TOWARDS A DICTIONARY SUPPORT ENVIRONMENT FOR REAL TIME PARSING

Hiyan Alshawi, Bran Boguraev, Ted Briscoe

Computer Laboratory, Cambridge University
Corn Exchange Street
Cambridge CB2 3QG, U.K.

ABSTRACT

In this article we describe research on the development of large dictionaries for natural language processing. We detail the development of a dictionary support environment linking a restructrured version of the *Longman Dictionary of Contemporary English* to natural language processing systems. We describe the process of restructuring the information in the dictionary and our use of the Longman grammar code system to construct dictionary entries for the PATR-II parsing system and our use of the Longman word definitions for automated word sense classification.

INTRODUCTION

Recent developments in linguistics, and especially on grammatical theory - for example, Generalised Phrase Structure Grammar (GPSG) (Gazdar et al., In Press), Lexical Functional Grammar (LFG) (Kaplan & Bresnan, 1982) - and on natural language parsing frameworks - for example, Functional Unification Grammar (FUG) (Kay, 1984a), PATR-II (Shieber, 1984) - make it feasible to consider the implementation of efficient systems for the syntactic analysis of substantial fragments of natural language. These developments also demonstrate that if natural language processing systems are to be able to handle the grammatical and logical idiosyncracies of individual lexical items elegantly and efficiently, then the lexicon must be a central component of the parsing system. Real-time parsing imposes stringent requirements on a dictionary support environment; at the very least it must allow frequent and rapid access to the information in the dictionary via the dictionary head words.

The idea of using the machine-readable source of a published dictionary has occurred to a wide range of researchers - for spelling correction, lexical analysis, thesaurus construction, machine-translation, to name but a few applications - very few however have used such a dictionary to support a natural language parsing system. Most of the work on automated dictionaries has concentrated on extracting lexical or other information in, essentially, batch processing (eg. Amsler, 1981; Walker & Amsler, 1983), or on developing dictionary servers for office automation systems (Kay, 1984b). Few parsing systems have substantial lexicons and even those which employ very comprehensive grammars (eg. Robinson, 1982; Bobrow, 1978) consult relatively small lexicons, typically generated by hand. Two exceptions to this generalisation are the Linguistic String Project (Sager, 1981) and the Epistle Project (Heidorn et al., 1982); the former employs a dictionary of less than 10,000 words, most of which are specialist medical terms, the latter has well over 100,000 entries, gathered from machine-readable sources, however, their grammar formalism and the limited grammatical information supplied by the dictionary make this achievement, though impressive, theoretically less interesting.

We chose to employ the *Longman Dictionary of Contemporary English* (Procter 1978, henceforth LDOCE) as the machine-readable source for our dictionary environment because this dictionary has several properties which make it uniquely appropriate for use as the core knowledge base of a natural language processing system. Most prominent among these are the rich grammatical subcategorisations of the 60,000 entries, the large amount of information concerning phrasal verbs, noun compounds and idioms, the individual subject, collocational and semantic codes for the entries and the consistent use of a controlled 'core' vocabulary in defining the words throughout the dictionary. (Michiels (1982) gives further description and discussion of LDOCE from the perspective of natural language processing.)

The problem of utilising LDOCE in natural language processing falls into two areas. Firstly, we must provide a dictionary environment which links the dictionary to our existing natural language processing systems in the appropriate fashion and secondly, we must restructure the information in the dictionary in such a way that these systems are able to utilise it effectively. These two tasks form the subject matter of the next two sections.

THE ACCESS ENVIRONMENT

To link the machine-readable version of LDOCE to existing natural language processing systems we need to provide fast access from Lisp to data held in secondary storage. Furthermore, the complexity of the data structures stored on disc should not be constrained in any way by the method of access, because we have little idea what form the restructured dictionary may eventually take.

Our first task in providing an environment was therefore the creation of a 'lispified' version of the machine-readable LDOCE file. A batch program written in a general editing facility was used to convert the entrire LDOCE typesetting tape into a sequence of Lisp s-expressions without any loss of generality or information. Figure 1 illustrates part of an entry as it appears in the published dictionary, on the typesetting tape and after lispification.

rivet^2 *v* 1 [T1;X9] to cause to fasten with RIVET*s*^{1}:...

```
28289801<R0154300<rivet
28289902<02<   <
28290005<v<
28290107<0100<T1;X9<NAZV<     H   XS
28290208<to cause to fasten with
28290318<{*CA}RIVET{*CB}{*46}s{*44}{*8A}:
.........


((rivet)
  (1 R0154300 !< rivet)
  (2 2 !< !<)
  (5 v !<)
  (7 100 !< T1 !; X9 !< NAZV !< ----H---XS)
  (8 to cause to fasten with
     *CA RIVET *CB *46 s *44 *8A :
      ........))
```

Figure 1

This still leaves the problem of access, from Lisp, to the dictionary entry s-expressions held on secondary storage. Ad hoc solutions, such as sequential scanning of files on disc or extracting subsets of such files which will fit in main memory are not adequate as an efficient interface to a parser. (Exactly the same problem would occur if our natural language systems were implemented in Prolog, since the Prolog 'database facility', refers to the knowledge base that Prolog maintains in main memory.) In principle, given that the dictionary is now in a Lisp-readable format, a powerful virtual memory system might be able to manage access to the internal Lisp structures resulting from reading the entire dictionary; we have, however, adopted an alternative solution as outlined below.

We have implemented an efficient dictionary access system which services requests for s-expression entries made by client Cambridge Lisp programs. The lispified file was sorted and converted into a random access file together with indexing information from which the disc addresses of dictionary entries for words and compounds can be recovered. Standard database indexing techniques were used for this purpose. The current access system is implemented in the programming language C. It runs under UNIX and makes use of the random file access and inter-process communication facilities provided by this operating system. (UNIX is a Trade Mark of Bell Laboratories.) To the Lisp programmer, the creation of a dictionary process and subsequent requests for information from the dictionary appear simply as Lisp function calls.

We have provided for access to the dictionary via head words and the first words of compounds and phrasal verbs, either through the spelling or pronunciation fields. Random selection of dictionary entries is also provided to allow the testing of software on an unbiased sample. This access is sufficient to support our current parsing requirements but could be supplemented with the addition of further indexing files if required. Eventually access to dictionary entries will need to be considerably more intelligent and flexible than a simple left-to-right sequential pass through the lexical items to be parsed, if our processing systems are to make full use of the information concerning compounds and idioms stored in LDOCE.

RESTRUCTURING THE DICTIONARY

The lispified LDOCE file retains the broad structure of the typesetting tape and divides each entry into a number of fields - head word, pronunciation, grammar codes, definitions, examples and so forth. However, each of these fields requires further decoding and restructuring to provide client programs with easy access to the information they require (Calzolari (1984) discusses this need). For this purpose the formatting codes on the typesetting tape are crucial since they provide clues to the correct structure of this information. For example, word senses are largely defined in terms of the 2000 word core vocabulary, however, in some cases other words (themselves defined elsewhere in terms of this vocabulary) are used. These words always appear in small capitals and can therefore be recognised because they will be preceded by a font change control character. In Figure 1 above the definition of "rivet" includes the noun definition of "RIVET^1", as signalled by the font change and the numerical superscript which indicates that it is the noun entry homograph; additional notation exists for word senses within homograhps. On the typesetting tape, font control characters are indicated within curly brackets by hexadecimal numbers. In addition, there is a further complication because this sense is used in the plural and the plural morpheme must be removed before "RIVET" can be associated with a dictionary entry. However, the restructuring program can achieve this because such morphology is always italicised, so the program knows that in the context of non-core vocabulary items the italic font control character signals the occurrence of a morphological variant of a LDOCE head entry.

A suite of programs to unscramble and restructure all the fields in LDOCE entries has been written which is capable of decoding all the fields except those providing cross-reference and usage information for complete homographs. Figure 2 illustrates a simple lexical entry before and after the application of these programs.

```
((pair)
 (1 P0008800 < pair)
 (2 1 < <)
 (3 peER)
 ........
 (7 200 < C9 !, esp !. *46 of < CD-- < ----J---Y)
 (8 *45 a *44 2 things that are alike or of the same
   kind !, and are usu !. used together : *46 a pair of
   shoes !| a beautiful pair of legs *44 *63 compare
   *CA COUPLE *CB *8B *45 b *44 2 playing cards of the
   same value but of different *CA SUIT *CB *46 s *8A
   *44 (3) : *46 a pair of kings)
 (7 300 < GC < ---- < --S-U---Y)
 (8 *45 a *44 2 people closely connected : *46 a pair
   of dancers *45 b *CA COUPLE *CB *8B *44 (2)
   (esp !. in the phr !. *45 the happy pair *44) *45 c
   *46 sl *44 2 people closely connected who cause
   annoyance or displeasure : *46 You!'re a fine pair
   coming as late as this !!)
  ........)


(Word-sense (Number 2)
   ((Sub-definition
     (Item a) (Label NIL)
     (Definition 2 things that are alike or of the same
       kind !, and are usually used together)
     ((Example NIL (a pair of shoes))
       (Example NIL (a beautiful pair of legs)))
     (Cross-reference
      compare-with
       (Ldoce-entry (Lexical COUPLE)
         (Morphology NIL )
         (Homograph-number 2)
         (Word-sense-number NIL)))
    (Sub-definition
      (Item b) (Label NIL )
      (Definition 2 playing cards of the same value
        but of different
         (Ldoce-entry (SUIT)
           (Morphology s)
           (Homograph-number 1)
           (Word-sense-number 3))
      ((Example NIL (a pair of kings))))))
(Word-sense (Number 3)
   ((Sub-definition
     (Item a) (Label NIL)
     (Definition 2 people closely connected)
     ((Example NIL (a pair of dancers))))
    (Sub-definition
      (Item b) (Label NIL)
      (Definition
        (Ldoce-entry (Lexical COUPLE )
          (Morphology NIL)
          (Homograph-number 2)
          (Word-sense-number 2))
        (Gloss: especially in the phrase the happy pair )))
    (Sub-definition
      (Item c) (Label slang)
      (Definition 2 people closely connected who
        cause annoyance or displeasure)
      ((Example NIL
        (You!'re a fine pair coming as late as this!))))))
```

Figure 2

The development of the restructuring programs is a non-trivial task because the organisation of information on the typesetting tape presupposes its visual presentation, and the ability of human users to apply common sense, utilise basic morphological knowledge, ignore minor notational inconsistencies, and so forth. To provide a test-bed for these programs we have implemented an interactive dictionary browser capable of displaying the restructured information in a variety of ways and representing it in perspicuous and expanded form.

To illustrate the problems involved in the restructuring process we will discuss the restructuring of the grammar codes in some detail, however, the reader should bear in mind that this represents only one comparatively constrained field of an LDOCE entry and therefore, a small proportion of the overall restructuring task. Figure 3 illlustrates the grammar code field for the third word sense of the verb "believe" as it appears in the published dictionary, on the typesetting tape and after restructuring.

Multiple grammar codes are elided and abbreviated in the dictionary to save space and restructuring must reconstruct the full set of codes. This can be done with knowledge of the syntax of the grammar code system and the significance of punctuation and font changes. For example, semicolons indicate concatenated codes and commas indicate concatenated, elided codes. However, discovering the syntax of the system is difficult since no explicit description is available from Longman and the code is geared more towards visual presentation than formal precision; for example, words which qualify codes, such as "to be" in Figure 3, appear in italics and therefore, will be preceded by the font control character '*45'. But sometimes the thin space

```
believe v 3         [T5a,b;V3;X (to be) 1, (to be) 7]

(7 300 !< T5a !, b !; V3 !; X (*46 to be *44)
 1 !, (*46 to be *44) 7 !< ........)

word sense 3
                    head: X7x
                    head: X1x
                    head: V3
                    head: T5a
                    head: T5b
```

Figure 3

control character '*64' also appears; the insertion of this code is based solely on visual criteria, rather than the informational structure of the dictionary. Similarly, choice of font can be varied for reasons of appearance and occasionally information normally associated with one field of an entry is shifted into another to create a more compact or elegant printed entry. In addition to the 'noise' generated by the fact that we are working with a typesetting tape geared to visual presentation, rather than a database, there are errors in the use of the grammar code system; for example, Figure 4 illustrates the code for the first sense of the noun "promise".

```
promise n 1       [C (of),C3,5; under + U]
```

Figure 4

The occurrence of the full code "C3" between commas is incorrect because commas are clearly intended to delimit sequences of elided codes. This type of error arises because grammatical codes are constructed by hand and no automatic checking procedure is attempted (see Michiels, 1982). Finally, there are errors or omissions in the use of the codes; for example, Figure 5 illustrates the grammar codes for the listed senses of the verb "upset".

```
upset:
    for cat = v
          word sense 1    head T1
          word sense 2    head I
          word sense 3    head T1
          word sense 4    head T1
```

Figure 5

These codes correspond to the simple transitive and intransitive uses of "upset"; no codes are given for the uses of "upset" with sentential complements. Clearly, the restructuring programs cannot correct this last type of error, however, we have developed a system which is sufficiently robust to handle the other problems described above. Rather than apply these programs to the dictionary and create a new restructured file, they are applied on a demand basis, as required by the dictionary browser or the other client programs described in the next section; this allows us to continue to refine the restructuring programs incrementally as further problems emerge.

USING THE DICTIONARY

Once the information in LDOCE has been restructured into a format suitable for accessing by client programs, it still remains to be shown that this information is of use to our natural language processing systems. In this section, we describe the use that we have made of the grammar codes and word sense definitions.

Grammar codes

The grammar code system used in LDOCE is based quite closely on the descriptive grammatical framework of Quirk et al. (1972). The codes are doubly articulated; capital letters represent the grammatical relations which hold between a verb and its arguments and numbers represent subcategorisation frames which a verb can appear in. (The small letters which appear with some codes represent a variety of less important information, for example, whether a sentential complement will take an obligatory or optional complementiser.) Most of the subcategorisation frames are specified by syntactic category, but some are very ill-specified; for instance, 9 is defined as "needs a descriptive word or phrase". In practice anything functioning as an adverbial will satisfy this code, when attached to a verb. The criteria for assignment of capital letters to verbs is not made explicit, but is influenced by the syntactic and semantic relations which hold between the verb and its arguments; for example, I5, L5 and T5 can all be assigned to verbs which take a NP subject and a sentential complement, but I5 will only be assigned if there is a fairly close semantic link between the two arguments and T5 will be used in preference to I5 if the verb is felt to be semantically two place rather than one place, such as "know" versus "appear". On the other hand, both "believe" and "promise" are assigned V3 which means they take a NP object and infinitival complement, yet there is a similar semantic distinction to be made between the two verbs; so the criteria for the assignment of the V code seem to be syntactic.

The parsing systems we are interested in all employ grammars which carefully distinguish syntactic and semantic information of this kind, therefore, if the information provided by the Longman grammar code system is to be of use we need to be able to separate out this information and map it into the representation scheme used for lexical entries used by one of these parsing systems. To demonstrate that this is possible we have implemented a system which constructs dictionary entries for the PATR-II system (Shieber, 1984 and references therein). PATR-II was chosen because the system has been reimplemented in Cambridge and was therefore, available; however, the task would be nearly identical if we were constructing entries for a system based on GPSG, FUG or LFG.

The PATR-II parsing system operates by unifying directed graphs (DGs); the completed parse for a sentence will be the result of successively unifying the DGs associated with the words and constituents of the sentence according to the rules of the grammar. The DG for a lexical item is constructed from its lexical entry which will consist of a set of templates for each syntactically distinct variant. Templates are themselves abbreviations for unifications which define the DG. For example, the basic entry and associated DG for the verb "storm" are illustrated in Figure 6.

```
word storm:
    word sense ⇒ <head trans sense-no> = 1
                 V Takes NP Dyadic


worddag storm:
    [cat:  v
     head: [aux: false
            trans: [pred:  storm
                    sense-no: 1
                    arg1:  <DG15> = []
                    arg2:  <DG16> = []]]
    syncat: [first : [cat:  NP
                    head: [trans: <DG15>]]
             rest: [first: [cat: NP
                             head: [trans: <DG16>]]
                    rest: [first: lambda]]]]
```

Figure 6

The template Dyadic defines the way in which the syntactic arguments to the verb contribute to the logical structure of the sentence; thus, the information that "storm" is transitive and that it is logically a two-place predicate is kept distinct. Consequently, the system can represent the fact that some verbs which take two syntactic arguments are nevertheless logically one-place predicates.

It is not possible to automatically construct PATR-II dictionary entries for verbs just by mapping one full grammar code from the restructured LDOCE entry into a set of templates. However, it turns out that if we compare the full set of grammar codes associated with a particular sense of a verb, following a suggestion of Michiels (1982), then we can construct the correct set of templates. That is, we can extract all the information that PATR-II requires concerning the subcategorisation and semantic type of verbs. For example, as we saw above, "believe" under one sense is assigned the codes T5 and V3; the presence of the T5 code tells us that "believe" is a 'raising-to-object' verb and logically two-place under the V3 interpretation. On the other hand, "persuade" is only assigned the V3 code, so we can conclude that it is three-place with object control of the infinitive. By systematically exploiting the collocation of different codes in the same field, it is possible to distinguish the raising, equi and control properties of verbs. In effect, we are utilising what was seen as the transformational consequences of the semantic type of the verb within classical generative grammar.

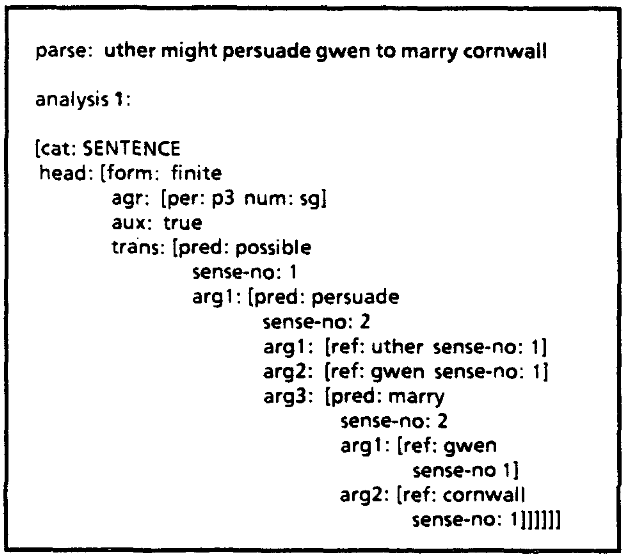
```
word marry:
    word sense ⇒ <head trans sense-no> = 1
                 V Takes NP Dyadic
    word sense ⇒ <head trans sense-no> = 1
                 V TakesIntransNP Monadic
    word sense ⇒ <head trans sense-no> = 2
                 V TakesNP Dyadic
    word sense ⇒ <head trans sense-no> = 3
                 V TakesNPPP Triadic

word persuade:
    word sense ⇒ <head trans sense-no> = 1
                 V Takes NP Dyadic
    word sense ⇒ <head trans sense-no> = 1
                 V TakesNPSbar Triadic
    word sense ⇒ <head trans sense-no> = 2
                 V TakesNP Dyadic
    word sense ⇒ <head trans sense-no> = 2
                 V TakesNPInf ObjectControl Triadic
```

Figure 7

The modified version of PATR-II that we have implemented contains a small dictionary and constructs entries automatically from restructured LDOCE entries for most verbs that it encounters. As well as carrying over the grammar codes, PATR-II has been modified to represent the word sense numbers which particular grammar codes are associated with. Thus, the analysis of a sentence by the PATR-II system now represents its syntactic and logical structure and the particular senses of the words (as defined in LDOCE) which are relevant in the grammatical context. Figure 7 illustrates the dictionary entries for "marry" and "persuade" constructed by the system from LDOCE.

In Figure 8 we show one of the two analyses produced by PATR-II for a sentence containing these two verbs. The other analysis is syntactically and logically identical but incorporates sense two of "marry". Thus, the system knows that further semantic analysis need only consider sense two of "persuade" and sense one and two of "marry"; this rules out one further sense of each, as defined in LDOCE.



```
parse:  uther might persuade gwen to marry cornwall

analysis 1:

[cat: SENTENCE
 head: [form:  finite
        agr:  [per: p3  num: sg]
        aux:  true
        trans: [pred: possible
                sense-no: 1
                arg1: [pred: persuade
                       sense-no: 2
                       arg1:  [ref: uther  sense-no: 1]
                       arg2:  [ref: gwen  sense-no: 1]
                       arg3:  [pred: marry
                               sense-no: 2
                               arg1: [ref: gwen
                                      sense-no 1]
                               arg2: [ref: cornwall
                                      sense-no: 1]]]]]]
```

Figure 8

Word sense definitions

The automatic analysis of the definition texts of LDOCE entries is aimed at making the semantic information on word senses encoded in these definitions available to natural language processing systems. LDOCE is particularly suitable to such an endeavour because of the 2000 word restricted definition vocabulary, and in fact only 'central' senses of the words in this restricted vocabulary occur in definition texts. It is thus possible to process the LDOCE definition of a word sense in order to produce some representation of the sense definition in terms of senses of words in the restricted vocabulary. This representation could then be combined, for the benefit of the client language processing system, with the other semantic information encoded for word senses in LDOCE; in particular the 'box codes' that give simple selectional restrictions and the 'subject codes' that classify senses according to subject area usage. (These are not in the published version of the dictionary, but are available on the tape.)

There are various possibilities for the form of the output resulting from processing a definition. The current experimental system produces output that is convenient for incorporating new word senses into a knowledge base organized around classification hierarchies, as discussed shortly. However, the system allows the form of output structures to be specified in a flexible way. Alternative possible output representations would be meaning postulates and definitions based on semantic primitives.

As mentioned above, the implemented experimental system is intended to enable the classification (see e.g. Schmolze, 1983) of new word senses with respect to a hierarchically organized knowledge base, for example the one described in Alshawi (1983). The proposal being made here is that the analysis of dictionary definitions can provide enough information to link a new word sense to domain knowledge already encoded in the knowledge base of a limited domain natural language application such as a database query system. Given a hand-coded hierarchical organization of the relevant (central) senses of the definition vocabulary together with a classification of the relationships between these senses and domain specific concepts, the LDOCE definition of a new word sense often contains enough information to enable the inclusion of the word sense in this classification, and hence allow the new word to be handled correctly when performing the application task.

The information necessary for this process is present, in the case of nouns, as restrictions on the classes which subsume the new type of object, its properties, and predications often expressed by relative clauses. There are also a number of more specific predications (such as "purpose" in the example given below) that are very common in dictionary definitions, and have immediate utility for the classification of the relationships between word senses. Similarly, the information relevant to the classification of verb and adjective senses present in sense definitions includes the classes of predicates that subsume the new predicate corresponding to the word sense, restrictions on the arguments of this predicate, and words indicating opposites as is frequently the case with adjective definitions.

Figure 9 below shows the output produced by the implemented definition analyser for lispified LDOCE definitions of one of the noun senses and one of the verb senses of the word "launch". It should be emphasized that the output produced is not regarded as a formal language, but rather as an intermediate data structure containing information relevant to the classification process.

```
(launch)

(a large usu . motor-driven boat used for carrying people
 on rivers , lakes , harbours , etc .)

((CLASS BOAT) (PROPERTIES (LARGE))
  (PURPOSE
    (PREDICATION (CLASS CARRY) (OBJECT PEOPLE))))


(to send (a modern weapon or instrument) into the sky or
 space by means of scientific explosive apparatus)

((CLASS SEND)
  (OBJECT
    ((CLASS INSTRUMENT) (OTHER-CLASSES (WEAPON))
      (PROPERTIES (MODERN))))
  (ADVERBIAL ((CASE INTO) (FILLER (CLASS SKY)))))
```

Figure 9

The analysis process is intended to extract the most important information from definitions without necessarily having to produce a complete analysis of the whole of a particular definition text since attempting to produce complete analyses would be difficult for many LDOCE definition texts. In fact the current definition analyser applies successively more specific phrasal analysis patterns; more detailed analyses being possible when relatively specific phrasal patterns are applied successfully to a definition. A description of the details of this analysis mechanism is beyond the scope of the present paper. Currently, around fifty phrasal patterns are used altogether for noun, verb, and adjective definitions. A major difficulty encountered so far in this work stems from the liberal use in LDOCE definitions of derivational morphology and phrasal verbs which greatly expands the effective definition vocabulary.

CONCLUSION

The research reported in this paper demonstrates that it is both possible and useful to restructure the information contained in LDOCE for use in natural language processing systems. Most applications for natural language processing systems will require vocabularies substantially larger than those typically developed for theoretical or demonstration purposes and it is often not practical, and certainly never desirable, to generate these by hand. The use of machine-readable sources of published dictionaries represents a practical and feasible alternative to hand generation.

Clearly, there is much more work to be done with LDOCE in the extension of the use of grammar codes and the improvement of the word sense classification system. Similarly, there is a considerable amount of information in LDOCE which we have not attempted to exploit as yet; for example, the box codes, which contain selection restrictions for verbs or the subject codes, which classify word senses according to the Merriam-Webster codes for subject matter (see Walker & Amsler (1983) for a suggested use for these). The large amount of semi-formalised information concerning the interpretation of noun compounds and idioms also represents a rich and potentially very useful source of information for natural language processing systems. In particular, we intend to investigate the automatic generation of phrasal analysis rules from the information on idiomatic word usage.

In the longer term, it is clear that no existing published dictionary can meet all the requirements of a natural language processing system and a substantial component of the research reported above has been devoted to restructuring LDOCE to make it more suitable for automatic analysis. This suggests that the automatic construction of dictionaries from published sources intended for other purposes will have a limited life unless lexicography is heavily influenced by the requirements of automated natural language analysis. In the longer term, therefore, the automatic construction of dictionaries for natural language processing systems may need to be based on techniques for the automatic analysis of large corpora (eg. Leech et al., 1983). However, in the short term, the approach outlined in this paper will allow us to produce a sophisticated and useful dictionary rapidly.

ACKNOWLEDGEMENTS

We would like to thank the Longman Group Limited for kindly allowing us access to the LDOCE typesetting tape for research purposes. We also thank Karen Sparck Jones and John Tait for their comments on the first draft, which substantially improved this paper. We are very grateful to the SERC for funding this research.

REFERENCES

Alshawi, H.(1983) *Memory and Context Mechanisms for Automatic Text Processing*, PhD Thesis, Technical Report 60, University Computer Laboratory, Cambridge

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

Bobrow, R.(1978) *The RUS System*, BBN Report 3878, Bolt, Beranek and Newman Inc., Cambridge, Mass

Calzolari, N.(1984) 'Machine-Readable Dictionaries, Lexical Data Bases and the Lexical System', *Proceedings of the 10th International Congress on Computational Linguistics, Stanford*, CA, pp.460-461

Gazdar, G., Klein, E., Pullum, G. and Sag, I.(In press) *Generalised Phrase Structure Grammar*, Blackwell, Oxford

Heidorn, G. et al.(1982) 'The EPISTLE text-critiquing system', *IBM Systems Journal, vol.21*, 305-326

Kaplan, R. and Bresnan, J.(1982) 'Lexical-Functional Grammar: A Formal System for Grammatical Representation' in J.Bresnan (éd.), *The Mental Representation of Grammatical Relations*, The MIT Press, Cambridge, Mass, pp.173-281

Kay, M.(1984a) 'Functional Unification Grammar: A Formalism for Machine Translation', *Proceedings of the 10th International Congress on Computational Linguistics*, Stanford, CA, pp.75-79

Kay, M.(1984b) 'The Dictionary Server', *Proceedings of the 10th International Congress on Computational Linguistics*, Stanford, California, pp.461-462

Leech, G., Garside, R. and Atwell, E.(1983), *The Automatic Grammatical Tagging of the LOB Corpus*, Bulletin of the International Computer Archive of Modern English, Norwegian Computing Centre for the Humanities, Bergen

Michiels, A.(1982) *Exploiting a Large Dictionary Data Base*, PhD Thesis, Université de Liège, Liège

Procter, P.(1978) *Longman Dictionary of Contemporary English*, Longman Group Limited, Harlow and London

Quirk, R. et al.(1972) *A Grammar of Contemporary English*, Longman Group Limited, Harlow and London

Robinson, J.(1982) 'DIAGRAM: A Grammar for Dialogues', *Communications of the ACM, vol.25*, 27-47

Sager, N.(1981) *Natural Language Information Processing*, Addison-Wesley, Reading, Mass

Shieber, S.(1984) 'The Design of a Computer Language for Linguistic Information', *Proceedings of the 10th International Congress on Computational Linguistics*, Stanford, CA, pp.362-366

Schmolze, J.G., and Lipkis, T.A.(1983) 'Classification in the KL-ONE Knowledge Representation System', *Proceedings, IJCAI-83, Karlsruhe*, pp.330-332

Walker, D. and Amsler, A.(1983) *The Use of Machine-Readable Dictionaries in Sublanguage Analysis*, SRI International Technical Note, Menlo Park, CA