readable dictionaries should, and have, become part of the methodology for studying the interactions between the (human) lexicon and the other language components. This is, however, outside of the scope of this chapter.

Of more immediate relevance are two related facts. First, studies in lexical semantics, even at a level where no richer representation is offered than named roles (Levin and Rappaport, 1986), have been shown to have immediate applications for improving the robustness of NLP systems (see, for instance, Katz and Levin, 1988). Second, an orthogonal view of MRDs, namely, regarding them as repositories of lexical knowledge and seeking to map their content onto a lexical knowledge base (Boguraev et al., 1989), inevitably commits to blurring the boundaries of individual lexical entries. A lexical knowledge base, in the sense of 'knowledge base' as used by the Artificial Intelligence community and employing representation techniques developed for the purpose, is likely to evolve as a highly interconnected tangled network. Hence it needs a richer notion of lexical relation than the conventional dictionary categories (of e.g., antonymy, synonymy, taxonomy, and so forth) provide.

The two issues above link work of a more theoretical flavor in the area of lexical semantics with the applied question of representation. It is not only the case that both of them require a better understanding of the global organization of the lexicon; it is also true that until it is well known what facts about the lexicon need to be represented formally, there is very little to be said (in specific terms) of the descriptive adequacy of existing representations.

Thus we arrive at the question of distributed lexical knowledge, because this is

where we are most likely to find clues for imposing structure on the lexicon. Examples here would be, for instance, facts like

1. the choice of a word usually carries with it a set of implications of semantic nature (Gruber, 1976);
2. partitioning verbs into classes on the basis of common semantic properties seems to have implications for the types of syntactic constructions that each class admits (Levin, 1990a, 1990b);
3. the notion of subtle shifts of meaning, such as lexical coercion, depends on the particular lexical decomposition assigned to a word (Pustejovsky, 1989; 1990).

In the remainder of this section we look at some examples illustrating ways in which such clues can be derived simply as configurational patterns over entry tree representations, described in terms provided by our detailed analyses of machine-readable dictionaries, and learned by studying the spread of different tree shapes across entire dictionary sources.

4.3.2 Paths in lexical databases and semantic fields

The discussion of global organization of the lexicon above and our search for a richer notion of lexical relation suggests that, informally, we assume a field structure for the lexicon. It turns out that this allows a particular interpretation of the 'path' concept, introduced in Section 4.2.2 above.

Consider the use of the implicit cross-reference notation, as exemplified in the entry for "book" (Section 4.2.2). As the guide for the dictionary states, implicit cross-references are used to draw the reader's attention to "related words in other parts of the dictionary". However, the convention is one of the most pervasive in the dictionary; consequently, implicit cross-references can be found in examples (as in the use of "SEAL" in the entry for "hermetic"), as part of definition strings (e.g. "ALCHEMY" in the entry for "hermetic"), as components of parenthetical expressions (e.g., "HYMN" in the definition of "chorale[1]", or "CAPITAL" in "stock[10]"), as auxiliary definitions (e.g., "CHORUS in "chorale[2]" or "LIVESTOCK" in "stock[7]" and so forth:

> **cho.rale** / . . . / *n* **1** (a tune for) a song of praise (HYMN) sung in a church: *a Bach chorale* **2** CHOIR(1); CHORUS[1](1)
>
> **her.metic** . . . *adj* **1** concerning magic or ALCHEMY **2** very tightly closed; AIRTIGHT: *a hermetic SEAL[2](4) is used at the top of this glass bottle*
>
> **stock . . . 7** farm animals, usu. cattle; LIVESTOCK . . . **10** the money (CAPITAL) owned by a company, divided into SHAREs

A superficial analysis of a dictionary, simply assigning **implicit_xref** labels to all instances above, would miss the different functions these serve in their respective definitions. Alternatively, an analysis that not only decomposes explicit cross-references, but also factors out implicit cross-references (as described in

Section 4.2.3), as well as parenthetical expressions and embedded (auxiliary) definitions (as introduced by ";" in "chorale[2]", "hermetic[2]", "stock[7]" above), naturally associates different semantics to the **implicit_xref** pre-terminal nodes. These interpretations can be read off directly from the path specifications, and the constraints associated with them: thus the case already discussed in Section 4.2 above ("LIBRETTO" in the definition of "book") is represented by the following path.

LDOCE:
entry
.homograph
.sense_def
.explicit_xref
.implicit_xref
.to: _ixref ;

The use of an implicit cross-reference as an auxiliary definition, as in "chorale[2]" ("CHORUS") and "hermetic[2]" ("AIRTIGHT") gives rise to

LDOCE:
entry
.homograph
.sense_def
.aux_def
.implicit_xref
.to: _ixref ;

Finally, for the cross-reference functioning as a parenthetical remark as well, as in "chorale[1]" ("HYMN") and "stock[10]" ("CAPITAL") have

LDOCE:
entry
.homograph
.sense_def
(.par_string: _par ;
.implicit_xref
.to: _ixrf ;)
CONDITION (_par = _ixrf)

These different structural analyses correspond to the different functions associated with the uses of the 'small caps' notation. In the first case above, the lexical item 'pointed to' as a cross-reference introduces a related word within a larger notion of domain: consider the lexical relationship between "book" and "libretto". In the second case, the relationship is that of an apposite synonym: "chorale" and "chorus", "hermetic" and "airtight", "stock" and "livestock". In the third case, an embedded defining pointer (denoted by its inclusion in parenthesis

without any auxiliary text) punctuates a concept used in the definition string proper: "stock" in its finance-related sense is defined not only as "money" (via the genus relation), but also as "capital" – the latter concept being a specialization of the former.

Such analyses help to explain our intuitive understanding of the different nature of the semantic classes of items that might occur in the respective fields of the dictionary. For instance, given the cluster of words (concepts) related to "libretto" (e.g., "PLAY", "SCORE", and so forth), it would not be surprising to find any of them in the same field, or textually in an adjacent position, in the entry for "book". Similarly, "BEGINNER" and "NOVICE" (in the entry for "neophyte" below) are fairly interchangeable with e.g., "APPRENTICE", "PUPIL", "LEARNER", precisely because the part of an entry they are to be found in is that denoting a synonymy relationship. On the other hand, because words like "MANUAL," "PUBLICATION", "ALBUM", or "DIARY" are not semantically related to "book[3]", they are unlikely to be found in the same position as "LIBRETTO".

> **ne.o.phyte** . . . a student of an art, skill, trade, etc., with no experience; BEGINNER; NOVICE

Viewed from an alternative perspective, the path specifications above, when suitably constrained and applied to appropriate dictionary entry contexts, can be embedded in queries to the database that would retrieve sets of words closely related along a particular semantic dimension – achieving in this way a mapping from a structural configuration (i.e., an LDB path) to a semantic field. A very simple example would be to use the second path above for extracting a fairly precise list of synonyms: a small sample of this list looks as follows:

hermetic	AIRTIGHT
hyperbole	EXAGGERATION
keep back	WITHHOLD
lead	CLUE
impoverish	DEPLETE
meddle	INTERFERE
metempsychosis	TRANSMIGRATION
shrink	PSYCHIATRIST
skin	PEEL
skin	FLEECE
percentage	PROPORTION
apparition	GHOST
coterie	CLIQUE

A different example would use the third path configuration above to refine a perhaps already existing taxonomic structure. Although methods have been developed for extracting such structures from dictionary sources (most notably by

Amsler, 1981, and Chodorow et al., 1985), the resulting taxonomies are typically broad and shallow, lacking in detail along the specialization dimension. However, extra depth in such structures can be introduced by observing that the configurational pattern exemplified in the definition of "stock[10]" (and "chorale[1]") above is quite common in the dictionary:

> **dandelion** . . . sometimes eaten but often considered a useless plant (WEED)
>
> **claymore** . . . a type of explosive weapon (MINE) for setting into the ground, . . .

These entries support the following semantic relationships, which introduce an additional level of specialization into a taxonomic structure (the symbol = > denotes an **is_a** link):

dandelion	= >	weed	= >	plant
claymore	= >	mine	= >	weapon
chorale	= >	hymn	= >	song (of praise)
stock	= >	capital	= >	money

In these examples, semantic fields are defined by sets of words collected, on the basis of a structurally defined common semantic function, from across the entire dictionary source. This is precisely the sense in which we define the notion of distributed lexical/semantic knowledge. Even though distributed semantic knowledge ultimately is compiled on an entry-by-entry basis, by collecting together suitable fragments of certain entries, the insights of what constitutes a 'suitable fragment', as well as the nature of the semantic relationship between such fragment(s) and/or headwords, can only come from fully exploiting the ability to cast very specific projections (or, equivalently, to look from very different perspectives) at a multi-dimensionally structured dictionary source.

4.3.3 Entry configurations and semantic regularities

With a structured dictionary representation available on-line, queries can be constructed to exploit the fact that configurational features of dictionary definitions have a mapping onto a unifying semantic property. In the following example, the internal structure of subdefinitions reflects a linguistic generalization that holds for a class of English verbs.

Case study – transitivity alternations

Katz and Levin (1988) make a strong case for the need to exploit lexical regularities in the design of natural language processing systems. The emphasis of their analysis rests with a class of verbs that undergo a number of transitivity alternations (Levin, 1985, 1990a). Levin further argues not only that machine-readable dictionaries *could* be used to evaluate hypotheses concerning the global

organization of the lexicon, but also that they *should* be used as major resources in the construction of (computational) lexicons (Levin, 1990b).

On the assumption that one aspect of lexical knowledge required for full-scale language processing would involve information concerning ergativity, it is not surprising that various attempts have been made to extract lists of verbs participating in a transitivity alternation from dictionary sources (e.g., by Levin, and by Klavans, personal communication). Such attempts have hitherto looked at flat (unanalyzed) dictionary sources and exploited various 'transparent' notational devices in the verb entries: for instance, grammar code collocations denoting both transitive and intransitive use of a verb in the same sense and/or the inclusion of a parenthetical "cause to" phrase in the verb definition. Strategies like this one have been designed to be triggered by the causative construction which, intuitively, might have been employed by lexicographers to express the alternation property. The resulting lists, however, turned out to be of a heterogeneous nature, including a large number of verbs that do not participate in a transitivity alternation, e.g.:

> **foul**[1] / . . . / *v* [T1;I0] **1** to (cause to) become dirty, impure, or blocked with waste matter: *The dog's fouled the path. One pipe has fouled, and the water won't go down.*

Even careful analysis of the range of defining patterns underlying the 'deep' semantics of transitivity alternations (such as carried out by Fontenelle and Vanandroye, 1989) fails to capture a particular structural regularity employed by the lexicographers to represent precisely the nature of a transitivity alternation. Yet we would expect that since regular linguistic properties of language (of which transitivity alternations are one example) tend to get reflected in the structure of dictionary definitions, an analysis like the one cited above would be enhanced by exploiting the structured representation of (verb) entries.

The query below, run against the *Longman Dictionary of Contemporary English* (LDOCE), is designed to extract those verbs marked both transitive (T1) and intransitive (I0) in the same sense, which also have two sub-senses (sub_defn).

```
LDOCE:
  entry
    .homograph
      (.word: _hw;
        .syncat: _pos;
        sense_def
          (.sense_no: _sno;
            .sub_defn
            (.seq_no: _sa;
              .defn.def_string: _defa;)
          .sub_defn
            (.seq_no: _sb;
```

https://doi.org/10.1017/CBO9780511659478.006 Published online by Cambridge University Press

```
                .defn.def_string; _defb;)
            .g_code_field: _gc;))
CONDITION (_pos('v',_pos) = 1 &
            _sa = 'a' &
            _sb = 'b' &
            pos('I0',_gc)>0 &
            pos('T1',_gc)>0 )
```

The results, a sample of which is given below, show a pervasive pattern: most of the subdefinitions contain a parenthetical expression, denoting the same object as the (syntactic) object and subject appropriate to the transitive and intransitive sub-senses of the verb. This pattern maps precisely onto the linguistic definition of a 'transitivity alternation' – the typical object in transitive use of the verb, e.g. "to cause (a horse) to go at the fastest speed", "to confuse (someone or someone's brain)" is the same as the typical subject in the intransitive form: "(of someone's brain) to become confused", "(of a horse) to go at the fastest speed" – but can only be expressed in structural terms. In addition, the sample definitions below demonstrate why the simpler queries of the earlier strategies failt to be triggered by a number of suitable verb candidates. In the case of searching for a "cause to" phrase in the definition, semantically related "to allow" or "to become" would miss out the entries for "float" and "addle(2)". Even enhancing a query to reflect such a relatedness (as, for instance, embodied in the search patterns of Fontenelle and Vanandroye who cast a wider net by positing lexicalizations of causativity to be denoted by phrases like "to become", "to allow", "to help", "to make", "to bring", and so forth) still would miss (the more naturally sounding) "loosen" and "separate" in the definition of "disengaged".

addle(1)	**a:** to cause (an egg) to go bad **b:** (of an egg) to go bad
addle(2)	**a:** to confuse (someone or someone's brain) **b:** (of someone's brain) to become confused
adjourn(1)	**a:** to bring (a meeting, trial, etc.) to a stop, esp. for a particular period or until a later time **b:** (of people at a meeting, court of law, etc.) to come to such a stop
careen(1)	**a:** (of a ship) to lean to one side **b:** to cause (a ship) to lean to one side
carry(18)	**a:** (esp. of a law or plan) to be approved; PASS **b:** to cause (esp. a law or plan) to be approved; (cause to) PASS
curl(1)	**a:** (of hair) to twist into or form a curl or curls **b:** to cause (hair) to twist into or form a curl or curls
disengage(1)	**a:** (esp. of parts of a machine) to come loose and separate

	b: to loosen and separate (esp. parts of a machine)
float(7)	**a:** to allow the exchange value of (a country's money) to vary freely from day to day
	b: (of a country's money) to vary freely in exchange value from day to day
flutter(4)	**a:** (of a thin light object) to wave quickly up and down or backwards and forwards
	b: to cause (a thin light object) to do this
gallop(1)	**a:** (of a horse) to go at the fastest speed
	b: to cause (a horse) to go at the fastest speed

4.4 Computational lexicography and lexical semantics

Structural analysis of on-line dictionary sources is only a necessary condition for setting up a framework for extracting lexical data and populating a lexical knowledge base. Just as no effective lexicon can be constructed from an MRD without reference to a formal theory of grammar (Boguraev and Briscoe, 1989), a semantic component of lexical knowledge cannot be derived outside a theory of lexical semantics. Without such a theory, any collection of dictionary LDBs is no more than a heterogenous body of seemingly *ad hoc,* and apparently incomplete and inconsistent, lexical annotations of words (see e.g., Atkins, 1990, for an analysis of the impoverished nature of individual dictionary definitions).

Furthermore, lexical databases, even though better suited for locating arbitrary fragments of lexical data, ultimately mimic the overall organization of dictionaries for humans – namely, that of individual entries, in alphabetical order, and with very little indication of interrelations between different words, as well as between word senses within an entry. Still, there is no reason to attempt to mimic this organization for the purposes of computational lexicons. Not only the nature of lexical access and language analysis is different from that of essentially consulting a reference book, but the technology of, e.g., knowledge representation (KR) makes it possible to consider novel ways of structuring lexical knowledge on a large scale. To quote Randolph Quirk: "we should re-evaluate lexicographic practice not to suit the computer, but because of it".[4] Indeed, a number of proposals already exist for importing tools and methodologies of KR into the design of computational lexicons (see e.g., Evans and Gazdar, 1989; Boguraev and Pustejovsky, 1990); as a parallel development lexicographers are also becoming increasingly dissatisfied with the limitations of the conventional form and media of dictionaries: "the traditional dictionary entry is trying to do what the language simply will not allow to be done" (cf. Atkins, 1990, on the inadequacy of the linear sense definition, subordinated by hierarchically organized

[4]From an invited speech at the Fifth Annual Conference of the University of Waterloo Center for the *New Oxford English Dictionary*, September 1989, Oxford.

sense distinctions, to convey the rich interdependencies between words and word senses).

4.4.1 From lexical description to language processing

From the perspective of natural language processing, the value of any lexical knowledge base is ultimately to be judged by the degree of support it offers for tasks like syntactic analysis and semantic interpretation. For the purposes of effective integration of the representation of lexical semantic information with its use in a compositional semantics model of interpretation, it is necessary to adopt a theory of lexical semantics that satisfies at least two criteria. It should be amenable to strict formalization; i.e., it should go beyond just descriptive adequacy. At the same time, it should fit naturally into a general framework of linguistic description and processing.

This is, arguably, the only way in which the distributed lexical information available in machine-readable dictionaries can be made directly usable for natural language processing. It follows that any questions concerning the contribution of computational lexicography to the enterprise of building a (computational) lexicon can be answered in a positive way only by incorporating into its framework a particular view on the issue of lexical decomposition. Some remarks concerning the view taken in this research were already made in the previous section.

To summarize here, the approach underlying this particular exercise in computational lexicography takes as starting points current research in lexical semantics. Of specific relevance are two lines of work. Pustejovsky's notion of generative lexicon, designed to cope naturally with phenomena underlying the creative use of language – such as ambiguity and lexical coercion – provides a framework for his proposal for different levels of lexical representation, and an especially rich lexical semantics for nominals, based on qualia theory (Pustejovsky, 1989, 1990). Levin's study of verbal diathesis (Levin, 1985, 1990a) looks at the global organization of the lexicon; more specifically, one question concerns the ways in which the relationship between semantic categorization and syntactic behavior reveals semantically interesting classes of verbs.

There is a particular justification for attempting to develop a model of the lexicon that encodes the kinds of lexical properties and generalizations posited by these theories. If such a lexicon can be made to support a processing model of language within the current (unification-based) accounts, and if its structure naturally fits the kind of a (lexical) semantic taxonomy underlying the incorporation of global semantic constraints into such accounts (as proposed by e.g., Moens et al., 1989), then at least two propositions hold.

First, the kinds of lexical distinctions highlighted as relevant from a theoretical standpoint (and required by the processing framework) offer substantial direction to the efforts of extracting lexical data from MRDs, by means of the tools and methodologies developed by computational lexicography. Second, since these

theories embody to a certain extent the notion of distributed lexical knowledge, they provide, by the same token, a hitherto uncharted route for exploring the lexical data in MRDs, itself distributed across entire dictionary sources.

Briscoe et al. (1990) present a proposal for integrating current research on lexical semantics with a unification-based account of language processing along the lines sketched above. More specifically, the approach is to develop "a model of lexical semantic representation which supports a (somewhat enriched) compositional account of (sentence) meaning by enriching lexical representations of nouns and collapsing those for verbs with alternate grammatical realisations". Starting from a position like this, the task of computational lexicography shifts in emphasis: now we are concerned not so much with local questions like what might constitute an acceptable (semantic) definition of a given word, but with more global ones like what words might be grouped together under a particular projection of lexical meaning.

For more details on the processing framework itself, and the support it offers for the treatment of e.g. lexical coercion, logical metonymy and so forth, the reader is referred to Briscoe et al. (1990). Below we illustrate specifically how the tools of computational lexicography facilitate the location and extraction, from structured dictionary representations, of lexical data required by such a framework. We sketch two possible applications of these tools: fleshing out qualia structures and seeking verbs which induce type coercion.

What can be done to a book?

Central to the issues brought above is the notion of 'spreading the semantic load' equally among the different category types. In such a framework nouns and adjectives, for instance, have a better founded notion of the kinds of semantic contexts in which they can function. As a result, problems of, e.g., lexical ambiguity and underspecification can be tackled on the basis of a richer characterization of an entry in terms of a set of distinctions that generalize on established notions like predicate-argument structure, primitive decomposition, conceptual organization and selectional constraints. In essence, the answer to the question "what can be done to a book?" – namely, in the absence of other overriding information, books can be read, written, reviewed, criticized, and so forth – can be used to guide an analysis and interpretation procedure in the derivation of a semantic structure for inputs like "a long book" and "he enjoyed the book". Such functional specifications of nominals – together with other aspects of their lexical semantics, such as physical attributes, distinguishing characteristics, and so forth – constitute a system of relations (Pustejovsky calls these "qualia structures") that characterize the meaning of a noun, much in the same way as a system of arguments characterizes the meaning of a verb. (For a more detailed account of a theory of lexical semantics instrumental in derivations equivalent to "a long book to read" and "he enjoyed reading the book", see Pustejovsky, 1989, 1990.)

Our approach to dictionary analysis, as discussed in Section 4.2, takes the view that entry fragments performing more than one function are replicated as many times as necessary in the LDB representation; furthermore, the functions of a fragment are made explicit through the appropriate path specifications. In particular, since phrases in parentheses serve, among other things, to denote typical arguments of verbs, parenthetical expressions in dictionary definitions are factored out as separate terminal nodes in the LDB representations. Consider, for example, the (fragment of) the entry for "censor" below, together with its structured analysis that assigns a separate path to the parenthetical expression "books, films, letters, etc." denoting a range of typical objects.

> **censor . . .** to examine (books, films, letters, etc.) with the intention of removing anything offensive

entry
|
+ − homograph
+ − word: *censor*
|
. . . . . .
|
+ − sense_def
|
+ − defn
+ − par_string: *books, films, letters, etc.*
+ − def_string: *to examine (books, films, letters, etc.) with the intention of removing anything offensive*

On the basis of this analysis, and following an observation that a fairly precise distinction between a parenthetical denoting typical subject(s) and one denoting typical object(s) can be drawn both on the basis of its position in the definition string, and on its internal structure (consider the entry for "sag" below), it is possible to construct a query against the dictionary database that effectively lists a number of verbs (actions) typically applied to books.

> **sag . . .** (of a book, performance, etc.) to become uninteresting during part of the length

The query extracts from the database those verbs that have a non-empty, not definition-initial, parenthetical expression containing the string "book".

```
LDOCE:
  entry:
    (hdw: _hw ;
      homograph:
      (syncat: 'v' ;
        sense_def:
        (defn:
```

https://doi.org/10.1017/CBO9780511659478.006 Published online by Cambridge University Press

```
                (par_string: _par;
                  def_string: _def ))));
CONDITION (_def \= '''' &
                _par \= '''' &
                pos (_par,  _def) > 1
                pos ("book", _par) > 0 );
```

The resulting list, a sample of which is given below, is a (not necessarily complete) answer to the question of the title.

annotate, censor, consult, excoriate, autograph, classify, cross-index, expurgate, bowdlerize, collate, dramatize, footnote, catalogue, compile, entitle, page, pirate.

It is worth emphasizing the similarity between the nature of the query and the type of data retrieved from the dictionary, in this case and in the earlier search for ergative verbs (Section 4.3.3). Due to the fine-grained analysis of a dictionary, carried out in the process of converting it into a lexical database, we are able to express a semantically interesting query in purely structural terms.

A sample from a similar query, tailored to the specific structural properties of the *Collins English-French Dictionary,* is illustrated below.

abridge, ban, bring out, castrate, abstract, bang about, burlesque, chuck away, appreciate, borrow, call in, churn out, autograph, bowdlerize, castigate, commission.

A very similar query from the *Collins English-German Dictionary* results in even more verbs.

Although techniques like this seem to yield significant data, it is clear that they could, and should, be improved. Further level of detail in the queries, yielding richer results, can be achieved in at least two ways. An obvious improvement is to refine the ways in which typical objects are introduced in dictionary definitions, as had to be done in the case of the *Collins* bilinguals. Even more coherent data is obtained by 'spreading the net wider', which overcomes two inherent problems of dictionary sources: their inconsistency and economy of representation. Since it is unrealistic to expect an entirely uniform pattern of dictionary definitions across the entire dictionary, the original question is generalized by expanding the set of preferred (typical) objects from a single "book" to include suitably semantically related concepts: "literature", "something written", "something printed", and so forth. Such sprouting techniques have been proposed earlier in different contexts (e.g., by Byrd et al., 1987), and they are particularly well-suited to our concerns here. The parameters of the sprouting – namely, the set of semantically close terms and phrases – can be identified on the basis of an analysis of the defining vocabulary and the related conceptual structure underlying the dictionary definitions.

Why is regretting similar to enjoying?

In the processing framework assumed here, the interpretation of a phrase like "enjoy the book" is not triggered only by the specific knowledge of what can be done to books; that is, the qualia structure alone is insufficient to activate the process of type-raising an object (e.g., as that denoted by "book") to an event (such as "reading the book"). Rather, it is the composition of a particular category of verb with the lexical semantics of its object that triggers this process. It follows that "enjoy", and a number of verbs similar to it in that they denote type-coercing predicates, have to be suitably marked. We are then faced with the question of how well the methods of computational lexicography might find such verbs in our dictionary sources.

We approach the problem from two angles. On the one hand, we start with a lexical 'seed' – that is, a set of verbs representative of the phenomenon we seek – and apply suitably constrained sprouting techniques to grow the sample. On the other hand, we design a query against a structured lexical database, incorporating salient properties of the lexical class in question. The query is then incrementally refined, as we intersect the two search/retrieval strategies.

In this particular case, one way of deriving an initial lexical seed is to consult Levin's verb classification system. One of the fundamental assumptions behind that work is that the organization of lexical items (verbs) into classes reflects shared components of meaning. More specifically, commonalities within a class cover, among other things, possible expressions of arguments and possible extended meanings. Even allowing for multiple class membership, it is likely that a set of verbs 'similar' to "enjoy" (where similarity is along some unspecified dimension) will contain more than one entry representative of a type-coercing predicate. Levin (1990a) categorizes, under the exemplary member "admire", a set of psychological verbs (the class is subdivided into two groups, representative of positive and negative emotions):

admire, adore, appreciate, cherish, enjoy, esteem, exalt, fancy, idolise, like, love, miss, respect, revere, stand, support, tolerate, value, worship, . . .

abhor, deplore, despise, detest, disdain, dislike, distrust, dread, envy, fear, hate, lament, loathe, mourn, pity, resent, scorn, . . .

Although not all members of this class resemble "enjoy" fully, items like "appreciate", "fancy", "hate", "lament", "like", "loathe", "love", "miss", "tolerate" are capable, in some of their senses, of demanding type-raising of their arguments in object position for full interpretation of verb phrases. Consider, for instance,

Do you fancy a cup of tea?
I find I miss the telephone, since we've moved,
I particularly loathed team games at school, and
It is not unheard of to tolerate opinions other than your own.

There are a number of ways in which a lexical seed can be grown; we follow the general method of Byrd et al. (1987), which involves traversing, perhaps bidirectionally, a lexical network along hypernym, synonym, and antonym links. Thus the initial set can be expanded[5] to include, for instance,

1. "imagine" and "desire" (as "fancy" is defined in LDOCE to be "to form a picture of; imagine", while COBUILD (Sinclair, 1987) gives "desire" as its superordinate),
2. "relish", "permit", and "allow" (as "to relish" is "to enjoy; be pleased with", and "to tolerate" is "to allow (something that one does not agree with) to be done freely and without opposition; permit"),
3. "prefer", "be partial to", "wish", "incline towards" (as these, together with "fancy" and "desire", are synonyms in the *Collins Thesaurus;* see McLeod, 1984).

One problematic issue in traversing networks derived from MRD sources is that of ambiguity: since the relationships embodied in a lexical relation structure (e.g., one representing taxonomy or synonymy) are between words, we still have to determine the exact word senses (with respect to a dictionary) for which the relationship holds. Although in principle this may turn out to be an arbitrarily complex problem to solve, in this particular case we can use additional information to constrain our search.

Starting from the position that there is some correlation between semantic properties of verbs and their syntactic behavior, we only consider those senses of the words derived by sprouting that are marked in the dictionary to take NP objects and/or progressive or infinitive verb phrase complements.[6] This considerably narrows down the search space. In fact, looking for verbs with particular predicate-taking properties is the basis of the second strategy for extracting type-coercing verbs similar to "enjoy".

In essence, we construct a query against a lexical database that encapsulates salient syntactic and semantic properties of such verbs. In addition to specifying subcategorization frames, the query embodies properties of dictionary definitions that reflect the common, representative semantics of this verb class. Thus we look for words like "experience", "action", "event" used to denote the object of a verb in its definition. This finds entries like

[5]Even though the data structures used in such traversal are, logically, equivalent to networks, they are represented using the LDB format described earlier; consequently, suitably constrained and chained queries against lexical databases can be used to emulate chain traversal.

[6]A number of English monolingual dictionaries utilize some system of encoding the complement-taking properties of verb entries. Examples here are the grammar coding systems of the *Longman Dictionary of Contemporary English* and the *Oxford Advanced Learner's Dictionary of Current English* (Hornby, 1974), as well as the more mnemonic notation used by the Collins COBUILD dictionary: consider for instance, the entry for "enjoy" in COBUILD, which is annotated V+O or V+-ING.

enjoy . . . 1 [T1,4] to get happiness from (things and experiences)

finish . . . 1 [I0; T1,4] to reach or bring to an end; reach an end of (an activity)

prefer . . . 1 [T1(*to*),3,4(*to*) . . .] to choose (one thing or action) rather than another; like better

regret . . . 1 [T1,4,5a] be sorry about (a sad thing or event)

The above definitions are similar in many respects: they share common grammar code descriptions ("T1" and "T4" are the Longman equivalent of the COBUILD "V+O" and "V+-ING" annotations) and parenthetical expressions impose type restrictions of the kind we seek on their objects. Furthermore, we observe from the initial response to the query another configurational regularity: there is a trailing preposition at the end of the definition proper (occasionally followed by the object denoting parenthetical expression). This suggests a further refinement to the query; in effect, we are again 'spreading the net', but rather than using sprouting techniques over lexical networks, we are elaborating a set of queries that return different, and partly overlapping, projections of a lexical database. For instance, one way of 'relaxing' the query would be to allow for an underspecified object: this technique yields entries like

allow . . . 1 [T1,4;V3] to let (somebody) do something; let (something) be done; permit . . . **2** [T1;V3] to make possible (for); provide (for): *This plan allows 20 minutes for dinner . . .*

After incorporating the specification for a trailing preposition in the definition string, we obtain a new set of candidate type-coercing verbs including

hate . . . 1 [T1,3,4;V3,4] to have a great dislike of . . . **2** [T1,3,4;V3,4] (*infml*) dislike: *She hates fish . . .*

fancy . . . 1 [T1;V4;X(*to be*)1] to form a picture of; imagine 3[T1;X(*to be*)1] to have a liking for; wish for: *I fancy a swim*

loathe . . . [T1,4] to feel hatred or great dislike for

relish . . . [T1,4] enjoy; be pleased with: *to relish a funny story*

Even with this small sample of the results, we can observe that the two strategies outlined above begin to converge. Thus, there are direct overlaps between the lists produced by sprouting and query refinement: for instance, both "fancy" and "allow" are found by either method. In addition, indirect, but strong, evidence in support of entries on the lists is furnished by the lexical overlaps underlying the network traversal: all of "visualise", "envisage" and "imagine", which have been identified as candidates by the criteria encoded in the lexical query, are listed as synonyms in the *Collins Thesaurus*. Note that this is in addition to yet more links being provided by the dictionary itself, via the mechanism of indirect synonym introduction within dictionary definitions discussed in 4.3.2 above.

A more representative (but still incomplete) list of type-coercing verbs similar to "enjoy" extracted by the methods described here is illustrated below:

acknowledge, advise, advocate, avoid, be partial to, begin, deny, desire, discourage, endure, enjoy, envisage, fancy, finish, forbid, hate, imagine, incline towards, justify, lament, like, loathe, love, necessitate, prefer, propose, regret, relish, resume, suggest, tolerate, warrant, wish.

From a dictionary to a lexicon

One final point needs to be made here. Dictionaries are incomplete and unreliable, as well as not fully consistent in form and content of definitions. This is an uncontroversial statement, and has been argued for (and against) quite extensively. Thus it is hardly surprising to find the lists derived in the two earlier sections missing certain (more or less obvious) necessary elements. For instance, in the case of looking for the default predicates naturally composable with "book", the most common – and by that token the most relevant – ones, namely, "read" and "write", are not part of any of the answers.

One of the concerns of computational lexicography is to remain aware of this fact, and consequently to develop techniques and methods for ensuring that the computational lexicons derived from machine-readable sources are more consistent, as well as fully representative with respect to the various lexical phenomena encoded in them.

In the general case, the real issue here is not about any particular verbs; in fact, enhancing the techniques for elaborating the lexical semantics of nominals (as in the case of "book" above) along the lines presented earlier does give reading and writing as actions suitable for composition with books. Rather, the question is: how can we make absolutely certain that complete lists of collocationally appropriate forms, ranked by relevance, can be derived systematically for any kind of input? The answer to this question comes from a separate line of research, becoming integral to the study of word meaning and already beginning to extend the definition of "computational lexicography" given in the beginning of this chapter. Machine-readable dictionaries are not the only type of large-scale lexical resource available; equally important, and arguably richer and more representative of real language use, are the text corpora. Traditionally the basis for inducing stochastic models of language (see e.g., Garside et al., 1987), text corpora more recently have been used for extraction of a variety of lexical data (e.g., by Atkins, 1987; Wilks et al., 1989; Church and Hanks, 1990; and Hindle, 1990).

This chapter has looked specifically into a particular framework for populating a lexical knowledge base with data extracted from MRD sources. The complementary aspect of this same question, namely the use of corpora for enhancing dictionary data, is the subject of separate studies. The reader is referred to Boguraev et al. (1990a,b) for more specific remarks on applying methodologies similar to the ones discussed here, in the context of deriving semantics of nominals, to large-scale machine-readable corpora; similarly, Briscoe et al. (1990) discuss the role of corpora in evaluating the results of type-coercing predicate extraction procedures.

4.5 Conclusion: what computational lexicography is and isn't

In this chapter we have attempted to place computational lexicography in the larger context of related studies of lexical semantics and, to a lesser extent, knowledge representation. From the perspective of one particular goal in applied computational linguistics – namely, that of building large-scale, comprehensive lexicons for natural language processing – we propose a framework for locating and extracting a range of lexical semantic relations, ultimately of use to practical implementations of formal accounts of language.

This framework substantively relies on detailed analysis of existing machine-readable sources, followed by theory-driven search for lexical properties across the entirety of these sources. In particular, strong connections are sought between a lexical characteristic and ways(s) in which it might be encoded configurationally, i.e., as a structural pattern common to a set of dictionary entries. Such connections, once established, are then used to 'carve out' projections from the dictionary, by suitably composing arbitrarily complex queries against structured dictionary database(s).

We put forward a view that the tools and methods of computational lexicography are best put to use in such a goal-driven context: rather than attempting to develop a procedure (or a set of procedures) for automatically constructing a lexicon on the basis of the information available in published dictionaries, the emphasis should be on striving for better understanding of the mapping between language description and dictionary design, and exploiting this for fleshing out components of the lexicon as required by any particular theory of (formal) lexical semantics.

The methods and techniques outlined in this chapter fall in the category of 'weak strategies', insofar as they can never be trusted to deliver complete and coherent results. Ultimately, two independent lines of work are needed to expand the scope of what has been traditionally considered the subject area of computational lexicography.

There is a natural extension of the notion of distributed lexical knowledge, in that any large text sample contains such knowledge – albeit in even less structured form than a dictionary. Given that one of the characteristics of lexical information extracted from dictionary sources is its incompleteness, it is clear that more raw data is required. Text corpora provide such data, and techniques for large corpus studies (many of them, in fact, parallel to the structured analysis paradigm proposed here) should rightly belong to the arsenal of computational lexicography. A different kind of concern is that, given the inherently noisy nature of any data derived (whether from a dictionary or from a corpus), it is imperative to develop methodologies for its evaluation and validation.

Unlike developments in corpus analysis, this latter line of work is still in very early stages. However, there is a growing realization that before a (practical) computational lexicon is instantiated with semantic information on a large scale,

it is necessary to go through an intermediate stage of a lexical knowledge base: a holding store for information concerning words and their use of language (see, for instance, Boguraev and Levin, 1990). The enterprise of constructing such a knowledge base should exploit recent developments in the area of knowledge representation; one side effect of this strategy would be the ability to exploit, and build on, an inventory of general purpose data validation techniques.

In this chapter, then, we take the position that computational lexicography is not all there is to constructing a large-scale lexicon for natural language processing; however, it is an essential part of the enterprise.

References

Alshawi, H. (1989). "Analyzing the Dictionary Definitions", in B. Boguraev and E. Briscoe (Eds.), *Computational Lexicography for Natural Language Processing,* Longman, London, 153–170.

Alshawi, H., Boguraev, B., and Carter, D. (1989). "Placing LDOCE On-line", in B. Boguraev and E. Briscoe (Eds.), *Computational Lexicography for Natural Language Processing,* Longman, London, 41–64.

Amsler, R. A. (1981). "A Taxonomy for English Nouns and Verbs", *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics,* Stanford, California, 133–138.

Atkins, B. T. (1987). "Semantic ID Tags: Corpus Evidence for Dictionary Senses", *The Uses of Large Text Databases,* Third Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary: Waterloo, Canada, 17–36.

Atkins, B. T. (1990). "Building a Lexicon: Beware of the Dictionary", MS, Oxford University Press (paper presented at a BNN Symposium on Natural Language Processing, Cambridge, MA) (this volume).

Boguraev, B. and Briscoe, E. J. (Eds.). (1989). *Computational Lexicography for Natural Language Processing,* Longman, London.

Boguraev, B. et al. (1989). "Acquisition of Lexical Knowledge for Natural Language Processing Systems", Technical Annexe, ESPRIT Basic Research Action No. 3030, Brussels.

Boguraev, B., Briscoe, T., Carroll, J., and Copestake, A. (1990a). "Database Models for Computational Lexicography", *Proceedings of Euralex-VOX – Fourth International Congress on Lexicography,* Malaga, Spain.

Boguraev, B., Byrd, R., Klavans, J., and Neff, M. (1990b). "From Structural Analysis of Lexical Resources to Semantics in a Lexical Knowledge Base", RC #15427, IBM T. J. Watson Research Center, Yorktown Heights, New York.

Boguraev, B. and Pustejovsky, J. (1990). "Lexical Ambiguity and the Role of Knowledge Representation in Lexicon Design", *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, Finland.

Boguraev, B. and Levin, B. (1990). "Models for Lexical Knowledge Bases", *Proceedings of the 6th Annual Conference of the UW Centre for the New OED,* Waterloo, Ontario.

Briscoe, T., Copestake, A., and Boguraev, B. (1990). "Enjoy the Paper: Lexical Semantics via Lexicology", *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, Finland.

Byrd, R. J. (1989a). "LQL User Notes: An Informal Guide to the Lexical Query Lan-

guage", Research Report RC 14853, IBM Research Center, Yorktown Heights, New York.

Byrd, R. J., Calzolari, N., Chodorow, M., Klavans, J., Neff, M., and Rizk, O. (1987). "Tools and Methods for Computational Lexicology", *Computational Linguistics,* vol. 13(3–4), 219–240.

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

Calzolari, N. (1988). "The Dictionary and the Thesaurus Can Be Combined", in M. Evens (Ed.), *Relational Models of the Lexicon,* Cambridge University Press, Cambridge,UK, 75–96.

Calzolari, N.and Picchi, N. (1986). "A Project for a Bilingual Lexical Database System", *Advances in Lexicology,* Second Annual Conference of the UW Centre for the New Oxford English Dictionary, Waterloo, Ontario, 79–92.

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

Chodorow, M. S., Byrd, R. J., and Heidorn, G. E. (1985). "Extracting Semantic Hierarchies from a Large On-line Dictionary", *Proceedings of the Association for Computational Linguistics,* Chicago, Illinois, 299–304.

Church, K. and Hanks, P. (1990). "Word Association Norms, Mutual Information and Lexicography", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics,* Vancouver, British Columbia, 76–83. (Full version in *Computational Linguistics,* 16[1].)

Evans, R. and Gazdar, G. (1989). "Inference in DATR", *Proceedings of the Fourth Conference of the European Chapter of the ACL,* Manchester, 66–71.

Fontenelle, T. and Vanandroye, J. (1989). "Retrieving Ergative Verbs from a Lexical Data Base", MS, English Department, University of Liege.

Garside, R., Leech, G., and Sampson, G. (1987). *The Computational Analysis of English: A Corpus-Based Approach,* Longman, London and New York.

Gonnet, G. (1987). "Examples of PAT", Technical Report OED-87-02, University of Waterloo Center for the New Oxford English Dictionary, Waterloo, Ontario.

Gruber, J. (1976). *Lexical Structures in Syntax and Semantics,* North-Holland, Amsterdam.

Hanks, P. (1987). "Definitions and Explanations", in J. M. Sinclair (Ed.), *Looking Up: An Account of the COBUILD Project in Lexical Computing,* Collins ELT, London and Glasgow, 116–136.

Hindle, D. (1990). "Noun Classification from Predicate-Argument Structures", *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics,* Pittsburgh, PA, 268–275.

Hornby, A. S. (1974). *Oxford Advanced Learner's Dictionary of Current English* (Third Edition), Oxford University Press, Oxford, UK.

Katz, B. and Levin, B. (1988). "Exploiting Lexical Regularities in Designing Natural Language Systems", *Proceedings of the 12th International Conference on Computational Linguistics,* Budapest, Hungary, 316–323.

Kazman, R. (1986). "Structuring the Text of the Oxford English Dictionary through Finite State Transduction", University of Waterloo Technical Report No. TR-86-20.

Levin, B. (1985). "Lexical Semantics in Review: An Introduction", in B. Levin (Ed.), *Lexical Semantics in Review,* Lexicon Working Papers 1, Massachusetts Institute of Technology, 1–62.

Levin, B. (1990a). *The Lexical Organization of English Verbs,* Department of Linguistics, Northwestern University, Evanston, Illinois.

Levin, B. (1990b). "Building a Lexicon: the Contribution of Linguistic Theory", MS, Oxford University Press (paper presented at a BBN Symposium on Natural Language Processing, Cambridge, MA) (this volume).

Levin, B. and Rappaport, M. (1986). "The Formation of Adjectival Passives", *Linguistic Inquiry,* 17(4), 623–661.

McLeod, W. T. (1984). *The Collins New Thesaurus,* Collins, London and Glasgow.

Moens, M., Calder, J., Klein, E., Reape, M., and Zeevat, H. (1989). "Expressing Generalisations in Unification-Based Grammar Formalisms", *Proceedings of the Fourth Conference of the European Chapter of the ACL,* Manchester, 174–181.

Neff, M. and Boguraev, B. (1989). "Dictionaries, Dictionary Grammars and Dictionary Entry Parsing", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics,* Vancouver, British Columbia, 91–101.

Neff, M. and Boguraev, B. (1990). "From Machine-Readable Dictionaries to Lexical Databases", *International Journal of Lexicography.*

Neff, M., Byrd, R., and Rizk, O. (1988). "Creating and Querying Hierarchical Lexical Data Bases", *Proceedings of the Second Conference on Applied Natural Language Processing,* Austin, Texas, 84–93.

Procter, P. (1978). *Longman Dictionary of Contemporary English,* Longman, Harlow, UK.

Pustejovsky, J. (1989). "Current Issues in Computational Lexical Semantics", Invited Lecture, *Proceedings of the Fourth Conference of the European Chapter of the ACL,* Manchester, England, xvii–xxv.

Pustejovsky, J. (1990). "The Generative Lexicon", *Computational Linguistics,* vol. 17.

Raymond, D. and Blake, E. G. (1987). "Solving Queries in a Grammar-Defined OED", Unpublished Technical Report, University of Waterloo Centre for the New Oxford English Dictionary, Waterloo, Ontario.

Sinclair, J. (Ed.). (1987). *Collins COBUILD English Language Dictionary,* Collins, London and Glasgow, UK.

Vossen, P., Meijs, W. and den Broeder, M. (1989). "Meaning and Structure in Dictionary Definitions", in B. Boguraev and E. Briscoe (Eds.), *Computational Lexicography for Natural Language Processing,* Longman, London, 171–192.

Wilks, Y., Fass, D., Guo, C.-M., McDonald, J., Plate, T., and Slator, B. (1989). "A Tractable Machine Dictionary as a Resource for Computational Semantics", in B. Boguraev and E. Briscoe (Eds.), *Computational Lexicography for Natural Language Processing,* Longman, London, 193–228.