Sense extensions as Lexical Rules

Ted Briscoe and Ann Copestake

University of Cambridge Computer Laboratory
New Museums Site, Pembroke Street, Cambridge, CB2 3QG, UK

June 1991

(In Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language, Sydney, Australia)

ESPRIT BRA-3030 ACQUILEX Working Paper No. 022

Sense extensions as Lexical Rules

Ted Briscoe and Ann Copestake

Computer Laboratory, University of Cambridge, Pembroke Street, Cambridge, CB2 3QG, UK
ejb@cl.cam.ac.uk aac@cl.cam.ac.uk

Abstract

We define a declarative and computationally tractable concept of lexical rule as a component of a unification-based lexicon employing (default) inheritance and typed feature structures. We argue that this concept of a lexical rule is expressive enough to cover both derivational morphological processes and metonymic and metaphoric sense extensions, and that the application of such rules is productive within finely specified subsets of the lexicon identifiable via the type system. In addition, we argue that formulation of these processes in a uniform framework uncovers similarities between them; such as the blocking of metonymic sense extension processes in a fashion similar to the blocking of derivational morphological ones. We illustrate these points principally with reference to a family of increasingly specific rules of "grinding" and indicate how machine-readable versions of conventional dictionaries can be utilised as a source of information to guide identification and specification of such rules.

1 Introduction

In this paper, we take the position, persuasively argued in Pustejovsky (1989a,b), that the lexicon is not a passive list of unrelated entries, but rather a highly structured and generative device. The rule-governed processes which can be expressed as lexical rules extend far beyond those discussed under the rubric of lexical redundancy rule in classical generative grammar. We present a very general, though declarative and computationally tractable, definition of a lexical rule, formalised within the framework of unification-based approaches to lexical organisation employing typed feature structures and (default) inheritance. We are currently exploring the possibility that this account of lexical rules is sufficiently expressive to describe processes of derivational morphology, conversion or zero-derivation, and metonymic as well as metaphoric sense extension. Linguistic research has emphasised the phonological / graphological consequences of derivational processes to the extent that conversion or zero-derivation has often been treated in terms of "zero-morphemes" and "zero-affixation". In our work, we are concentrating on the syntactic and semantic consequences of such processes and treat all of them as mappings between lexical entries expressed as typed feature structures. In this framework, it is affixation, rather than conversion, which is the marked operation since it requires changes to the phonological or orthographic part of the lexical entry (see, e.g., Cahill (1990) for a theory of such operations couched in a unification-based account of the lexicon). We argue that there are sufficient similarities between these processes to treat them as lexical rules and that the distinction between metonymic processes of sense extension and more clearly derivational rules is not as clear-cut as it is sometimes assumed.

A standard example of metonymic sense extension is the use of a word denoting a place to refer to (some of) the people inhabiting that place (e.g. "village", "palace"). This process seems to involve the foregrounding of one component of the meaning of the place denoting word – we follow Pustejovsky (1989a,b,c) in assuming that the lexical semantic representation of nouns is richer than is standardly assumed in generative linguistic theory and that, in particular, the "qualia structure" for such nouns will contain (telic) information which allows direct access to the information that they are inhabited. A well-known example of metaphoric sense extension is that involving use of a word denoting an animal to refer to humans ("John is a pig", "John is a wombat" etc). Although the sense extension from animals into metaphorical senses denoting humans with some particular characteristic is apparently productive, the actual characteristics involved, and even whether the word can be applied to men or to women or both, cannot be predicted from knowledge of the animal sense. Thus, the properties ascribed to a person by "pig" are arguably no more than stereotypical associations with the animal, rather than central aspects of its meaning / qualia structure. In the case of "wombat" we would argue that the association of foolishness derives

from the phonological form of the word, rather than beliefs about the animal. Despite the more associative or analogical nature of metaphorical sense extension, we would argue that there is a core component to such processes which should be expressed in terms of a lexical rule, rather than in terms of general purpose reasoning. As with the metonymic cases (Pustejovsky, 1989c; Briscoe et al., 1990), we believe that the notion of coercion during syntactic and semantic interpretation provides an account of when a metaphorical interpretation will be adopted, and we would like to characterise the limits of coercion in terms of possible mappings defined by lexical rules.

An example of a derivational morphological process is the addition of the "er" suffix to verbs, typically creating a noun denoting the agent of the action denoted by the verb (e.g. "teach", "teacher"). There are several apparent differences between this type of process and the metonymic and metaphoric sense extensions considered above. The derivational rule involves a change of syntactic class, it affects the argument structure of the derived predicate, it involves affixation, and although there is a foregrounding of one aspect of the verb meaning, the result would not traditionally be described as a metonymic, or indeed metaphorical, usage. Nevertheless, there are clearly derivational processes which do not affect syntactic class (e.g. "re-program", "un-reprogrammable") and sense extensions which do; for example, countability of nouns changes depending on whether they are interpreted as types, substances or portions (e.g. "There was beer all over the table", "John drank a beer"). Not all derivational processes affect argument structure (e.g. "un-kind"), whilst metonymic sense extensions (e.g. "John enjoyed the film", John finished the beer") can; at least when given the analysis of Pustejovsky (1989c) and Briscoe et al. (1990). Finally, processes of conversion and derivation can be identical; for example, both "purchase" and "replace" have deverbal nominal forms "purchase" and "replacement", both nouns can denote the action involved and take appropriate complements ("Bill's purchase of his new car", "Bill's replacement of Sue with Mary"), and both can denote the result of the action ("Bill's purchases were many and varied" "Bill's replacement was Sue"). Traditionally, this latter resultative meaning would be described as metonymic and probably specialised and non-productive. We think that our definition of lexical rule will allow an account of both as productive syntactic and semantic operations mapping between lexical entries. The difference between metaphorical and metonymic operations is a matter of the degree to which the lexical rule determines or circumscribes the eventual interpretation.

There are other similarities between sense extension and derivational morphology; clearly, productivity is an issue in both, and in particular, sense extension processes may apparently be blocked (preempted by synonymy), in a way comparable to the situation in derivational morphology (see e.g. Bauer 1983:87f). For example, the regular form "stealer" does not generally occur, apparently because of the availability of "thief". Another productive metonymic sense extension is that of animal denoting (count) nouns to (mass) nouns denoting their meat (e.g. "lamb"), but this process too is blocked by the presence of a synonymous lexeme with different form ("pig", "pork"). By representing such processes in terms of lexical rules mapping between entries, we hope to account for blocking in terms of syntactic and semantic identity with an entry defined without recourse to the relevant lexical rule. In addition, we hope to express sense extension processes, and indeed derivational ones, as fully productive processes which apply to finely specified subsets of the lexicon, defined in terms of both syntactic and semantic properties expressed in the type system.

As part of the ACQUILEX project[1] we are investigating the occurrence of such extended and derived senses in machine readable dictionaries (MRDs). Although many regular relationships are recognised by lexicographers (Ostler and Atkins 1991), the representation of these extended and derived senses in existing MRDs is unsystematic. Thus our lexical knowledge base has to be able to deal with extensions which are never given a separate sense ("John is a wombat"), even if it is not possible to represent them completely. In this paper, we first examine in more detail some of the classes of sense extensions and their realisation in *The Longman Dictionary of Contemporary English* (LDOCE). We then describe the representation language of the lexical knowledge base (LKB) which is being used to represent information extracted from MRDs on the ACQUILEX project, and illustrate how it is possible to represent two of the examples mentioned.

[1] 'The Acquisition of lexical knowledge for Natural Language Processing systems' (Esprit BRA-3030)

The LKB is unification-based; it supports a restricted range of operations; (default) unification, (default) inheritance and lexical rule application and is intended as a lexical rather than a general purpose knowledge representation. However, we are using it to represent a relatively rich range of semantic information which allows us to indicate some of the shifts in meaning exemplified both in sense extension and in derivational morphology and which we hope will allow us to investigate the examples where productive processes are blocked.

2 The representation of sense extensions in dictionaries

We will consider two types of sense extension, both of which apply to words which denote animals. The first is the metaphorical use, mentioned earlier. The second involves the extension to a non-count sense with a meaning which can be paraphrased as "some substance derived from that animal", where this will normally be either fur/skin or meat/flesh. We in fact think of this as a special case of a more general sense extension rule which we will refer to as "grinding". It is well known that any count noun denoting a physical object can be used in a mass sense to denote a substance derived from that object, when it occurs in a sufficiently marked context. We refer to this as 'grinding' because the context normally suggested is the "Universal Grinder" (see Pelletier and Schubert 1986). So if "a table" is ground up the result can be referred to as "table" ("there was table all over the floor"). Several regular sense extensions can be regarded as special cases of 'grinding', where the extension may have become lexicalised. Thus besides the animal/meat examples, trees used for wood ("beech") have a sense denoting the wood, and so forth (see Copestake and Briscoe (1991) for further discussion).

We are investigating the representation of these sense extensions in LDOCE. Both extensions appear productive, at least in a sufficiently marked context; for example in the LOB corpus we find the use of "mole" as a mass term:

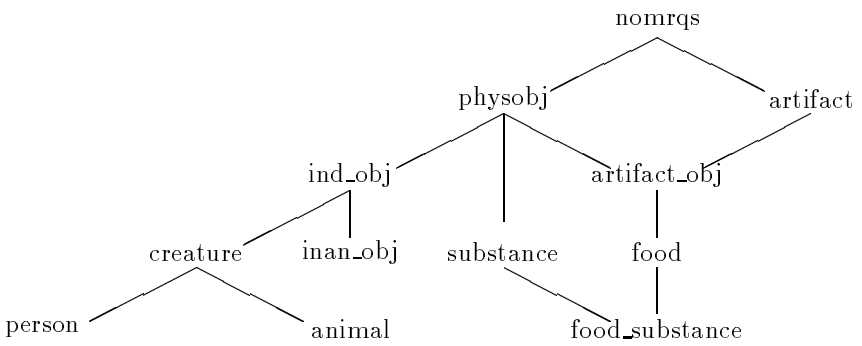
> Badger hams are a delicacy in China while mole is eaten in many parts of Africa.

but even when the use seems conventional it may or may not be represented as a separate sense in LDOCE ("lamb" is given a meat/flesh sense but "haddock" is not). Since space for definitions is very restricted in printed dictionaries, senses may be omitted even when the animal is conventionally used for food (the position is also confused by lexicographers' use of various techniques such as bracketing to "collapse" two senses into one). Furthermore, words which primarily denote the meat (eg "pork") normally block the sense extension process. Nevertheless this can often still apply in a more "marked" fashion. For example using "pig" instead of "pork" is possible but not lexicalised, and its use is marked, perhaps suggesting that the meat is of very inferior quality.

One technique which we are developing to investigate the extent to which regular sense extensions can account for polysemy in dictionary definitions, is to examine exhaustively all cases where a word has a primary sense in one semantic class (such as animal) and a secondary sense in another (such as person or substance). We can achieve this by deriving relatively complete sense-disambiguated taxonomies from the dictionary (using techniques described in Copestake (1990)), starting from the primary senses of words such as "person" and "animal" and intersecting the resulting taxonomies. Thus in the animal/person case there are over 100 words with senses in both taxonomies, the majority of which appear to be metaphorical uses. The other major class involves nominalisations such as "scavenger" being given separate senses (in this case one denoting a creature and the other a person). Furthermore the majority of the polysemy, at least for the class of words denoting animals, appears to be regular.

3 Representation in the LKB

The LKB is designed to represent any information which may be extracted from MRDs as part of the ACQUILEX project, including both syntactic and relatively complex semantic information. The LKB's representation language is based on the use of typed feature structures similar to those described in Carpenter (1990). Feature structures must be well-formed with respect to types



and particular features will only be appropriate to specified types and their subtypes. Types are hierarchically ordered; the association of *constraints* with types allows non-default inheritance. We augment this with a restricted concept of default inheritance (allowing only 'orthogonal' multiple inheritance (Touretzky 1986)); default inheritance is formalised in terms of default unification of feature structures ordered by an inheritance hierarchy. The type system constrains both default inheritance and lexical rule application. The LKB's representation language is described in detail in Copestake et al (1991); the following sections are an informal description illustrated with relevant examples.

3.1 The type system

The type hierarchy defines a partial ordering (notated $\sqsubseteq$) on the types and specifies which types are *consistent*. Only feature structures with mutually consistent types can be unified — two types which are unordered in the hierarchy are assumed to be inconsistent unless the user explicitly specifies a common subtype. Every *consistent* set of types $S \subseteq \mathsf{TYPE}$ has a unique greatest lower bound or meet (notation $\sqcap S$). This condition allows feature structures to be typed deterministically — if two feature structures of types **a** and **b** are unified the type of the result will be $\mathbf{a} \sqcap \mathbf{b}$, which must be unique if it exists. If $\mathbf{a} \sqcap \mathbf{b}$ does not exist unification fails. Thus in the fragment of a type hierarchy shown in Figure 1 **artifact** and **physobj** are consistent with $\mathbf{artifact} \sqcap \mathbf{physobj} = \mathbf{artifact_obj}$.

Our system differs somewhat from that described by Carpenter (1990) in that we adopt a different notion of well-formedness of typed feature structures. In our system every type must have exactly one associated feature structure which acts as a constraint on all feature structures of that type; it subsumes all well-formed feature structures of that type. The constraint also defines which features are *appropriate* for a particular type; a well formed feature structure may only contain appropriate features. Constraints are inherited by all subtypes of a type, but a subtype may introduce new features (which will be inherited as appropriate features by all its subtypes). A constraint on a type is a well-formed feature structure of that type; all constraints must therefore be mutually consistent. Constraints can be seen as extending the PATR-II notion of templates in that the inheritance of constraints allows concise definitions of all feature structures, not just lexical entries; however the type system also prevents inappropriate features occurring.

For example the constraints associated with the types **artifact** and **physobj** might be:

$$\left[\begin{array}{l}\textbf{artifact} \\ \textsc{purpose} = \textbf{formula}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{physobj} \\ \textsc{form} = \textbf{physform}\end{array}\right]$$

Bold case indicates types; thus, for instance the "value" **formula** can be any feature structure of

this type. The constraint on **artifact_obj** will contain information inherited from both parents, thus:

$$\begin{bmatrix} \textbf{artifact_obj} \\ \text{FORM} = \textbf{physform} \\ \text{PURPOSE} = \textbf{formula} \end{bmatrix}$$

Further examples of constraints and features which we will use in this paper are:

$$\begin{bmatrix} \textbf{ind_obj} \\ \text{FORM} = \begin{bmatrix} \textbf{physform} \\ \text{SHAPE} = \textbf{individuated} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{creature} \\ \text{AGE} = \textbf{scalar} \\ \text{SEX} = \textbf{gender} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{animal} \\ \text{EDIBLE} = \textbf{boolean} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{substance} \\ \text{FORM} = \begin{bmatrix} \textbf{physform} \\ \text{SHAPE} = \textbf{unindividuated} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textbf{food} \\ \text{PURPOSE} = \begin{bmatrix} \textbf{formula} \\ \text{PRED} = \textbf{eat} \end{bmatrix} \end{bmatrix}$$

All these types are intended to represent that part of the lexical entry which contains relatively complex semantic information, the "relativised qualia structure" (Calzolari 1991).

We shall also make use of the following types to define syntactic properties etc:

$\textbf{lex-sign} \sqsubseteq \textbf{top}$ $\quad \begin{bmatrix} \textbf{lex-sign} \\ \text{ORTH} = \textbf{string} \end{bmatrix}$

$\textbf{noun} \sqsubseteq \textbf{lex-sign}$ $\quad \begin{bmatrix} \textbf{noun} \\ \text{SYNTAX} = \begin{bmatrix} \text{COUNT} = \textbf{boolean} \end{bmatrix} \\ \text{RQS} = \textbf{nomrqs} \end{bmatrix}$

$\textbf{count-noun} \sqsubseteq \textbf{lex-sign}$ $\quad \begin{bmatrix} \textbf{noun} \\ \text{SYNTAX} = \begin{bmatrix} \text{COUNT} = + \end{bmatrix} \end{bmatrix}$

$\textbf{mass-noun} \sqsubseteq \textbf{lex-sign}$ $\quad \begin{bmatrix} \textbf{noun} \\ \text{SYNTAX} = \begin{bmatrix} \text{COUNT} = - \end{bmatrix} \end{bmatrix}$

The feature structure below is well-formed since it contains all the appropriate features and no inappropriate ones, it is subsumed by the constraints on its type and all its substructures are well-formed.

$$\begin{bmatrix} \textbf{count-noun} \\ \text{ORTH} = \textbf{"haddock"} \\ \text{SYNTAX} = \begin{bmatrix} \text{COUNT} = + \end{bmatrix} \\ \text{RQS} = \begin{bmatrix} \textbf{animal} \\ \text{SEX} = \textbf{gender} \\ \text{AGE} = \textbf{scalar} \\ \text{EDIBLE} = \textbf{boolean} \\ \text{FORM} = \begin{bmatrix} \textbf{physform} \\ \text{SHAPE} = \textbf{individuated} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Given the type system introduced above, a lexical entry in the LKB:

```
haddock 1 count-noun
       <rqs> = animal.
```

would be expanded out into such a feature structure[2] .

3.2 Lexical rules

A lexical rule in the LKB is a feature structure of type **lexical-rule**. The expanded constraint for the type is:

$$\begin{bmatrix} \textbf{lexical_rule} \\ 0 = \textbf{lex_sign} \\ 1 = \textbf{lex_sign} \end{bmatrix}$$

thus all lexical rules have to have the features 0 and 1 which must both have values which are of type **lex_sign**.

New lexical signs may be generated by unifying a copy of the lexical entry with the feature structure at the end of the path `<1>` in a copy of the lexical rule — the feature structure at the end of the path `<0>` is then the new lexical sign. Lexical rules are indexed by the type of their "input" and "output" feature structures, so they will only be applied to entries of the appropriate type.

A general type for grinding lexical rules could be specified in the LKB as follows:

$$\textbf{grinding} \sqsubseteq \textbf{lexical_rule} \qquad \begin{bmatrix} \textbf{grinding} \\ 1 = \begin{bmatrix} \textbf{count-noun} \\ \text{ORTH} = \boxed{1} \\ \text{RQS} = \textbf{ind_obj} \end{bmatrix} \\ 0 = \begin{bmatrix} \textbf{mass-noun} \\ \text{ORTH} = \boxed{1} \\ \text{RQS} = \textbf{substance} \end{bmatrix} \end{bmatrix}$$

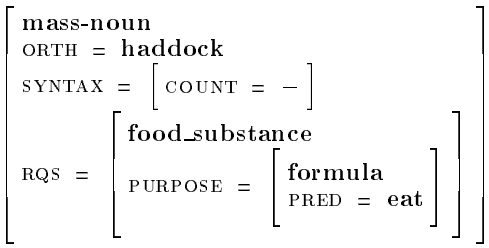
The effect of the lexical rule is to transform a count noun with the "relativised qualia structure" (RQS) properties appropriate to an individuated physical object **ind_obj** into a mass noun with properties appropriate for a substance **substance**.

We specialise the grinding rule to allow for cases such as the animal/meat extension explicitly. The typed framework provides us with a natural method of characterising the subparts of the lexicon to which such rules should apply. The lexical rules can, in effect, be parameterised by inheritance in the type system. For example given the type hierarchy shown in Figure 1 we can then give rules which inherit information from **grinding** such as animal_grinding:

$$\text{animal_grinding} \qquad \begin{bmatrix} \textbf{grinding} \\ 1 = \begin{bmatrix} \text{RQS} = \begin{bmatrix} \textbf{animal} \\ \text{EDIBLE} = + \end{bmatrix} \end{bmatrix} \\ 0 = \begin{bmatrix} \text{RQS} = \textbf{food_substance} \end{bmatrix} \end{bmatrix}$$

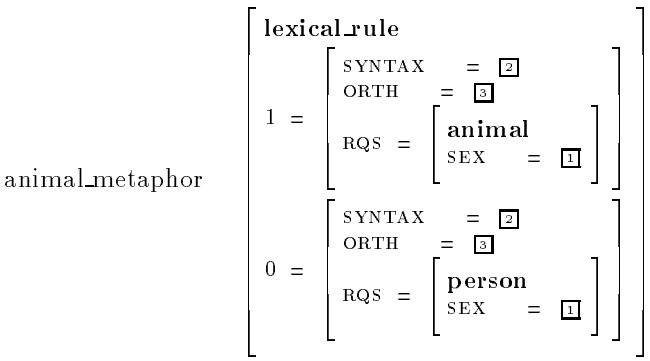
Thus given the lexical entry for "haddock" shown above we can apply the lexical rule to generate a sense meaning "haddock-flesh" (partially represented as):

[2] The actual type system being employed is considerably more complex, since only the relevant features are being shown in these examples.



(where the specification of the value **eat** for the purpose role arises from the constraint on the type **food_substance**, inherited from **food**, and the type mass-noun arises from the grinding type.)

The lexical rule for the metaphorical use of animal words might be something like:



In principle, multiple lexical rules may be applied in sequence. For example another lexical rule which we might specify is "portioning" which would convert food (or drink) denoting mass nouns into count nouns denoting a portion of that substance (eg "three beers", and even "three haddocks" in the context of a restaurant). All outputs of lexical rules must be potentially valid lexical entries. In the case of zero-derivational processes, like those we have been defining, we wish to restrict the set of lexical rules so that application may not be circular — that is if there is a lexical rule which could generate the set of feature structures F2 from the set F1, no other lexical rule or sequence of lexical rules may be specified which could generate any member of F1, or a feature structure subsuming any member of F1, starting from any member of the set F2, since lexical rule application would not then terminate. One straightforward way of ensuring this condition is met is to constrain the set of lexical rules so that if a rule which transforms feature structures of type **t_1** to **t_2** exists no sequence of rules exists which transforms **t_2** to **t_1** or any supertype of **t_1**. However, this condition is overrestrictive because some types of derivational rule can apply to their own output iteratively ("meta-meta-theory", "anti-anti-missile", "great-great-grandmother", "re-re-program"). Such differing properties of lexical rules can be formalised as they emerge by creating distinct types of lexical rule and placing well-formedness constraints on each type. Expressing constraints of this type will require a more expressive language than we currently deploy in the LKB, in the spirit of that described in Carpenter et al. (1991). The richer these constraints the more a substantive theory of lexical processes will emerge within the formal computational framework outlined above.

3.3 Default inheritance

To allow default inheritance we introduce the concept of *psort*; a feature structure from which another feature structure inherits information, by default. The hierarchical ordering on psorts (which must be consistent with the type hierarchy) provides an order on defaults. Default inheritance is implemented by a version of default unification. Only orthogonal multiple inheritance (Touretzky 1986) is allowed; information inherited from multiple parents must not be contradictory. A default inheritance hierarchy which connects semantic parts of lexical entries can be derived semi-automatically from taxonomies extracted from MRDs (Copestake 1990). Defaults may also be useful in the representation of syntactic information in the lexicon (e.g. Flickinger, 1987)

The use of default inheritance in which we are primarily interested here is to allow more specific information about an extended word sense extracted from an MRD to augment and possibly override information inherited from the result of application of a lexical rule. In cases where the lexical rule predicts the extended sense correctly, the specific information will duplicate information already present. If the rule is correct, but incomplete, the specific information will augment the inherited information. If it is partially incorrect, the more specific information will override that inherited from the result of lexical rule application.

For example, if the result of applying the lexical rule to a sense is notated as sense+rule-name (eg lamb_1+animal_grinding) then the representation of the LDOCE sense lamb (2) (the meat) in the LKB might be:

```
lamb 2 < lamb_1+animal_grinding.
```

where < is used to indicate default inheritance from a psort. In this case no extra information need be added. In contrast the entry for lamb (3) ("a young gentle person") might augment the information inherited from the lexical rule:

```
lamb 3 < lamb_1+animal_metaphor
< rqs : age > = low.
```

In the case of haddock, where no LDOCE entry is found, the structure derived from the lexical rule alone would be used.

Both the types of sense extension that we have been considering seem to be productive. Examples such as:

> Don't be such a wombat!

(said to someone behaving stupidly, in the speaker's opinion) seem to be perfectly acceptable, but such a use of "wombat" is probably not found in any dictionary. If we are to attempt to process such utterances, the structure generated by the lexical rule itself has to be sufficient. This can give the information that "wombat" may denote a person, but not what the characteristics imputed are, since these are not predictable. However, most such uses occur in marked contexts where it is relatively obvious (to a human reader) what characteristics are meant (and in any case information from the lexical rule alone would be sufficient for many NLP applications). Such lexical rules should only be invoked in a suitably marked context and we have argued elsewhere, on the basis of corpus data, that coercion is triggered by predicational environment and subsequent overriding of the default metonymic interpretation (or perhaps elaboration of the metaphorical one) is triggered by marked, informationally-rich contexts (see Briscoe et al., 1990; Copestake and Briscoe, 1991). Below we give an example of metaphorical use of "shark" taken from the SEC corpus which illustrates these (and some other) processes. (Note that the sense of "shark" involved here is not related to the arguably distinct homograph relating to (financial) malpractice or criminality.)

> Nowhere more so than on the world's golf courses, where a man they call "the great white shark" was on the prowl... That one little tap in at Turnberry put paid... to the cruel jibe that Greg Norman was just a great white fishfinger. His transformation to a genuine killer shark began on that windswept course...

4 Conclusion

We have illustrated how some cases of metonymic and metaphorical use of words might be at least partially represented in a lexical knowledge base system, with much the same apparatus as is needed for the representation of derivational morphology. Our work on using MRDs to investigate this behaviour is at an early stage; we have illustrated how relatively crude techniques like taxonomy intersection can be used to get at some of the classes of words which undergo this behaviour, but in order to do this more completely we need to use more sophisticated extraction techniques. The

(partial) representation of information derived from MRDs in the LKB should give us a way to investigate such sense extensions as well as being practically useful in NLP systems.

It is at least arguable that much of the information that we are representing here is real world knowledge rather than lexical information, and it is clear that our representation is only partial; both because we are simplifying information from the definitions and forcing it into limited templates provided by the type system and because the LKB's limited operations fall far short of an inference system that is capable of reasoning about real world knowledge. We have argued elsewhere (Briscoe et al (1990), following Pustejovsky (1989)) that lexical processes do involve far more semantic information than is generally represented in the lexicons of NLP systems, but a further practical argument for attempting this sort of representation is that it is needed to detect when lexical rule application may be "blocked" by the existence of other word forms.

As mentioned above the phenomenon of blocking appears to occur with some cases of regular sense extension in a way that seems similar to derivational morphology. In the cases where a word sense exists such as "pork" we need to recognise its equivalence with the sense obtained from "pig" using the lexical rule. To do this we need a rich representation which indicates information such as "origin" which has not been mentioned above. Because words like "pig" can be used in the extended sense, in marked contexts, the correct way to represent blocking in the LKB is not to prevent application of the lexical rule but the generated occurrence should be noted as being in some way peculiar. Sense extensions may also be blocked if a phonologically similar word exists which could be confused with the extended sense; for example it appears unusual for "deer" to be used in a metaphorical sense presumably because it could be confused with "dear"; again this phenomenon apparently also exists in derivational morphology (Bauer 1983:97). However these effects are not well-understood — for example blocking does not seem to apply to the metaphorical sense extensions in the same way (for example **colt (2)**, **fledgling (2)** and **greenhorn (1)** are given very similar definitions in LDOCE). Rather than attempting to decide on a representation at this point we intend to investigate the effect on some comprehensive set of examples, using the MRDs, to try and establish a suitable treatment.

References

Bauer L(1983) *English Word-formation,* CUP

Briscoe E J, Copestake A A and Boguraev B K(1990) 'Enjoy the paper: Lexical semantics via lexicology', *Proceedings of the 13th Coling,* Helsinki, pp.42–47

Cahill L(1990) 'Syllable based morphology', *Proceedings of the 13th Coling,* Helsinki, pp.48–54

Calzolari N(1991) 'Acquiring and representing semantic information in a lexical knowledge base', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp.188–197

Carpenter R(1990) 'Typed feature structures: Inheritance, (In)equality and Extensionality', *Proceedings of the Workshop on Inheritance in Natural Language Processing,* Tilburg, pp.9–18

Carpenter R, Pollard C and Franz A(1991) 'The specification and implementation of constraint-based unification grammars', *Proceedings of the Second International Workshop on Parsing Technologies,* Cancun, Mexico, pp.143–153

Copestake A A(1990) 'An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary', *Proceedings of the Workshop on Inheritance in Natural Language Processing,* Tilburg, pp.19–29

Copestake A A, de Paiva V C V, Sanfilippo A and Briscoe E J(1991) *Functionality of the LKB,* Ms. Computer Laboratory, University of Cambridge

Copestake A A and Briscoe E J(1991) 'Lexical Operations in a Unification Based Framework', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp.88-101

Flickinger D(1987) *Lexical rules in the hierarchical lexicon,* PhD dissertation, Stanford University

Ostler N and Atkins B T S(1991) 'Predictable Meaning Shift: Some Linguistic Properties of Lexical Implication Rules', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp.76–87

Pelletier F J, and Schubert L K(1986, forthcoming) 'Mass Expressions.' in Gabbay and Guenthner (eds.), *Handbook of Philosophical Logic, Vol 4,* Reidel, Dordrecht

Pustejovsky J(1989a) *The Generative Lexicon,* Ms. Brandeis University

Pustejovsky J(1989b) 'Current issues in computational lexical semantics', *Proceedings of the 4th European ACL,* Manchester, pp.xvii–xxv

Pustejovsky J(1989c) 'Type coercion and selection', *Proceedings of the West Coast Conference on Formal Linguistics,* Vancouver

Touretzky D F(1986) *The mathematics of inheritance systems,* Morgan Kaufmann, Los Altos