12 Lexicon Acquisition for NLP: A Consumer Report
SERGEI NIRENBURG

1. INTRODUCTION

Current natural language processing systems typically operate in a 'demo' mode—they sometimes feature sizeable grammars but seldom sizeable lexicons containing information about meaning. This difficulty of going beyond toy systems is one of the main bottlenecks of artificial intelligence in general. Scaling up the dictionaries (and other knowledge bases) of a knowledge-based system is, however, essential for the overall success of the field. There are several ways in which the indispensable massive knowledge acquisition program can in principle be conducted. One can envisage developing a machine learning system for automatic acquisition of vast quantities of knowledge through experimentation on a (sub)world. Such a system can in principle delimit concepts and thus build an ontology and/or the model of a specific domain.1 Connecting elements of a natural language with elements of this ontology will, of course, be a separate non-trivial task. Alternatively, one can design a natural language processing system whose task would be to read natural language texts and understand them. As the next step, these programs would compile facts they had learned about entities in the world as well as elements of language into on-line encyclopaedias and dictionaries suitable for use in computational-linguistics and general artificial intelligence applications. It is doubtful whether these approaches can succeed, at least in the short run, because in order to build a machine learning or a natural language processing system with the above-mentioned capabilities we first need to supply it with knowledge bases

1 I would like to thank Yorick Wilks for a number of thought-provoking discussions; Ken Goodman and Christine Defrise for commenting on early versions of this text; and Donna Gates, Todd Kaufmann, Teruko Mitamura, Eric Nyberg, and Inna Nirenburg for help with the illustrative material.

1 The terms 'ontology' and 'domain model' are used here in roughly the same sense as 'epistemological' and 'conceptual' knowledge in Brachman (1979).
and dictionaries of a size and sophistication comparable to those which these systems would be supposed to produce (cf. the discussion in Lenat and Guha 1988: 19). We arrive at the bootstrapping problem which seems to be central in many approaches to (at least semi-) automated knowledge acquisition.

It is very time-consuming to handcraft lexicons of the kind that most knowledge-based language processing systems require. But until recently the prevalent attitude was that it is impossible to finesse the knowledge-acquisition problem, and that one must therefore be prepared to 'bite the bullet' even though it might take several person-centuries to be completed. Let us call this the pessimistic position.

It is natural, however, to look for ways of simplifying the acquisition task. The recent availability of on-line human-oriented dictionaries has raised hopes for the success of a large-scale automatic dictionary acquisition task. Significant results have been obtained in extracting information supporting morphological (e.g. Byrd et al. 1986) and syntactic (e.g. Boguraev and Briscoe 1987) analysis. Researchers working in this paradigm are optimistic—they believe that the amount of effort to produce a large, computer-program-oriented lexicon through processing human-oriented dictionaries will be significantly smaller than that required for hand-coding it. However, relatively few practical results have been so far obtained in extracting meaning from MRDs and in representing it in a format suitable for the use of application programs. Considering the complexity of this task, it is not surprising.² Often research of this type leads not directly to creating a program-oriented lexicon but rather to a prerequisite for it—such is, for instance, the research on extraction of large hierarchies of genus terms (Amsler 1980; Chodorow et al. 1985).

The advent of the MRD era has brought about yet another alternative to handcrafting large computational lexicons—the direct use of human-oriented dictionaries to support natural language processing applications. A representative example of this approach is the work of Jensen and Binot (e.g. 1987), in which heuristic rules are developed for determining semantically appropriate prepositional phrase attachment based on the information contained in human-oriented dictionaries. Even though it is unlikely that this approach will become practical soon, exploring novel ways of using MRDs should be encouraged alongside the goal-driven efforts for providing adequate lexicons for particular systems and application domains.

Developers of comprehensive natural language applications such as

² Of course, it is not easy to construct even a purely syntactic 'polytheoretical' lexicon—as is, for instance, shown by Ingrina (1987), in which a large set of information items for syntactic lexicons is sketched, and many existing program-oriented syntactic dictionaries are compared.
machine translation (MT) systems or human–computer interfaces can already use some of the results of MRD processing, especially with respect to syntactic information. Recent results and new projects, such as, for instance, Knight’s (1989) research on lexicon acquisition in the framework of the CYC project (Lenat and Guha 1988), Boguraev’s (personal communication) new work on extracting semantic information from MRDs and especially the results of the New Mexico State University group (e.g. Wilks et al. 1988) promise significant practical reverberations in the not too distant future. At the same time, it does not seem realistic to expect all the problems of lexicon acquisition for computational applications to be solved through research in transforming MRDs serving humans into machine-tractable dictionaries (MTDs in Wilks’s terminology) serving computer programs.

At present no natural language system employs automatic means of dictionary acquisition. In part this is because many such systems are of the ‘demo’ kind and do not require sizeable lexicons. The direct use of machine-readable dictionaries as resources for NLP does not seem to be a viable option, since the size of the knowledge base of specialized heuristics and techniques for extracting the various types of information from such dictionaries will, for any realistic system, be of the same order of magnitude as a specially handcrafted lexicon. (For a more detailed justification of this position, with respect to machine translation applications, see Slocum and Morgan 1987.) Work on automatic compilation of MTDs from MRDs has become by now a well-defined research area whose results will be immediately useful for acquiring knowledge in NLP systems. However, this point will not be reached overnight, and the potential consumers of the automatic knowledge extraction technology should not (and could not) postpone their own work until such results are obtained. A reasonable strategy is to develop a methodology of gradual enhancement for automation in lexicon acquisition, preferably in the framework of a hybrid, computer- and human-initiative knowledge acquisition environment. I will return to this central point later in section 5.

Some such hybrid work can already be done today. Consider transfer-oriented MT systems. Few of them employ extensive semantic analysis, even though their lexicons sometimes list semantic and pragmatic information. Thus, for example, in an early version of the METAL machine translation system (Bennett and Slocum 1985) monolingual lexicon entries include the TAG features which lists subject domain information, the TY feature which stands for the semantic type (presumably from an independently derived semantic type list), and the ARGS feature which specifies, albeit in syntactic terms, the valency information about the entry head. In the current
METAL system the subject domain information is used for lexical disambiguation (Geert Adriaens, personal communication) during the transfer stage:

Fehler → bug when TAG: DP (data processing)
Fehler → mistake when TAG: GV (general vocabulary).

It seems quite conceivable that a significant portion of such a dictionary could be produced automatically using current dictionary processing techniques. The extraction of the above semantic features can be achieved using a program like Slator's lexicon-provider (Wilks et al. 1988). Thus, the TAG feature in METAL is similar to Slator's 'pragramatic' slot; the TY and ARGS features could be derived from Slator's 'type' slot. Of course, additional research has to be performed to allow for automatic reformatting of human-oriented bilingual dictionaries as well as dictionaries of proper names, abbreviations, and specialized subject domain terminology.

Knowledge of specialized terminology, including abbreviations and proper names (see a discussion in Amsler 1987a, 1987b) is also needed for the support of knowledge-based machine translation systems. Dictionaries for such systems are typically required to contain significantly more semantic and pragmatic information than transfer-oriented MT systems. In addition, they have to rest on a language-independent knowledge base describing the subworld of translation as well as some very general ontological statements about the world, about speakers and hearers, and about the structure of texts and dialogues. These features are needed to support semantic and pragmatic disambiguation during the analysis stage. Knowledge requirements for knowledge-based MT systems are discussed in some detail in Nirenburg (1987). In principle, such systems must possess most of the types of inference-making capabilities of natural language processing systems in AI (cf. e.g. Waltz 1982, for a survey of relevant problems). Dictionaries supporting target language generation in knowledge-based MT must include information about synonyms, antonyms, and hyperonyms, which is needed to enhance the expressive power of the lexical selection module, notably, to support the ability to generate definite descriptions (cf. Sondheimer et al. 1988). Another generation-related requirement is providing information about syntagmatic collocation constraints on target language lexical units to ensure that the target text is colloquially adequate. Thus, while WIDE and BROAD are synonyms, the former collocates with VARIETY and the latter does not. Collocation constraints, unlike selectional restrictions, are highly idiosyncratic and do not lend themselves easily to generalizations. If a degree of automation in the
acquisition of this information is desired, special dictionaries must be used (e.g., the BBI dictionary (Benson et al. 1986)).

In section 2 I will briefly describe the structure of the lexicons in two application-oriented NLP systems developed at Carnegie-Mellon University in KBMT-89, a knowledge-based machine translation system, and DIOGENES, a natural language generator. Detailed descriptions of these systems can be found in Goodman (1989) and Nirenburg et al. (1988a), respectively. Knowledge acquisition in these projects has been performed manually, with the help of specially designed tools, briefly described in section 3. Having gone through relatively large-scale knowledge acquisition, we can clearly see the advantages of at least partial automation of this process. In section 4 a preliminary design for a knowledge-acquisition environment is presented. This environment would include some functionalities from the MRD-to-MTD research as well as enhancing human-directed acquisition. Developing a ‘knowledge-acquisition workstation’ of this sort can be accomplished without having to wait for the MRD-to-MTD research to reach a high level of practicality.

2. LEXICONS IN KBMT-89

KBMT-89 is an interlingual MT system in which source language texts are analysed into a language-independent internal representation, an interlingua text (ILT) from which the corresponding target text is then generated. The system architecture is illustrated in Fig. 12.1.

KBMT-89 employs three types of lexicons: the concept lexicon, the analysis lexicon and the generation lexicon. The KBMT-89 concept lexicon records the semantic information necessary for parsing source texts and generating target texts (the domain of translation in KBMT-89 is that of computer hardware installation manuals). World knowledge in KBMT-89 is organized as a multiply interconnected network of concepts implemented as frame structures in the knowledge-representation language FrameKit (Nyberg 1988). General ontological postulates are encoded in the higher, more abstract nodes of this network and serve as a map to help knowledge enterers determine where domain concepts fit into the knowledge hierarchy. They also include properties and constraints that can be further specified in defining domain concepts. For KBMT-89, we have built a domain model describing interactions between personal computers and their users and the manuals that guide this interaction. Elements of the ontology and the domain model are used to describe the meanings of lexical units in the source and target languages. Figs.
Fig. 12.1 The architecture of the KBMT-89 system
12.2–12.5 present knowledge available in the concept lexicon about several ontological concepts, including the relevant fragments of the inheritance hierarchies and the actual detailed concept descriptions. (see Nirenburg and Raskin 1987a, 1987b, for a detailed discussion of the KBMT approach to lexicon acquisition).

2.1 The analysis lexicon

Analysis lexicons in KBMT-89 contain the morphological and syntactic information, word-to-concept mapping rules, and mapping rules that link case role structures of concepts with subcategorization patterns of their realizations in the source and target languages. In KBMT-89 these are known as lexical and structural mapping rules, respectively.

The English and Japanese analysis lexicon entries generally include
the following parts: a word, its category, its inflection, its root form, and the syntactic features and mapping rules associated with that word. Syntactic features and structural mapping rules can be defined in an analysis lexicon entry either locally or through an inheritance mechanism, via a pointer to a location in the syntactic feature hierarchy.
and the *structural mapping hierarchy*, respectively. Thus, these hierarchies become a component of the analysis lexicon. The structure of the KBMT-89 analysis lexicon entry is given below.

Here are some examples of the KBMT-89 English analysis lexicon entries:

(‘command’ (CAT N)
 (CONJ-FORM SINGULAR)
 (FEATURES
   (CLASS DEFAULT-NO-NOUN-FEAT)
   (all-features ((PERSON 3) (NUMBER SINGULAR) (COUNT YES) (PROPER NO)
     (MEAS-UNIT NO) (ROOT COMMAND))))
 (MAPPING
   (local
     (HEAD (REQUEST-EVENT-OF-COMPUTER)))
   (local
     (HEAD (COMMUNICATIVE-CONTENT (THEME-OF
       (INSTRUCT)))))
   (local
     (slots (GOAL = OBJ)
      (‘word’
        (CAT <cat>)
      $?INFL [+ (<infl-type> <form>)]))
Fig. 12.5 Concept description: record-information

```prolog
?('INF-FORM <infl-type>)
?('ROOT &value)
?('FEATURES)
?('CLASS [+ <class-name-FEAT>])
([" (LOCAL [+ (<feature> <value>)])])
```
(ALL-FEATURES <f-structure>)

(MAPPING
(CLASS [+ <class-name-MAP>])
[*

(LOCAL
(SEM-TEST [+ <test>])
(SYN-TEST [+ <test>])
(+ (HEAD <sem-head> [* <slot-value-pair>]) |
  (MODIFIER <slot-value-pair>))|
  (SLOT [+ <structural-mapping>])))))
(SOURCE = (PPADJUNCT (PREP = FROM)))))
(CLASS OBJECT-MAP)))

('note' (CAT N)
(CONJ-FORM SINGULAR)
(FEATURES
(CLASS DEFAULT-NOUN-FEAT)
(all-features ((PERSON 3) (NUMBER SINGULAR)
  (COUNT YES) (PROPER NO)
  (MEAS-UNIT NO) (ROOT NOTE)))
(MAPPING
(local
  (HEAD (MENTAL-CONTENT)))
(local
  (HEAD (TEXT-GROUP (CONVEY (COMMUNICATIVE-
    CONTENT))))))
(CLASS OBJECT-MAP)))

('note' (CAT V)
(CONJ-FORM INFINITIVE)
(FEATURES
(CLASS CAUS-INCHO-VERB-FEAT)
(all-features
  (*OR*
    ((FORM INF) (VALENCY (*OR* INTRANS TRANS))
      (COMP-TYPE NO))
  
  (ROOT NOTE))
  ((PERSON (*OR* 1 2 3)) (NUMBER PLURAL) (TENSE
    PRESENT)
    (FORM FINITE) (VALENCY (*OR* INTRANS TRANS))
    (COMP-TYPE NO) (ROOT NOTE))
  ((PERSON (*OR* 1 2)) (NUMBER SINGULAR)
    (TENSE PRESENT)
    (FORM FINITE) (VALENCY (*OR* INTRANS TRANS))))
(COMP-TYPE NO) (ROOT NOTE)))
(MAPPING (local
 (HEAD (RECORD-INFORMATION)))
 (CLASS AG-TH-VERB-MAP))

And here is an example entry from the Japanese analysis lexicon.

('torinozoku' (CAT V)
 (MAPPING
   (local
     (HEAD (REMOVE)))
     (CLASS AG-TH-VERB-MAP))))

2.1.1 Structural mapping rules

The structural mapping rules map grammatical functions, such as subject and object, on to semantic roles in ILTs. The general format of the structural mapping rules is as follows:

(<class-name-MAP>
 (MAPPING
   [*
     *(LOCAL
       *(TEST [+ <test>])
         *(SLOT [+ <structural-mapping>]]))
     *(CLASS [+ <class-name-MAP>]]))

In the specification above the LOCAL slots are used for mapping rules pertaining to a specific lexical item or a subclass of such items. The CLASS slot introduces inheritance of structural mapping properties which pertain to larger word classes. In the example below we illustrate several levels of the structural mapping rule hierarchy. The basis for assigning a verb to a certain class is its transitivity pattern and the similarity of mappings between its grammatical functions and case roles of the corresponding ontological concepts. The structural mapping rule for the English verb CHANGE is:

('CHANGE' (cat V)
 (FEATURES (class CAUS-INCHO-VERB-FEAT))
 (MAPPING (local (slot
   (goal = (ppsubjunct (prep = to)))))
   (class C-I-VERB-MAP)))

The necessary mapping rule is inherited by CHANGE from the following:

(C-I-VERB-MAP
 (MAPPING
   (local
   (SLOT
     (goal = (ppsubjunct (prep = to))))
     (CLASS CAUS-INCHO-VERB-FEAT))))
(test (passive = —) (valency = intrans))
(slot (theme = subj))
(class CB-TH-VERB-MAP CB-TR-VERB-MAP CB-TP-VERB-MAP))

The verb belongs to the causative-inchoative verb class (c-i-verb-map). It can be inchoative, as in the English sentence Therefore, even if you push the Kana key, the square on the screen will not change. In this sentence the corresponding ontological concept is *to-replace-display-object. The subject NP should be mapped to the semantic slot theme. The verb change can also be causative, as in On-line button changes the on-line mode and off-line mode. Here, the appropriate concept is again *to-replace-display-object. Since the subject NP (on-line button) is inanimate, it should be mapped to the semantic slot caused-by. (Only animate entities and the computer can be agents). The object NP (the on-line mode and off-line mode) maps into the theme semantic slot. The necessary rule is inherited by the c-i-verb-class from the following:

(CB-TH-VERB-MAP
(MAPPING
   (local
      (test (passive = —) (valency = trans))
      (slot (caused-by = subj)
         (theme = obj)))
   (local
      (test (passive = +))
      (slot (caused-by = obl-agent)
         (theme = subj))
      (sem-test (focus theme)))
   (local
      (test (passive = +))
      (slot (theme = subj))
      (sem-test (focus theme)
         (caused-by unknown)))
   (class AG-TH-VERB-MAP))

The structural mapping rule for ‘agent-theme’ verbs (the AG-TH-VERB-MAP class in the rule above) is as follows:

(AG-TH-VERB-MAP
(MAPPING
   (local
      (test (passive = —) (valency = trans)
         (mood = (*not* imperative)))
      (slot (agent = subj)
(theme = obj))
(local
  (test (passive = −) (valency = trans)
   (mood = imperative))
  (slot (theme = obj))
  (sem-test (agent * reader)))
(local
  (test (passive = +))
  (slot (agent = obl-agent)
   (theme = subj))
  (sem-test (focus theme)))
(local
  (test (passive = +))
  (slot (theme = subj))
  (sem-test (focus theme)
   (agent unknown)))
  (class ALL-VERB-MAP))

2.2 The generation lexicon

The KBMT-89 generation lexicons are used in the generation phase by
the lexical selection module. This module accesses them to determine,
for a target language, the correct open-class lexical items for a given
interlingua text. The generation lexicons contain all open-class items
(nouns, verbs, adjectives, and adverbs). Much of the information in
the analysis lexicons can be used in generation lexicons, too. For
logistical reasons, however, in KBMT-89 the format of the generation
lexicons is slightly different from that of the analysis lexicons.
Generation lexicon entries can be automatically obtained from lexical
entry slots of concepts. Valency information for verbs, morphological
information, and subcategorization information are manually added to
the entries.

KBMT-89 generation lexicon entries contain a frame name, a
meaning pattern, a lexeme, syntactic features, and other data, in the
following format:

(make-gl-entry
  <entry-name>
  <meaning-pattern>
  <collocational-information>
  <syntactic-features>
  <subcategorization-information>
  <morphological-information>
Following is an example of the KBMT-89 Japanese generation lexicon entry, for arawar eru ('to appear'):

(make-gl arawar eru1
  (is-token-of (value *appear))
  (syntactic-info (cat v)
    (features (cat v) (root arawar eru)
     (subcat intrans) (vtype v-1dan)))
  (subcategorization-info (req theme) (opt goal)))

Note that the name of the entry arawar eru1 and the is-token-of slot can be automatically generated from the entry's counterpart definition in the concept lexicon:

(make-frame *appear
  ... (jhead ((arawar eru (cat v))))

The syntactic-info slot values are taken from the lexicon file of the Japanese grammar. The feature values ((cat v) (root arawar eru) (subcat intrans) (vtype v-1dan)) will constitute the initial f-structure with the *appear frame in an input ILT file and with the correct Japanese selection, i.e. arawar eru1. The subcategorization-info slot restricts the set of entries that can be selected. It does this by specifying which semantic slots in the ILT are appropriate for a given entry.

A domain model may have only one concept, *two-wheeled-vehicle, to represent the meanings of any of the following English lexical units: bicycle, motorcycle, tandem. But, if we need to generate those lexical units correctly from a uniform ILT representation, we need additional information about their differences. This information is stored in the meaning pattern component of an entry and is used for resolution of lexical synonymy during generation. Generally, the meaning pattern is a constrained ontological concept or property value (cf. Nirenburg et al. 1988a or Nirenburg and Nirenburg 1988 for a detailed description). Meaning patterns are essentially devices for taking care of the differences in lexicalization granularity in different languages. Knowledge-based machine translation has been often criticized on the grounds that it is allegedly unable to treat the dozens of words denoting different types of snow in Eskimo languages or different types of sand in Kushitic languages. Of course, one could introduce a subhierarchy with the concept *snow at its head in the domain model. But if this is considered unnecessary, then the lexicon for an Eskimo language will contain entries for the various words denoting types of snow, each entry having a different meaning pattern which
would in its particular way constrain the general concept of *snow. The following is an example of a KBMT-89 generation lexicon entry.

(make-gl-entry shipping-cardboard
  (is-token-of (value *PROTECTION-MATERIAL))
  (MADE-OF (value *CARDBOARD))
  (syntactic-info (CAT N)
    (FEATURES (count yes)
      (root shipping cardboard)))
  (subcategorization-info NIL)
  (morphological-info (infl (plural nil))))

In this example the collocation-information slot does not appear. This knowledge was not used in KBMT-89. However, it is used in the generation system DIOGENES (Nirenburg et al. 1988a), of which the generation module in KBMT-89 is a modified subset. A major difference between these two generators is that the latter uses information about collocation properties of lexical units, not only their selectional restrictions. As a result, the lexicons in DIOGENES have to include additional information. For example, if the theme of the causative action instance is lexically realized as step, the action itself should be realized as take. When the same theme is realized as measure, the causative action has to be realized as make. The important thing to note is that the above choice depends not on the meaning of the theme but rather on the particular lexical choice for it.3

3. ONTOS: A KNOWLEDGE ACQUISITION AND MAINTENANCE SYSTEM

The KBMT-89 ontological and domain models, as well as the lexical mapping rules for both analysis and generation, were acquired with the help of the knowledge acquisition and maintenance system ONTOS. The system provides an interactive environment that includes facilities for interacting in multiple windows through menus and graphics. ONTOS contains ontological postulates that are the basis for:

- memory and decision aids to help users to find the appropriate way to describe a newly-entered concept; and

3 The treatment of collocations in DIOGENES is discussed in detail in Nirenburg et al. (1988a) and Nirenburg and Nirenburg (1988). Cumming (1987) is a survey of generation-oriented lexicon research and also includes a chapter on collocations. Mel'čuk’s explanatory-combinatorial dictionaries (e.g. Mel’čuk and Polgaire 1987) are designed to contain a wealth of collocation information.
• a filtering capability that helps identify potential inconsistencies and problems in the knowledge base.

These ontological postulates are embodied in a network representation of objects, events, and situations, characterized by attributes and relations, and organized in taxonomic and meronymic hierarchies. Integrated acquisition of domain concepts proceeds through elaboration and specification of concepts framing ontological postulates. The result is what we call an ontology. We believe that such an environment facilitates both:

• human-initiated knowledge acquisition and
• semi-automatic and automatic knowledge acquisition in which a knowledge-acquisition and maintenance system acquires knowledge from MRDs, machine-readable encyclopaedias, and corpora by suggesting new, partially characterized concepts to be elaborated on, refined, and sometimes deleted by humans or by the system itself as new information is encountered in the process of running an application.

A detailed discussion of the functionalities of ONTOS can be found in Nirenburg et al. (1988b). A view of the ONTOS screen during an acquisition session is given in Fig. 12.6, which shows a typical ONTOS screen with two browser windows displaying taxonomic subnetworks for the concepts ALL and EVENT, a system menu, a query command menu, a man–machine dialogue area, and an editor window for the frame corresponding to the concept EVENT. Actual modifications to the content of the ontology are carried out through the editor interface. Communication with the machine is carried out through typing, menus, and mouse clicks in browser windows.

4. DISCUSSION

In this chapter we first briefly sketched the directions of MRD-related research. Next, we reviewed the knowledge-acquisition requirements and environments of two related knowledge-based NLP systems. At this point we would like to discuss the methodology of the MRD research vis-à-vis knowledge acquisition for NLP systems.

In KBMT-89 and DIOGENES we were able to acquire an ontology of about 1,000 concepts, with the corresponding lexical mappings attached, in approximately five person-months. This was performed manually, using ONTOS and the augmentor as tools. It is clear that for a larger-scale domain a significant enhancement of acquisition capabilities will be necessary. In particular, it would be desirable to use
results of the research on format conversion of MRDs. The question is: how realistic is this expectation?

The MRD-to-MTD research operates at present under several methodological constraints which, while making certain results feasible, somewhat limit the kinds of semantic and lexical knowledge that can be included in the outputs of current systems. Wilks et al. articulate some such constraints as the following methodological questions (1988: 5):

- **sufficiency**: ‘whether a dictionary is a strong enough knowledge base for English’;
- **extractability**: ‘whether it is possible to specify a set of computational procedures that operate on an MRD and extract, through their operation alone and without any human intervention, general and reliable semantic information on a large scale and in a general format suitable for, though independent of, a range of subsequent NLP tasks’;
- **bootstrapping**: whether it is possible to create the initial
information on which the MRD-to-MTD procedures rely from the dictionary itself.

Wilks et al. answer all these questions in the affirmative. However, from the point of view of application development the above constraints seem to be too strong. As briefly noted above, a single human-oriented dictionary can hardly be considered sufficient for supporting a realistic application. Even if we agree that LDOCE can be used for extracting a general ontology, it is not a useful source of specialized domain information, abbreviations, or proper names. Moreover, one can find better sources for synonymy and hyperonymy information, collocations, idioms, etc. In addition to LDOCE, Longman has published, for example, a thesaurus-like Lexicon of Contemporary English as well as a Dictionary of Phrasal Verbs and a Dictionary of English Idioms. A comparison between an output of Slator's Lexicon Producer program and the contents of a lexicon entry in a system such as KBMT-89 shows that the latter contains a significantly larger amount of knowledge. One can also argue for using machine-readable encyclopedias alongside MRDs (cf. Amsler 1987a; Walker 1987). An existing, handcrafted domain model (such as, for instance, a rule base for an expert system or a handcrafted computer-oriented encyclopedia, as in the CYC project (Lenat and Guha 1988)) or a set of core theories of the world, as advocated by the TACITUS project (e.g. Hobbs et al. 1987), can serve as the basis for at least some computational-lexicographic work. Finally, to serve a realistic-size domain, lexical knowledge-acquisition systems will routinely have to make use of text corpora from this domain, as training texts on which to test and improve a system.

As far as extractability is concerned, this assumption has been partially shown correct by the fact that the programs by Plate, Guo, and Slator from New Mexico State University (see Wilks et al. 1988)—as well as other programs, such as Chodorow et al. 1985—already produce useful and largely reliable information. Thus, in the NMSU environment the output of the MRD-to-MTD program by Slator produces dictionary entries used in at least two consumer programs—meta5 (Fass 1988) and PREMO (Slator 1988). However, the amount of knowledge provided for MTDs by Slator's LDOCE parser is somewhat limited. Many selectional restrictions and preferences are left out of the definitions. In order to be successful, the tree interpreter routine used by Slator for adding such information through analysis of differentiae in LDOCE has to rely on a very detailed and extensive set of heuristic rules for meaning assignment to nominal, adjectival, prepositional, and other modifiers. The acquisition of this set of rules
is difficult to automate. At the same time, their availability is a prerequisite for producing comprehensive MTDs. Complete extractability remains, thus, a research objective, not a reality.

The further requirement of format generality (and therefore transportability) is more difficult to meet in practice without developing systems for translation among different knowledge-representation languages. Even when such systems are developed, the added complexity of software maintenance will be a strong vote against this method of achieving knowledge shareability.

Finally, internal bootstrapping will only work for closed knowledge repositories, such as LDOCE. Not all lexical units in text corpora used for testing and tuning an application can be expected to appear in an MRD. Whether additional concepts can be described using the primitives derived from an MRD (such as the set of modified controlled vocabulary items in LDOCE, as used by Guo's program, see Wilks et al. 1988) or whether additional primitives will have to be introduced remains an open question to be treated empirically.

5. SYNTHESIS

One way to make the MRD-to-MTD research results more usable today is to relax some or all of the above methodological assumptions, especially that of sufficiency. This will, however, make building automatic MRD-to-MTD transformation systems more difficult. A more realistic and practically feasible way of making this research more useful in the immediate future is, at least temporarily, to forgo the requirement of complete automation.

The gap between the immediate needs of the application systems and the current capabilities of the MRD-to-MTD systems can be bridged in a mixed-initiative knowledge acquisition system that would allow for human augmentation of results obtained through automatic processing of human-oriented knowledge repositories such as dictionaries of various kinds, encyclopedias, and corpora.

Just as in KBMT-89 the augmentor is used for improving and enhancing the results of automatic text analysis, in a hybrid knowledge-acquisition environment a similar module can be used to augment the candidate MTD entries produced by an MRD-to-MTD system. In such an environment, the human knowledge enterer will have an interactive access to a variety of machine-readable information resources, including MRDs, encyclopedias, and corpora. Current advances in the design of interfaces facilitate the integration of all these diverse resources in a single workstation. By the same token, the
human user will have a window into the existing portions of the MTDs.

Several components of such a system already exist in the KBMT-89 knowledge-acquisition environment. One of our current projects is devoted to enhancing current acquisition capabilities with a view of developing a full-fledged lexical knowledge-acquisition workstation for machine translation and other NLP applications. We intend to achieve this by adding an MRD-to-MTD capability and a battery of human-oriented machine-readable lexical and interface resources to the KBMT-89 knowledge-acquisition environment.

6. A MULTI-PURPOSE PROCESSING AND ACQUISITION ENVIRONMENT

A typical configuration of an advanced NLP application is as shown in Fig. 12.7. This is an abstraction of the structure of a KBMT system in Fig. 12.1. In the figure, the double arrows on the left represent the flow of data from the input text through various internal representations to an output text and the flow of control in the system. The numbered single arrows represent the flow of reference data between the various data repositories ('static knowledge sources') and the various processing modules. In the very general view of Fig. 12.7, the static knowledge sources are the various natural language grammars, the various machine-tractable dictionaries, and the representations of the system's domain model(s) on which the semantic components of these dictionaries are based. The processing modules are, in order of application, the analysis module, the reasoning module, and the generation module. The analysis module has the task of obtaining a natural language input (which could be a text or a dialogue turn) and producing a representation of its meaning, sufficiently detailed for a particular reasoning module (a question-answering system, an information retrieval system, a machine-translation system, etc.) to function successfully. Depending on the type of the latter, the analyser will have to perform some or all of the following types of work on any given input: morphological analysis, syntactic analysis, lexical-semantic analysis, semantic dependency determination, treatment of reference, determination of pragmatic parameters, such as, for instance, speech act information, discourse cohesion, speaker attitudes, or speech situation parameters.

The task of the generation module is to obtain an internal representation of the text that the reasoning component has produced and realize its meaning in a target language of choice. Depending on
Fig. 12.7 A prototypical comprehensive NLP system
the type of input and the type of application, the generation component can be as basic and primitive as a selector of prefabricated 'canned' messages or it can involve such component tasks as target text sentence boundary and configuration determination, lexical selection of target language open- and closed-class lexical items, determination of the syntactic structure of target language clauses, introduction of anaphoric, deictic, and elliptical ways of rendering input meanings, ordering of target language elements (e.g. ordering of adjectives in a noun phrase), realization of the various pragmatic factors (e.g. politeness levels, verbosity/lapidity, discourse cohesion clues, speech acts, etc.), and syntactic and morphological decisions.

The reasoning components of the system differ dramatically from application to application. For instance, if the whole system is an advanced spelling checker, the analyser will be constrained to dealing with inflectional and possibly derivational morphology; it will operate with a single word at a time, and the result of the analysis will be either an 'OK' message or an error flag, optionally, with suggestions for correction. The reasoning system in this case will obtain an incorrectly spelled word and will attempt to analyse it, with the help of its internal knowledge about typical misspellings. This knowledge will constitute the ontology or the domain model for this particular application. The results of this operation will be passed on to the generator, which will produce appropriate messages for the user, possibly through a set of 'canned' messages.

In a knowledge-based machine translation system the analyser performs morphological, syntactic, semantic and pragmatic analysis of the input text and produces an internal representation from which the generator later produces a target language text. In a completely automatic full-quality MT system of this sort the reasoning module is empty. However, in reality this type of system is not immediately feasible, the reason being the lack of capability of always producing complete, correct, and unambiguous results from the analysis module. Therefore, a special type of the reasoning module has to be added to such a system. Its task will be to aid the analyser in selecting the most appropriate candidate reading of the input text. This reasoning system can be either an 'expert system' with heuristic rules specific to the task at hand acquired by knowledge engineers in interviews with professional translators or it can be an interactive editor (for a detailed discussion of this concept see Kay 1971; Carbonell and Tomita 1987; Brown 1989) in which the decisions will be made by human users. The reasoning system in this latter case will contain the knowledge about ways of presenting human editors with a variety of decision problems in a convenient way.
In a natural language interface to a database the analyser's task is translating a natural language input into a query in a data manipulation language, such as QUEL. The reasoning system picks up this query and performs its instructions, returning an answer. The task of the generator in such a system is to render the answer in a form convenient for the user. In some systems the output is produced using a standard data manipulation language, so that the generator becomes spurious. In some other such systems canned responses (filled templates of answers) are used. Finally, some of the interfaces perform actual generation of responses in a natural language.

The nature of the dictionaries, ontologies, and grammars also depends on the particular application. In fact, depending on a theoretical approach taken to certain parts of the analysis and generation tasks, even the boundaries between the grammars and the dictionaries on the one hand and the dictionaries and the ontologies on the other can sometimes get blurred. Indeed, in some approaches the morphological information about lexical units is stored in the grammars, in some others, in the dictionaries. In some approaches most of the syntactic knowledge is stored in the grammars. In some others (notably, in the unification-oriented approaches), much of the syntactic knowledge is stored in the lexicon. In the lexico-semantic corner of the knowledge, in some approaches the meanings of natural language units are considered to be instances of concept types stored in the ontological models. In some other approaches the importance of the ontological model is downplayed, and the language in which lexical meanings are explained is that of word senses.

In Fig. 12.7 the single arrows have the following meanings. Arrow 1 symbolizes the flow of data from the grammar to the analyser in response to an analyser request for grammar rules for processing an input. Arrows, 2, 3, 4, 5, 6 and 8 have similar meanings in that they symbolize the flow of data from the static knowledge sources to the actual programs. The direct connection of the ontology to the analyser and to the reasoning system corresponds to the necessary use of heuristic rules based on world knowledge in order to attain realistically useful levels of understanding of input (arrow 2) and to support the actual reasoning processes (arrow 3).

The only link in Fig. 12.7 that does not involve a processing module is arrow 7, which symbolizes the relationship between the ontology and the lexicon. This connection can have two facets. First, some parts of the knowledge required in dictionary entries can be physically stored in the ontological models, because the meanings of open-class items can be interpreted as instances of particular ontological concepts. Second, all practical natural language processing systems
expect their ontologies and lexicons to grow in the process of their exploitation. Therefore, there is a knowledge acquisition importance to the connection between the ontology and the lexicons. The acquisition operations can be initiated on either side of the arrow. If a new open-class lexical unit is to be introduced into the system, it is first entered into the lexicon, and then its meaning is interpreted in terms of either existing ontological concepts of a new set of concepts that have to be included in the ontology in order to express this meaning. Conversely, if it becomes clear that the ontology for a particular application has to be expanded, a set of new concepts has to be added to the ontology. After this is done, it becomes necessary to add to the lexicons entries (or entry parts) that realize the meanings of the new concepts in any or all of the natural languages in the system.

Acquisition of ontologies and lexicons is typically manual and is usually performed in an 'off-line' mode, with the processing system turned off. This means that natural language processing systems are not usually designed for 'during the operation' updates of their grammar and lexicon support. In the advanced state-of-the-art natural language processing environments special background acquisition support systems are designed and implemented. Fig. 12.8 illustrates schematically the process of grammar, lexicon, and ontology acquisition in such systems. Usually, there are two types of interfaces—one for grammar acquisition and the other for dictionary acquisition. The latter system is sometimes also used for specifying ontologies, in those NLP systems which rely on ontology acquisition for their operation. Grammar acquisition interfaces allow for quick browsing through sets of grammar rules and types of structures that these grammars generate. An example of this type of system is the interface for LFG acquisition developed at XEROX or the InterCoder interface of the METAL machine translation system.

Similarly, current ontology and lexicon acquisition systems are typically manual. (There are a number of experimental systems attempting automatic lexicon acquisition—notably Wilks et al. 1990. However, they are not yet sufficiently general to support a realistic application.) A typical manual lexicon and ontology acquisition interface is comprised of a graphics interface with browsing capabilities, which allows the representation of ontological concepts and lexicon entries as networks of nodes and links, thus facilitating comparison, search, generalization, and information retrieval. An advanced acquisition system can also contain other means of access to the knowledge—for instance, a query subsystem that allows information retrieval through a menu-oriented and/or command-language interface. In order to support modification of the knowledge base, an
acquisition system must support an editor, preferably structured to reflect the basic data organization in the system (for example, representation systems can be either frame based or rule based). Finally, such a system will benefit from the existence of an automatic means of checking the validity of a newly introduced knowledge element as well as its consistency with objects already existing in the knowledge base. The above functionalities are all present in the knowledge acquisition system ONTOS (e.g. Monarch and Nirenburg 1989) or the BBN knowledge-acquisition environment.
The recent growth of interest toward creating on-line lexical resources and their corresponding user interfaces leads to a number of new possibilities for enhancing the productivity of the knowledge-acquisition components in natural language processing systems. The main ways in which knowledge-acquisition productivity can be enhanced is by allowing the knowledge engineer to consult on-line reference materials. Fig. 12.9 illustrates an enhanced knowledge acquisition environment which includes interfaces to machine-readable dictionaries and encyclopedias as well as to textual corpora. Thick lines in the figures represent control flow, thin lines, data flow. The link between the manual lexicon interface and the ontology represents both data and control flow: ontologies are consulted during the acquisition of MTD entries; but the manual lexicon interface is also used for updating the ontology itself. The unshaded barrels in the figure represent static knowledge sources that are used as aids in the acquisition of those static knowledge sources which will be used in actual processing. The use of textual corpora is extremely helpful for determining lexical collocations and selectional restrictions on co-occurrences of various word senses. On-line encyclopedias help in shaping ontologies and domain models. Machine-readable dictionaries, with appropriate search and retrieval facilities, provide a convenient means of deriving candidates for word-sense delimitation and other relevant information for entries of machine-tractable dictionaries.

At present there are no knowledge-acquisition environments for natural language processing which feature all of the above functionalities. It is, however, completely within the state of the art to put together a system of this sort—for instance, by combining ONTOS with an MRD interface such as WordSmith (Neff and Byrd 1987) and a set of corpus-processing routines such as those used by the IBM speech-processing group (Brown et al. 1988).

A major problem of current natural language processing systems is their brittleness, inability to deal with unconstrained input. A large part of this problem is due to the incompleteness of grammars and lexicons—indeed it is to be expected that at any given time in a natural language application which covers a broad domain previously unencountered lexical units and syntactic constructions will appear in input texts. In current systems, when such a situation happens, there are two typical strategies of behaviour. First, the processing system can be built in such a way that it will continue to process the input without taking into account an ‘offending’ input component. This approach enhances the robustness of the system at the expense of potential inaccuracies in the results of its processing. This type of
strategy has, for example, been successfully used in the EPISTLE, CRITIQUE, and PLNLP systems (e.g. Heidorn et al. 1982). Another way of dealing with unexpected input is to evoke specifically designed ‘emergency’ routines when the expected flow of data is broken. A simple example of such a routine may be a spelling correction program, which would, for instance, suggest using the word ‘the’ instead of an unknown and unexpected input string ‘htie’. See Weischedel (1987) for a survey of problems and approaches to treatment of ill-formed input.
One general problem with approaches to treating unexpected input in one of the above ways is that the system never actually learns. Unless its dictionaries and grammars are updated off-line to include the new lexical units and syntactic constructions that led to the necessity of special treatment or had to be excluded completely from the analysis, the system will continue to face exactly the same problems every time a particular phenomenon appears in the input text. A simple way out is to keep track of the causes of ill-formedness in the input and every so often introduce the necessary changes into the grammars and dictionaries for those cases where the ill-formedness is due to the incompleteness of these static knowledge sources. It may be even better to be able to perform this operation on-line, every time a need for this arises. One scenario would, for instance, involve running a newly developed natural language processing system through an extensive period of tests on actual inputs before installing it in its operating environment. During this 'run-in' period the augmentation of the dictionaries and grammars could be performed simultaneously with the system's processing.

It is quite natural to introduce a measure of automatic lexicon acquisition into such an environment. Since at present automatic lexicon-acquisition systems cannot be used separately in practical applications, a mixed automatic/manual environment suggests itself. The idea behind this approach is similar to that of an interactive editor in a machine translation system. Methodologically, this decision means combining a natural language processing system with a knowledge-acquisition system and corpus-processing packages. Figure 12.10 illustrates an integrated processing and acquisition system architecture. The acquisition side of the system has two automatic components—the MRD-to-MTD acquisition module and a set of corpus treatment utilities—and two manual acquisition components—the grammar-acquisition interface and the manual ontology and lexicon-acquisition system.

The system in Fig. 12.10 works as follows. The NLP system obtains a natural language input and processes it until it encounters an unfamiliar syntactic construction or lexical unit. At this point, the on-line acquisition system is called. (In this discussion we disregard the cases of genuinely ill-formed input. We assume the existence of a filter which passes only appropriate queries through to the acquisition system.) This call is represented by the solid thick line in Fig. 12.10. The grammar interface is called for grammar knowledge acquisition and the lexicon interface for acquiring new lexical units. The automatic

---

4 If no unfamiliar constructions or lexical units are found, the on-line acquisition system is not called, and the processing is the same as in a system with the architecture illustrated in Fig. 12.7.
Fig. 12.10 An acquisition system coupled with an NLP system

part of the lexicon acquisition module suggests the format of a new word sense and then attempts to construct as much of the MTD entry as possible through parsing and semantically interpreting the relevant MRD entries. This functionality is supported by the same natural language processing system to which the acquisition system is attached. This ‘recursive’ call to the analyser of the NLP system and the response from it to the acquisition system are represented using the dotted arrows in Fig. 12.10. The MRD-to-MTD module of the lexicon interface serves as the reasoning system in this version of the
NLP system, while the generator is not required for it, since the desired output is, in fact, a symbolic structure and not natural language text. When the automatic MRD-to-MTD interface ends its processing, typically, additional work is needed to produce a complete and correct MRD entry. Therefore, the MRD-to-MTD interface triggers the manual acquisition interface for control and actual incorporation into the lexicon.

During the operation of the reasoning system it can happen that a certain decision cannot be made because of insufficient detail in the ontological model. In this case an option exists of calling the acquisition system to augment the ontology. During the operation of the generator, in cases when it is not possible to eliminate lexical synonymy, a search in a textual corpus can be used in order to determine the preferential lexical selection in a given context by comparing the frequencies of collocations of each of the elements in the set of lexical realization candidates with the lexical selections for units in the sentential context. Queries of this sort can be formulated automatically, but when their results have to be recorded in the MTDs (as fillers of the lexical collocation slots of various word senses) it is necessary, just as in the case of automatic MRD-to-MTD conversion, to go through a human control step.

If we look at the automatic MRD-to-MTD systems in greater detail, we will notice that they are architecturally identical to the kind of the natural language processing systems for which we have been discussing knowledge-acquisition support. Indeed, their inputs are texts of entries in human-oriented MRDs. Their outputs are entries in MTDs. The reasoning component connecting the analyser of MRD entries and the generator of MTD entries is the interactive knowledge-acquisition system as illustrated in Fig. 12.9. This observation suggests a novel design for a combination processing/acquisition system in which processing modules originally designed for one of the processing modes are at least partially reusable in the other. Ideally, one and the same system can be used both for processing and for acquisition.

REFERENCES


— (1987a), 'Deriving Lexical Knowledge Base Entries from Existing Machine-Readable Information Resources', in D. Walker, A. Zampolli, and


