The Derivation of a Grammatically Indexed Lexicon from the Longman Dictionary of Contemporary English

Bran Boguraev[†], Ted Briscoe[§], John Carroll[†], David Carter[†] and Claire Grover[§]

† Computer Laboratory, University of Cambridge
Corn Exchange Street, Cambridge CB2 3QG, England

§ Department of Linguistics, University of Lancaster
Bailrigg, Lancaster LA1 4YT, England

Abstract

We describe a methodology and associated software system for the construction of a large lexicon from an existing machine-readable (published) dictionary. The lexicon serves as a component of an English morphological and syntactic analyser and contains entries with grammatical definitions compatible with the word and sentence grammar employed by the analyser. We describe a software system with two integrated components. One of these is capable of extracting syntactically rich, theory-neutral lexical templates from a suitable machine-readable source. The second supports interactive and semi-automatic generation and testing of target lexical entries in order to derive a sizeable, accurate and consistent lexicon from the source dictionary which contains partial (and occasionally inaccurate) information. Finally, we evaluate the utility of the *Longman Dictionary of Contemporary English* as a suitable source dictionary for the target lexicon.

1 Introduction

Within the larger framework of the Alvey Programme of advanced information technology — a research and development initiative set up in the UK to promote collaborative research projects aimed at several enabling key technologies — a coordinated effort to build a natural language toolkit for the use by the wider academic and industrial community is being carried out jointly by groups at the Universities of Cambridge, Lancaster and Edinburgh.

The goal of these three closely related projects is to produce directly compatible rule systems and associated software, capable of functioning together as an integrated system for morphological and syntactic parsing of texts. The projects aim to deliver, respectively, a *sentence grammar* of English together with a *word list* indexed to the grammar, a combined inflectional and derivational *morphological analyser* and *dictionary system*, and a *parser* for the grammatical formalism used. The work is being carried out within the theoretical framework of Generalized Phrase Structure Grammar (Gazdar et al., 1985), but many of the mechanisms would be usable without a theoretical commitment to GPSG. It is envisaged that the complete integrated toolkit will be used by a number of research and development groups, as a base component for a range of applications. The potential requirements of a diverse user community motivate, in particular, the need for a morphological and syntactic analyser with wide coverage of English grammar and vocabulary. Briscoe et al. (1987) describes the sentence grammar formalism and current coverage of the English grammar in detail. Russell et al. (1986) describes the morphological analyser and dictionary system. Further relevant details of both projects are provided in section 2.

As part of the grammar project, in tandem with the development of the grammar proper, work is underway to develop a sizeable word list which will be integrated with an existing lexicon of about 4000 words, hand crafted by the morphology project. The coverage of this word list and its compatibility with the sentence grammar, word grammar and existing lexicon is critical for the complete analysis system. The word list need only contain base and irregular entries, as productive inflectional and derivational variants are analysed at run-time on the basis of the word grammar. Therefore, when the word list is integrated with the existing lexicon and dictionary system it will form a dynamic system for word analysis, and not just a repository of word forms used for simple lookup.

An additional constraint on the content of the target word list comes from the fact that even though there is no provision for the analysis system to handle semantics, there is still the need to provide a minimal, theoretically neutral extension to the grammar rules and lexical entry format to allow subsequent integration of a semantic component: thus information concerning eg. the predicate-argument structure of verbs and their logical types must be made available in the lexical entries.

The question then arises of how to develop such a detailed and substantial word list. Our approach has been to make use of the machine-readable source of a published dictionary, namely the *Longman Dictionary of Contemporary English* (henceforth LDOCE) (Procter, 1978). Apart from the obvious motivation of attempting to derive a large list of words from a computerised source, LDOCE is particularly relevant to this project since it offers, among other things, through a highly elaborate and semi-formal system of *grammar codes*, detailed information about the grammatical behaviour of individual words. We have mounted the dictionary on-line and, following its conversion into a flexible lexical knowledge base (as described in Boguraev et al., 1987), a range of experiments have since been carried out with the aim of establishing LDOCE's appropriateness to the task of deriving a word list with associated grammatical definitions indexed to the analyser grammar. Section 3 below describes the syntactic level information available in, and extractable from, LDOCE and summarises the description of an operational program used to derive such information.

The attempt to use semi-formalised, and occasionally inaccurate, information for constructing a large computerised lexicon raises a number of practical problems. In order to make maximal use of the rich syntactic data in the source machine-readable dictionary (MRD), we have designed a lexicon development system which embodies a methodology for a semi-automatic interactive cycle of lexical entry generation and testing. This is described in section 4.

2 The target lexicon

Given the goal of the toolkit projects to provide a lexicon capable of supporting morphological and syntactic analysis of English, there is a precise definition of the information required in lexical entries. Both the grammar and morphology projects have adopted a feature system based largely on that described in Gazdar et al. (1985). A lexical entry will contain features relevant either to the word grammar or sentence grammar, or both, represented as a list of feature name / feature value pairs. In Figure 1 we show a fragment from the hand crafted lexicon developed as part of the morphology project (Russell et al., 1986). Here we concentrate on the feature-value sets carrying the syntactic information; the complete entries have also semantic and user fields, which are of no relevance to this paper.

```
believe
 [V +, N -, BAR 0, AGR [BAR 2, V -, N +,
    NFORM NORM], PRD -, NEG -, WORD +, AUX -,
    INFL +, FIN -, VFORM BSE, LAT -, SUBCAT OR]

 [V +, N -, BAR 0, AGR [BAR 2, V -, N +,
    NFORM NORM], PRD -, NEG -, WORD +, AUX -,
    INFL +, FIN -, VFORM BSE, LAT -, SUBCAT TWONP]

 [V +, N -, BAR 0, AGR [BAR 2, V -, N +,
    NFORM NORM], PRD -, NEG -, WORD +, AUX -,
    INFL +, FIN -, VFORM BSE, LAT -, SUBCAT NP_AP]

 [V +, N -, BAR 0, AGR [BAR 2, V -, N +,
    NFORM NORM], PRD -, NEG -, WORD +, AUX -,
    INFL +, FIN -, VFORM BSE, LAT -, SUBCAT SFIN]
```

Figure 1: Sample lexical entries

An almost complete list of the feature names and potential values which may occur as part of the lexical entry for a given morpheme is given in Figure 2 overleaf. Grover et al. (1987) contains a complete description of the features used in the sentence grammar; Ritchie et al. (1987) offers an equally complete description of the morphological and syntactic features relevant to the operations of the word grammar. For the purposes of this paper, we present a brief overview of the sentence grammar feature system.

With exception of the features N, V and BAR, used to define the major categories of the grammar, most features can be classified in terms of the categories they apply to. For each major category type there is a set of head features which must appear on all instances of that category type, regardless of their BAR feature value. Further features must (or may) be associated only with some instances of a category type, depending on the value of their BAR feature (or, on occasions, some other feature). The sets of head features for the four major categories are:

VERBALHEAD {PRD FIN AUX VFORM PAST AGR}

NOMINALHEAD {PLU POSS CASE PN COUNT PRD PRO PART NFORM PER}

PREPHEAD {PFORM LOC PRD}

ADJHEAD {AFORM PRD QUA ADV NUM NEG PART AGR DEF}.

The features appearing on certain categories in addition to the sets defined above are COMP, INV, NEG and SUBCAT which are relevant to verbal categories; SPEC, DEF and SUBCAT, applicable to nominal categories; GERUND, POSS and SUBCAT for prepositional categories; and SUBCAT alone for adjectival categories. With exception of SUBCAT, which must be specified for all lexical entries, and the respective head features sets, the only other features required by the lexical nodes in the grammar are NEG, INV and DEF. Features like SLASH, WH, UB and EVER, which are required by the grammar to implement the GPSG treatment of certain linguistic phenomena, are of no relevance to this paper.

The feature set in Figure 2 overleaf defines the information about lexical items which will be required to construct a lexicon compatible both in form and content with the rest of the analysis system. Some of these features, (such as FIX) are specific to bound morphemes (these include, for example, entries for "ative", "ing" or "ness"). Other features (for instance WH, REFL) are specific to closed class vocabulary items, such as interrogative, relative and reflexive pronouns. Bound morphemes and closed class vocabulary are exhaustively defined in the hand crafted lexicon. However, this lexicon inevitably only contains a few examples of the much larger open class vocabulary. In order for the word and sentence grammars to function correctly, open class vocabulary must be defined in terms of the feature set illustrated overleaf (Figure 2a).

The features relevant to the open class vocabulary can be divided into those which are predictable on the basis of the part of speech of the item involved, those which follow from the inflectional or derivational morphological rules incorporated into the system, and those which rely on more specific information than part of speech, but nevertheless must be specified for each individual entry. For example the values for the features N, V and BAR in the sample entries above follow from the part of speech of "believe". The values of PLU and PER are predictable on the basis of the word grammar rules and need not be independently specified for each entry. On the other hand, the values of SUBCAT and LAT are not predictable from either part of speech or general morphological information.

We concentrate on this last class of features which must be specified on an entry-by-entry basis in any lexicon which is going to be adequate for supporting the analysis system. Within this class of features some (eg. LAT, AT or BARE_ADJ) are only relevant to the word grammar. It is clear that those features that are derivable from the part of speech information are recoverable from virtually any MRD. However, most (if not all) of the features in the third class above are not recoverable from the majority of MRDs. As indicated above, LDOCE appears to be an exception to this generalisation, because it employs a system of grammatical tagging of major syntactic classes, offering detailed information about subcategorisation, morphological irregularity and broad syntactico-semantic information.

a. open class vocabulary

BAR	{-1 0 1 2}
V	{- +}
N	{- +}
PRD	{- +}
QUA	{- +}
ADV	{- +}
FIN	{- +}
PAST	{- +}
PLU	{- +}
AT	{- +}
LAT	{- +}
AGR	a category
STEM	a category
SUBCAT	{........PRED INF NP AP NOPASS SFIN VPINF SINF OR IT_SUBJ PPFROM PPTO TWONP FOR_S LOC S_SUBJ NP_NP NP_AP OE SR1 DETN AND ....}
INFL	{- +}
COUNT	{- +}
PN	{- +}
PER	{1 2 3}
CASE	{NOM ACC}
BARE_ADJ	{- +}
AFORM	{ER EST NONE}
NFORM	{IT THERE NORM}
VFORM	{BSE EN ING TO}

b. closed class vocabulary and affixes

FIX	{PRE SUF}
INV	{- +}
AUX	{- +}
NEG	{- +}
DEF	{- +}
SLASH	a category
COMPOUND	{NOUN VERB ADJ NOT}
TITLE	{- +}
POSS	{- +}
PFORM	{WITH OF FROM AT ABOUT TO ON IN FOR AGAINST BY}
REFL	a category
WH	{- +}
UB	{Q R}
EVER	{- +}
PRO	{- +}
PRT	{AS IN OFF ON UP}

Figure 2: Features and feature values

3 The source data

It turns out that even though the grammar coding system of LDOCE is not GPSG specific, it encodes much of the information which GPSG requires relating to the subcategorisation classes in the lexicon. The Longman lexicographers have developed a representational system which is capable of describing compactly a variety of data relevant to the task of building a lexicon with grammatical definitions; in particular, they are capable of denoting distinctions between count and mass nouns ("dog" vs. "desire"), predicative, postpositive and attributive adjectives ("asleep" vs. "elect" vs. "jocular"), noun and adjective complementation ("fondness", "fact") and, most importantly, verb complementation and valency.

3.1 The Longman grammar coding system

Grammar codes typically contain a capital letter , followed by a number and, occasionally, a small letter, for example [T5a] or [V3]. The capital letters encode information "about the way a word works in a sentence or about the position it can fill" (Procter, 1978: xxviii); the numbers "give information about the way the rest of a phrase or clause is made up in relation to the word described" (ibid.). For example, "T" denotes a transitive verb with one object, while "5" specifies that what follows the verb must be a *that* clause. (The small letters, eg. "a" in the case above, provide information related to the status of various complementisers, adverbs and prepositions in compound verb constructions: here it indicates that the complementiser is optional.) As another example, "V3" introduces a verb followed by one object and a verb form (V) which must be an infinitive with *to* (3).

In addition, codes can be qualified with words or phrases which provide further information concerning the linguistic context in which the described item is likely, and able, to occur; for example [D1(*to*)] or [L(*to be*)1]. Sets of codes, separated by semicolons, are associated with individual word senses in the lexical entry for a particular item, as the entry for "feel", with extracts from its printed form shown in Figure 3, illustrates. These sets are elided and abbreviated in the code field associated with the word sense to save space in the dictionary. Partial codes sharing an initial letter can be separated by commas, for example [T1,5a]. Word qualifiers relating to a complete sequence of codes can occur at the end of a code field, delimited by a colon, for example [T1;I0: (DOWN)].

feel[1] *v* 1 [T1,6] to get the knowledge of by touching with the fingers: ... 2 [Wv6;T1] to experience (the touch or movement of something): ... 3 [L7] to experience (a condition of the mind or body); be consciously: ... 4 [L1] to seem to oneself to be: ... 5 [T1,5;V3] to believe, esp. for the moment 6 [L7] to give (a sensation): ... 7 [Wv6;I0] to (be able to) experience sensations: ... 8 [Wv6;T1] to suffer because of (a state or event): ... 9 [L9 (*after*, *for*)] to search with the fingers rather than with the eyes: ...

Figure 3: Fragment of an LDOCE entry

This apparent formal syntax for describing grammatical information in a compact form occasionally breaks down: different classes of error occur in the tagging of word senses. These include, for example, misplaced commas or colon delimiters and occasional migration of other lexical information (e.g. usage labels) into the grammar code fields.

This type of error and inconsistency arises because grammar codes are constructed by hand and no automatic checking procedure is attempted (Michiels, 1982). They provide much of the motivation for our interactive approach to lexicon development, since any attempt at batch processing without extensive user intervention would inevitably result in an incomplete and inaccurate lexicon.

3.2 Making use of the grammar codes

The program which transforms the LDOCE grammar codes into lexical entries utilisable by the analyser first produces a relatively theory-neutral representation of the lexical entry for a particular word. As an illustration of the process of transforming a dictionary entry into a lexical template we show below the mapping of the third verb sense of "believe" below into a lexical entry incorporating information about the grammatical category, syntactic subcategorisation frames and semantic type of verb — for example a label like **(Type 2 ORaising)** indicates that under the given sense the verb is a two-place predicate and that if it occurs with a syntactic direct object, this will function as the logical subject of the predicate complement.

be-lieve ... v 1 [I0] to have a firm religious faith 2 [T1] to consider to be true or honest: *to believe someone|to believe someone's reports* 3 [T5a,b;V3;X (*to be*) 1, (*to be*) 7] to hold as an opinion; suppose: *I believe he has come.* | *He has come, I believe.* | *"Has he come?" "I believe so."* | *I believe him to have done it.* | *I believe him (to be) honest*
....

```
(believe verb (Sense 3)
  ((Takes NP SBar) (Type 2))
  ((Takes NP NP Inf) (Type 2 ORaising))
  ((or ((Takes NP NP NP) (Type 2 ORaising))
   ((Takes NP NP AuxInf) (Type 2 ORaising))
  ((or ((Takes NP NP AP) (Type 2 ORaising))
   ((Takes NP NP AuxInf) (Type2 ORaising))
```

Figure 4: A lexical template derived from LDOCE

This resulting structure is a lexical template, designed as a formal representation for the kind of syntactico-semantic information which can be extracted from the dictionary and which is relevant to a system for automatic morphological and syntactic analysis of English texts.

The overall transformation strategy employed by our system attempts to derive both subcategorisation frames relevant to a particular word sense and information about the semantic nature (i.e. the predicate-argument structure and the logical type) of, especially, verbs. In the main, the code numbers determine a unique subcategorisation. However, such semantic information is not explicitly encoded in the LDOCE grammar codes, so we have adopted an approach attempting to deduce a semantic classification of the particular sense of the verb under consideration on the basis of the complete set of codes assigned to that sense. In any subcategorisation frame which involves a predicate complement there will be a non-transparent relationship between the superficial syntactic form and the underlying logical relations in the sentence. In these situations the parser can use the semantic type of the verb to compute this relationship. Expanding on a suggestion of Michiels (1982), we classify verbs as subject equi (**SEqui**), object equi (**OEqui**), subject raising (**SRaising**) or object raising (**ORaising**) for each sense which has a predicate complement code associated with it. These terms, which derive from Transformational Grammar, are used as convenient labels for what we regard as a semantic distinction.

The five rules which are applied to the grammar codes associated with a verb sense are ordered in a way which reflects the filtering of the verb sense through a series of syntactic tests. Verb senses with an [*it*+I5] code are classified as **SRaising**. Next, verb senses which contain a [V] or [X] code and one of [D5], [D5a], [D6] or [D6a] codes are classified as **OEqui**. Then, verb senses which contain a [V] or [X] code and a [T5] or [T5a] code in the associated grammar code field, (but none of the D codes mentioned above), are classified as **ORaising**. Verb senses with a [V] or [X(*to be*)] code, (but no [T5] or [T5a] codes), are classified as **OEqui**. Finally, verb senses containing a [T2], [T3] or [T4] code, or an [I2], [I3] or [I4] code are classified as **SEqui**. Below we give examples of each type; for a detailed description see Boguraev and Briscoe (1987).

happen(3)	[Wv5;*it*+I5] (Type 1 SRaising)
warn(1)	[Wv4;I0;T1:(*of,against*),5a;D5a;V3] (Type 3 OEqui)
assume(1)	[Wv4;T1,5a,b;X(*to be*)1,7] (Type 2 ORaising)
decline(3)	[T1,3;I0] (Type 2 SEqui)

Figure 5: The four semantic types of verb

A generic lexical template of the form illustrated in Figure 4 can clearly be directly mapped into a feature cluster within the features and feature set declarations used by the dictionary and grammar projects. A comparison of the existing entries for "believe" in the hand crafted lexicon (Figure 1) and the third word sense for "believe" extracted from LDOCE demonstrates that much of the information available from LDOCE is of direct utility — for example the SUBCAT values can be derived by an analysis of the **Takes** values and the **ORaising** logical type specification above. Indeed, we have demonstrated the feasibility (Alshawi et al., 1985) of driving a parsing system directly from the information available in LDOCE by constructing dictionary entries for the PATR-II system (Shieber, 1984).

It is also clear, however, that it is unrealistic to expect that on the basis of only the information available in the machine-readable source we will be able to derive a fully fleshed out lexical entry, capable of fulfilling all the run-time requirements of the analysis system that the lexicon under construction here is intended for.

3.3 Utility of LDOCE for automatic lexicon generation

Firstly, the information recoverable from LDOCE which is of direct utility is not totally reliable. Errors of omission and assignment occur in the dictionary — for example, the entry for "consider" (Figure 6) lacks a code allowing it to function in frames with sentential complement (eg. *I consider that it is a great honour to be here*). The entry for "expect", on the other hand, spuriously separates two very similar word senses (1 and 5), assigning them different grammar codes.

consider ... **2** [Wv5, X (*to be*) 1,7; V3] to regard as; think of in a stated way: *I consider you a fool* (= I regard you as a fool). | *I consider it a great honour to be here with you today.* | *He said he considered me (to be) too lazy to be a good worker.* | *The Shetland Islands are usually considered a part of Scotland* ....

expect ... **1** [T3,5a,b] to think (that something will happen): *I expect (that) he'll pass the examination.* | *He expects to fail the examination.* | *"Will she come soon?" "I expect so."* .... **5** [V3] to believe, hope and think (that someone will do something): *The officer expected his men to do their duty in the coming battle* ....

acknowledge ... **1** [T1,4,5 (*to*) to agree to the truth of; recognise the fact or existence (of): *I acknowledge the truth of your statement.* | *They acknowledged (to us) that they were defeated* | *They acknowledged having been defeated* **2** [T1 (*as*); X (*to be*) 1,7] to recognise, accept, or admit (as): *He was acknowledged to be the best player.* | *They acknowledged themselves (to be) defeated* ....

Figure 6: Errors of omission and assignment in LDOCE

Errors like these ultimately cause the transformation program to fail in the mapping of grammar codes to feature clusters. We have limited our use of LDOCE to verb entries because these appear to be coded most carefully. However, the techniques outlined here are equally applicable to other open class items.

Furthermore, since some of the information requred is only recoverable on the basis of a comparison of codes within a word sense specified in the source dictionary, additional errors can be introduced. For example, we assign **ORaising** to verbs which contain subcategorisation frames for sentential complement, a noun phrase object and an infinitive complement within the same sense. However, this rule breaks down in the case of an entry such as "acknowledge", where the two codes corresponding to different subcategorisation frames are split between two (spuriously separated) word senses (Figure 6), and consequently incorrectly assigns **OEqui** to this verb. The rule consequently breaks down and "consider" is incorrectly assigned the logical type of an **Equi** verb.

We have tested the classification of verbs into semantic types using a verb list of 139 pre-classified items available in various published sources (eg. Stockwell et al., 1973). The overall error rate in the process of grammar code analysis and transformation was 14%; however, the rules discussed above classify verbs into **SRaising, SEqui** and **OEqui** very successfully. The main source of error comes from the misclassification of **ORaising** into **OEqui** verbs. This was confirmed by another test, involving applying the rules for determining the semantic types of verbs over the 7,965 verb entries in LDOCE. The resulting lists, assigning the 719 verb senses which have the potential for predicate complementation into appropriate semantic classes, confirm that errors in our procedure are mostly localised to the (mis)application of the **ORaising** rule. Arguably, these errors also derive mostly from errors in the dictionary, rather than a defect of the rule; see Boguraev and Briscoe (1987) for further discussion.

Secondly, the analysis system requires information which is simply not encoded in the LDOCE entries; for example, the morphological features AT, LAT and BARE_ADJ are not there. This type of feature is critical to the analysis of derivational variants, and such information is necessary for the correct application of the word grammar. Otherwise many morphologically productive, but nonexistant, lexical forms will be defined and be potentially analysable by the lexicon system. Therefore, lexical templates are not converted directly to target lexical entries, but form the input to second phase in which errors and inadequacies in the source MRD are corrected.

4 A methodology and a system for lexicon development

In order to provide for fast, simple, but accurate development of a lexicon for the analysis system we have implemented a software environment which is integrated with the transformation program described above and which offers an integrated morphological generation package and editing facilities for the semi-automatic production of the target lexicon. The system is designed on the assumption that no machine-readable dictionary can provide a complete, consistent, and totally accurate source of lexical information. Therefore, rather than batch process the MRD source, the lexicon development software is based around the concept of semi-automatic and rapid construction of entries, involving the continuous intervention of the end user, typically a linguist / lexicographer.

In the course of an interactive cycle of development, a number of entries are hypothesised and automatically generated from a single base form. The family of related surface forms is output by the *morphological generator*, which employs the same word grammar used for inflectional and derivational morphology by the analysis system and creates new entries by adding affixes to the base form in legitimate ways. The generation and refinement of new entries is based on repeated application of the morphological generator to suitable base forms, followed by user intervention involving either rejecting, or minimally editing, the surface forms proposed by the system. Below we sketch a typical pattern of use.

If the user asks the system to create an entry for "believe", the transformation program described in section 3.2 (see Figure 4) will create an entry which contains all the syntactic information specified in Figure 1. In addition, many surface forms with associated grammatical definitions will be generated automatically:

cobelieve	overbelieve	subbelieve	believed
disbelieve	postbelieve	unbelieve	believee
interbelieve	prebelieve	underbelieve	believer
misbelieve	rebelieve	believable	believing
outbelieve	semibelieve	believal	believes

Figure 7: Derivational variants of "believe"

The system generates these forms from the base entry in batches and displays the results in syntactic frames associated with subcategorisation possibilities. These frames, which are used to tap the user's grammaticality judgements, are as semantically 'bleached' as possible, so that they will be as compatible as possible with the semantic restrictions that verbs place on their arguments. Each possible SUBCAT feature value in the grammar is associated with such frames, for example:

SFIN:	*They* ⊏ ... ⊐ *that someone is something*
OR:	*They* ⊏ ... ⊐ *someone to be something* *They* ⊏ ... ⊐ *there to be a problem*
OE:	*They* ⊏ ... ⊐ *that someone is something* * *They* ⊏ ... ⊐ *there to be a problem*

Figure 8: Syntactic subcategorisation frames

Internally, frames are more complex than illustrated above. Surface phrasal forms with marked slots in them are associated with more detailed feature specifications of lexical categories which are compatible with the fully instantiated lexical items allowed by the grammar to fill the slots. Such detailed frame specifications are automatically generated on the basis of syntactic analysis of sentences made up from the frame phrase skeleton with valid lexical items substituted for the blank slot filler. Figure 9 below shows a fragment of the system's inventory of frames.

```
They ⊏ ... ⊐ that someone is something.
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT SFIN]

They ⊏ ... ⊐ someone to be something.
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT OE]
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT OR]
 [N -, V ´+, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT SE2]

They ⊏ ... ⊐ there to be a problem.
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT SE2]
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT OR]

* They ⊏ ... ⊐ there to be a problem.
 [N -, V +, BAR 0, AGR [N +, V -, BAR 2, NFORM
    NORM, PER 3, PLU +, COUNT +, CASE NOM],
    SUBCAT OE]
```

Figure 9: Complete syntactic frames

The system ensures that slots in syntactic frames are filled by surface forms which have the syntactic features the sentence grammar requires. Displaying such instantiated frames provides a double check both on the outright correctness of the surface form and on the correctness of a surface form paired with a particular definition. For example, the user can reject *They overbelieve that someone is something* completely, but *They believes that someone is something* is indicative of an incorrect definition, rather than surface form. Syntactic frames encoding other 'transformational' possibilities are often associated with particular SUBCAT values since these provide the user with more helpful data to accept or reject a particular assignment. Thus for example selecting between **Raising** and **OEqui** verbs is made easier if the frames for [SUBCAT OR] are instantiated simultaneously:

They believe someone to be something /
They persuade someone to be something

They believe there to be a problem /
They persuade there to be a problem

Figure 10: SUBCAT value selection

The user has two broad options: to reject a set of frames and associated surface form outright or to edit either the surface form or definition associated with a set of frames. Exercising the first option causes all instances of the surface form and associated syntactic frames to be removed from the screen and from further consideration by the user. However, this action has no effect on the eventual output of the system, so these morphologically productive but non-existent forms and definitions will still be implicit in the lexicon and morphology component of the English analyser. It is assumed that this overgeneration is harmless though, because such forms will not occur in actual input.

Editing a surface form or associated definition results in a new (non-productive) entry which will form part of the system's output to be included as an independent irregular entry in the target lexicon. If the user edits a surface form, the edited version is substituted in all the relevant syntactic frames. Provided the user is satisfied with the modified frames, a new entry is created with the new surface form, treated as an indivisible morpheme, and paired with the existing definition. Similarly, if the user edits a definition associated with a set of syntactic frames, a new set of frames will be constructed and if he or she is happy with these, a new entry will be created with existing surface form and modified definition. (The English analyser can be run in a mode where non-productive separate entries are 'preferred' to productive ones.)

The user can modify both the surface form and the associated definition during one interaction with a particular potential entry; for example, the definition for "believal" contains both an incorrect surface form and definition for a nominal form of the base form "believe". After the associated syntactic frames are displayed to the user, instead of rejecting the entire entry at this point, he or she can modify the surface form to create a new entry for "belief" — a process which results in the revised syntactic frames:

The believal
The believal that someone is something
The believal someone to be something

The belief
The belief that someone is something
The belief someone to be something

Figure 11: Frame-based refinement of "belief"

The user now has three options; rejecting the third syntactic frame, or alternatively deleting the associated sub-entry with a [SUBCAT OR] feature definition, followed by confirmation will result in the construction of a new entry for the lexicon. The third option, should the user decide that nominal forms never take OR complements, is to edit the morphological rules themselves. This option is more radical and would presumably only be exercised when the user was certain about the linguistic data.

The system described so far allows the semi-automatic, computer-aided production of base entries and irregular, non-productive derived entries on the basis of selection and editing of candidate surface forms and definitions thrown up by the derivational generator. However, this approach is only as good as the initial base entry constructed from LDOCE. If the base entry is inadequate, the predictions produced by the generator are likely to be inadequate too. This will result in too much editing for the system to be much help in the rapid production of a sizeable lexicon. Fortunately, the system of syntactic frames and editing facilities outlined above can also be used to refine base entries and make up for inadequacies in the LDOCE grammar code system (from the perspective of the target grammar). For example, LDOCE encodes transitivity adequately but does not represent systematically whether a particular transitive has a passive form. In the target grammar, there are two SUBCAT values NP and NOPASS which distinguish these types of verb. Therefore, all verbs with a transitive LDOCE code are inserted into the two sets of syntactic frames shown below. When these frames are instantiated with particular verbs rejection of one or other is enough to refine the LDOCE code to the appropriate SUBCAT value. For example, the instantiated frames for "cost" are:

NP:	*They* ⊏ ... ⊐ *that* *Those are* ⊏ ... ⊐ *by them*
	They cost that *Those are cost by them*
NOPASS:	*They* ⊏ ... ⊐ *that* * *Those are* ⊏ ... ⊐ *by them*
	They cost that * *Those are cost by them*

Figure 12: The SUBCAT / NOPASS distinction

The fact that "cost" does not fit into the NP passive (second) frame, behaving in a way compatible with the NOPASS predictions, means it acquires a NOPASS SUBCAT value. Since these frames will be displayed first and the operation changes the base entry, subsequent forms and definitions generated by the system will be based on the new edited base entry.

This example, also highlights one of the inherent problems in our approach to lexicon development. Syntactic frames are used in preference to direct perusal of definitions in terms of feature lists to speed up lexicon development by tapping the user's grammaticality judgements directly and to reduce the amount of editing and keyboard input. They also provide the user with a degree of insulation from the technical details of the morphological and syntactic formalism. However, semantically 'bleached' frames can lead to confusion when they interact with word sense ambiguity. For example, "weigh" has two senses one of which allows passive and one of which does not (compare *The baby was weighed by the doctor* with * *Ten pounds was weighed by the baby*).

Unfortunately, the syntactic frames given for NP / NOPASS are not 'bleached' enough because they tend to select the sense of "weigh" which does allow passive. The example raises wider issues about the integration of some treatment of word meaning with the production of such a lexicon. These issues go beyond this paper, but the problem illustrated demonstrates that the type of techniques we have described are heuristic aids rather than failsafe procedures for the rapid construction of a sizeable and accurate lexicon from a machine-readable dictionary of variable accuracy and consistency.

5 Conclusion

Practical natural language applications require vocabularies substantially larger than those typically developed for theoretical or demonstration purposes and hand crafting these is often not feasible, and certainly never desirable. The evaluation of the LDOCE grammar coding system suggests that it is sufficiently detailed and accurate (for verbs) to make the on-line production of the syntactic component of lexical entries both viable and labour saving. However, the less than 100% accuracy of the code assignments in the source dictionary suggests that a system using the machine-readable version for lexicon development must embody a methodology allowing rapid, interactive and semi-automatic generation and testing of lexical entries on a large scale.

We have outlined a lexicon development environment, which embodies a practical approach to using an existing MRD for the construction of a substantial computerised lexicon. The system splits the derivation of target lexical entries into two phases; an automatic transformation of the source data into a formalised lexical template containing as much relevant information as can be derived (directly or indirectly), followed by semi-automatic correction and refinement of this template into a set of base and irregular target entries.

6 Acknowledgements

This work was supported by research grants (Numbers GR/D/4217.7 and GR/D/05554) from the UK Science and Engineering Research Council under the Alvey Programme. We are grateful to the Longman Group Limited for kindly allowing us access to the typesetting tape of the *Longman Dictionary of Contemporary English* for research purposes.

7 References

Alshawi, Hiyan; Boguraev, Bran and Briscoe, Ted (1985) 'Towards a dictionary support environment for a real-time parsing system', *Proceedings of the 2nd European Conference of the Associaition for Computational Linguistics*, Geneva, Switzerland, pp. 171–178

Boguraev, Bran; Carter, David and Briscoe, Ted (1987) *A multi-purpose interface to an on-line dictionary,* Third Conference of the European Chapter of the Association for Computational Linguistics, Copenhagen, Denmark

Boguraev, Bran and Briscoe, Ted (1987) Large lexicons for natural language processing — exploring the grammar coding system of LDOCE, *Computational Linguistics,* vol.13

Briscoe, Ted; Grover, Claire; Boguraev, Bran and Carroll, John (1987) *A formalism and environment for the development of a large grammar of English,* Tenth International Conference on Artificial Intelligence, Milan, Italy

Gazdar, Gerald; Klein, Ewan; Pullum, Geoffrey K.; and Sag, Ivan A. (1985) *Generalized phrase structure grammar,* Oxford: Blackwell and Cambridge: Harvard University Press

Grover, Claire; Briscoe, Ted; Carroll, John and Boguraev, Bran (1987, forthcoming) *The Alvey natural language tools project grammar — a large computational grammar of English,* Lancaster Papers in Linguistics, Department of Linguistics, University of Lancaster

Michiels, Archibal (1982) *Exploiting a large dictionary database,* Ph.D. Thesis, Université de Liège, Belgium

Procter, Paul (1978) *Longman dictionary of contemporary English,* Longman Group Limited, Harlow and London, England

Ritchie, Graeme; Pulman, Stephen; Black, Alan and Russell, Graham (1987) A computational framework for lexical description, *Computational Linguistics,* vol.13

Russell, Graham; Pulman, Steve; Ritchie, Graeme; and Black, Alan (1986) 'A dictionary and morphological analyser for english', *Proceedings of the 11th International Congress on Computational Linguistics,* Bonn, Germany, pp. 277–279

Shieber, Stuart (1984) 'The design of a computer language for linguistic information', *Proceedings of the 10th International Congress on Computational Linguistics,* Stanford, California, pp. 362–366

Stockwell, Robert; Schachter, Paul and Partee, Barbara (1973) *The major syntactic structures of English,* Holt, Rinehart and Winston, New York, NY