Title

Predicating Nominal Compounds

Permalink

https://escholarship.org/uc/item/8wj1r8md

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 17(0)

Author

Jones, Barnard

Publication Date

1995

Peer reviewed

Predicating Nominal Compounds

Bernard Jones
Centre for Cognitive Science
University of Edinburgh
2 Buccleuch Place
Edinburgh, EH8 9LW
United Kingdom
bernie@cogsci.ed.ac.uk

Abstract

It is generally accepted that in the semantic interpretation of compound nominals there is a set of possible relationships that could apply between the nominal constituents. This, however, has not been reflected adequately in the literature, which favours very deterministic processing or analyses performed on a pragmatic level. This study extends the existing set of relationships described by Levi (1978), postulating a set of rules to predict a subset of these relationships for a particular compound using a unification-based formalism with typed feature-structures. The system shows that by operating on a purely semantic level a small set of valid predicates for the meaning of the whole compound can be obtained.

Introduction

Linguistic approaches to the problem of nominal compounding have tried to classify the compound on purely grammatical criteria, but fail to provide constraints that can consistently explain the semantic interpretation of nominal compound. More recently, however, there have been attempts to view nominal compound interpretation as governed by tight semantic constraints. Levi (1978) developed a theory of nominal compounding whereby either the head noun is a nominalisation and its modifier is interpreted as an argument of the related verb (a *ship builder* BUILDS ships), or the two elements of the compound are related by a possible nine specific deletable predicates (e.g. FROM relates *olive oil*, HAVE relates *government land*, and MAKE relates *honey bee*) The nine predicates are described in more detail later. Despite the fact that this approach was criticised for being too pragmatically dependent (Downing, 1977), many computational approaches to nominal compounds are based around Levi's work.

Finin (1980) was one of the first to use parts of Levi's theory to attempt interpretation of compounds. His system, working on a large but restricted set of compounds, generates a single 'strongest likelihood' semantic interpretation using a very specific set of productive and structural rules. The system has problems generalising, however, since it would require a virtually unbounded set of rules.

Isabelle (1984) addresses some of the shortcomings of Finin's work in a reasonably flexible nominal compound resolution system. He addresses compounding in two ways — either the head noun is treated as predicative (so the head subcategorises for the other compound element) or as a nominalisation. Isabelle differs from Levi by using six types of nominalisation where the nominalising verb does not necessarily have to be related to the root of the nominal. A drawback of this 'non-relation' is that the system has to be very rigidly specified. The system is also rather inflexible generally since each head noun only has one possible predicating verb associated with it.

The conclusions of linguistic research, that there is a constrained set of possible relations for any given nominal compound, seem to have been ignored in the main. One exception is the approach of Hobbs and Martin (1987), which postulates an unspecified predicate that acts between the compound elements, and tries to prove the identity of this predicate using a pragmatic knowledge base. This results in all applicable semantic representations that the knowledge base contains being assigned to the compound nominal, but will also mean that all semantic disambiguation is pragmatic, which is not really computationally or linguistically tractable with real-world data.

A totally different approach to the problem is taken by Bouillon et al. (1992), where a large but well defined and closed set of compounds are lexicalised to include a representation of their meaning. In any realistic non-closed system, however, a lexicalisation approach is unlikely to be suitable since a huge lexicon would be necessary.

Therefore some mechanism is needed that yields the possible interpretations of a compound without resorting to totally lexical or pragmatic methods. A small amount of pragmatic post-processing (which would always be necessary to handle the exocentric and metaphorical compounds anyway) could then produce the most appropriate interpretation. Of course, this does not preclude the possibility of lexicalising frequent or difficult compounds (*panty-hose, hatchback*).

This paper represents a novel attempt to use these linguistic insights in the development of a wide-coverage nominal compound interpreter that will yield a small set of possible relations for every compound processed, dependent on the semantic type of the compound elements.

Approach

The linguistic hypothesis, that for a particular compound there are only a restricted set of possible applicable relationships, can also be stated in a different way if we assume that the assignment of these relationships is non-arbitrary. Namely: for each semantic head of a compound there is a restricted set of relationships that can apply between the head and the other element of the compound (the modifier). Similarly, for each modifier there is also a set of possible relationships that can apply between it and the head. This simplifies matters considerably since rather than trying to recover the set of relationships given a novel compound, it should be possible to infer the relevant relationships from the components of that compound. In addition, given that multiple predicates will be associated with each nominal, it is easy to produce a set of predicates as the output of the analysis stage. The assignment of predicates will be based on the application of world-knowledge to the compound elements, and there are essentially two locations where this knowledge can be encoded.

If the information is lexicalised then the entry for each possible compound head would contain a *characteristic set*, $\mathcal{S}_{head}$, containing all the predicates that could possibly be applied to a compound with that head (1). Such a system could easily result in over-generation of interpretations, unless the choice of predicates is made with respect to the modifier, in which case some of the world information needs to be stored in the grammar anyway.

(1) $\mathcal{S}_{book}$ = {physical-composition, subject, use...}
paperback book, physics book, spelling book...

The alternative to lexicalisation is to place the information in the grammar and make use of the semantic features of the compound elements to decide which rules to apply. This solution is necessarily less specific than the previous one, since the identities of head and modifying elements must be generalised by some form of type system, but it is far more efficient to write a rule, $\mathcal{R}_{relation}$, or set of rules for every possible relation that could hold between elements than including a set of such relations in every single lexical entry. Thus basic grammatical rules are necessary, each encoding one predicate, that require a head and/or modifier of a particular semantic type in order to be applicable to the compound in question. (2) illustrates the putative rule for relation PHYSICAL-COMPOSITION.

(2) $\mathcal{R}_{phys-comp}$ *iff* [head=concrete & modifier=physical]
concrete dog, *but not* concrete idea *or* sun dog

The problems and relative merits of these approaches lie in the tradeoff between lexicon size (and plausibility) and restricting the possible relations. Both approaches have advantages in particular situations, but the problems with the lexical approach are of a more serious and fundamental nature if we are to aim for a general coverage of compounds. Therefore a rule-based approach, with its associated possibility of slight overgeneration, has been chosen here.

Predicate Hierarchy

The predicates necessary to semantically interpret nominal compounds will range from the most specific, designed to capture a precise relationship between specific nominal elements, to the most general, designed to act as a catch-all, encoding a very generalised meaning for nominal compounds which none of the more specific predicates have captured. The most appropriate format for such a set of predicates therefore is a hierarchy.

To act as the root node of the hierarchy a *general compound* predicate is necessary that has minimally specified arguments, but yields a standard interpretation for a nominal compound. A rule encoding such a predicate will have to generate a grammatical entity that incorporates the syntactic features of the head element of the compound and that semantically links the head to the modifier through a non-specific (or null) predicate.

At deeper levels of the hierarchy, there are many possible predicates that represent the set of possible relationships between two nominal elements. A compromise must be found between over-specificity and over-generality. The former will lead either to impossible degrees of over-generation in the result and over-specification of the predicate arguments, while the latter will lead to a set of results that are too general for productive semantic processing.

A suitable starting point for a sensible set of predicates is the set of nine that Levi argues are recoverably deletable in the process of complex nominal formation (Levi, 1978, p76). These are loosely defined as:

CAUSE	HAVE	MAKE
USE	BE	IN
FOR	FROM	ABOUT

In addition Levi's nominalisations (ibid, p168) introduce a potentially infinite set of predicates corresponding to verbs that have been nominalised through derivational morphology e.g. *builder* from *build*, *invention* from *invent* and *error* from *err*. There are four types of nominalisation under Levi's system; act, product, agent and patient. The further nominalisation types proposed by Isabelle (1984) are not strictly necessary since they are subsumed by other predicates.

Levi's predicates are rather general, but are suitable for the second level of the predicate hierarchy to act as 'attachment points' for more specific predicates. Whilst some researchers, notably Finin (1980), have come up with remarkably specific predicate rules, such as DISSOLVED-IN, this represents an intractable degree of specificity; therefore the deeper levels of the hierarchy should only contain a small number of predicates.

The CAUSATIVE predicate, for example can be subdivided into cases where the head causes the modifier (*disease cell*) or is caused by the modifier (*nicotine fit*). The hierarchy can similarly be extended under the nominalisation rules. Wheras agent and patient nominalisations (*meat cleaver*, *student invention*) require predicates of arity two (CLEAVE, INVENT), the act and product nominalisations (*birth control*, *ocean study*) require predicates of arity one (CONTROL, STUDY). Therefore both these nominalisations can be subdivided into cases where the modifier acts as subject or object of the predicating verb. The full predicate-hierarchy generated by extension of Levi's predicates is seen in Figure 1.

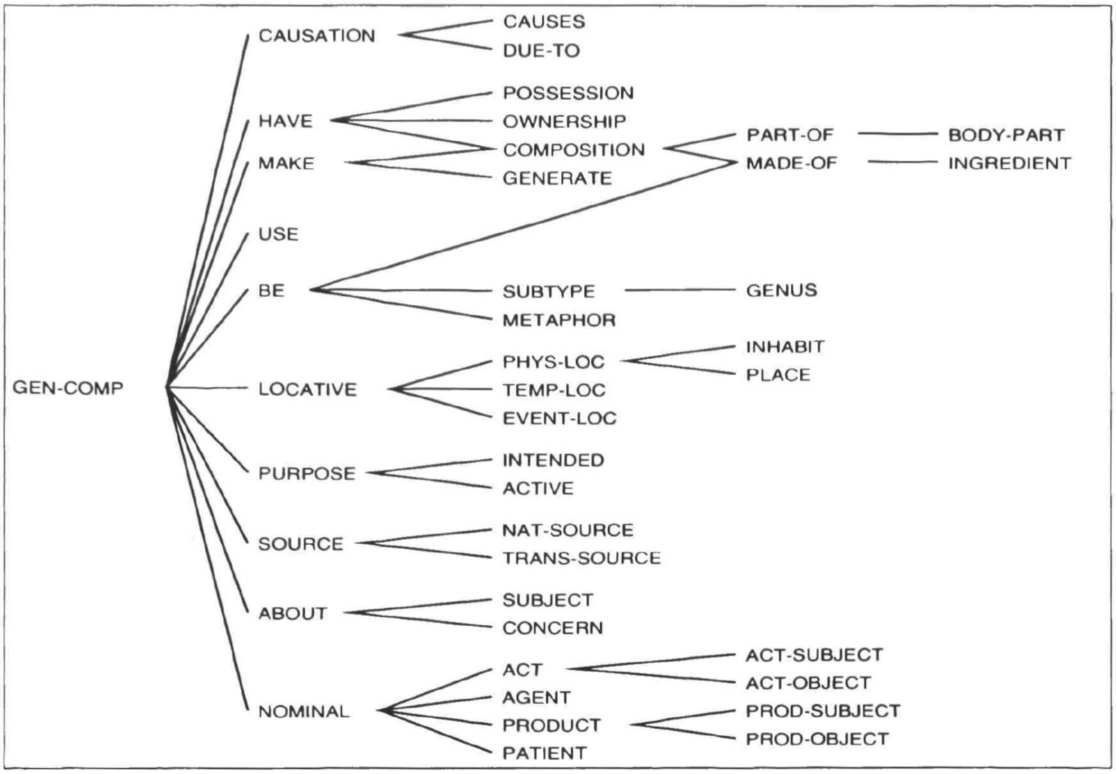

Figure 1: Predicate Hierarchy for Nominal Compounds

Implementation

The nominal compound rule system was implemented in the Acquilex LKB system (Copestake, 1992 & 1993), a typed graph-based unification formalism allowing default inheritance. The default inheritance machanism is a very useful way to implement the hierarchical rules — the more specific ones inheriting by default from their less specific mothers. The grammatical rules are represented as 3-element feature-structures (Figure 2) — the elements corresponding to the result of the rule application and the two arguments of the rule. These rule-elements are labelled 0, 1 (modifier) and 2 (head) respectively

The specification of the modifier and head noun is necessary to ensure the correct rules are being applied. This is straightforward using the type system in the LKB, in particular the class of nominal qualia types, and associated features such as physical form and properties. The LKB qualia type hierarchy is shown in Figure 3.

The most complex rule to define will be the base-level, general compounding rule (Figure 2), since this will incorporate all the mechanisms for the syntactic and normal semantic specification of arguments and resultant structure. Further rules however, for deeper predicates in the hierarchy, will simply inherit this information through the LKB's default inheritance mechanism, and further specify the qualia attributes of the arguments (and the identity of the predicate linking the compound elements). Examples (3) and (4) show how the HAVE rule inherits from GEN-COMP, and how the COMPOSITION

```
grammar-rule
0 = [ sign
      orth = [ complex-orth
               orth1 = [1]
               orth2 = [2] ]
      cat = [3]
      sem = [ binary-formula
              ind = [4]
              pred = and
              arg1 = [5]
              arg2 = [ binary-formula
                       ind = [4]
                       pred = logical-pred
                       arg1 = [4]
                       arg2 = [6] ] ]
      qualia = [7] ]
1 = [ sign
      orth = [1]
      sem = [6] ]
2 = [ sign
      orth = [2]
      cat = [3]
      sem = [5] [ unary-formula
                  ind = [4] ]
      qualia = [7] ]
```

Figure 2: Feature structure for the GENERAL-COMPOUND rule

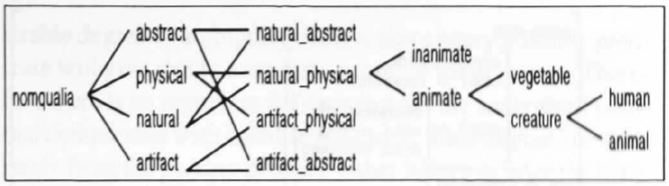

Figure 3: The LKB nominal qualia type system

rule inherits from HAVE, replacing the predicate and restricting the qualia-types of its arguments.

```
(3) have
    grammar-rule
    <> < general-compound<>[a]
    < 0:sem:arg2:pred > = "has"
```

[a] Default inheritance.

```
(4) composition
    grammar-rule
     <> < have<>
     < 0:sem:arg2:pred > = "composed-of"
     < 1:qualia > = physical
     < 2:qualia > = physical.
```

If a particular predicate can be applied in several qualia-circumstances, then several similar number rules are produced for each circumstance e.g. TRANS-SOURCE1 if the modifier is inanimate, TRANS-SOURCE2 if it is a physical artifact.

The nominalisation rules should depend crucially on derivational morphological processes to function properly. Since the LKB version used for this implementation did not support morphology[1], lexical entries are given for nominalisations, as though they had been through the derivational process. A system that supported derivational morphology, however, would not need the nominalisations lexicalised — the requisite predicating verb could be derived in processing.

Since the system will apply all the compounding rules possible, overgeneration results where specific rules from deep levels of the hierarchy apply, since all the rules higher up the hierarchical tree are also be applied. Therefore the implemented rules are also hierarchically organised within the grammar, via the default-inheritance principle. In the rule-application stage the less-specific rules are blocked if one of their child-rules has been applied. The final grammar contains 44 rules to implement the predicate hierarchy for nominal compounding.

Results

Since the rules for compounding take the form of grammar rules, nominal compounds must be fed into the parser to receive interpretations, producing a different parse tree for every predicate that can be applied. The set of examples in Figure 4 shows the rules that have been applied in the parsing

birth pain	(active3 DUE-TO causes)
disease cell	(active3 make CAUSES)
student power	(intended POSSESSION)
government land	(intended place2 place1 OWNERSHIP1 nat-source2)
dog leg	(intended place1 BODY-PART)
honey bee	(inhabit made-of1 GENERATE)
sweat gland	(phys-loc made-of1 GENERATE)
music box	(active3 GENERATE)
pine tree	(intended inhabit have SUBTYPE2)
mountain stream	(active1 PLACE1 made-of1 part-of1 trans-source1)
human vertebrate	(intended inhabit ownership1 SUBTYPE3)
field mouse	(active1 INHABIT made-of1 trans-source1)
spring shower	(TEMP-LOC generate)
morning prayer	(about TEMP-LOC due-to)
marital sex	(TEMP-LOC active3 due-to)
olive oil	(intended place1 have NAT-SOURCE2)
sea breeze	(place1 made-of1 active1 part-of1 TRANS-SOURCE1)
physics book	(ABOUT active3)
room temperature	(POSSESSION active2 trans-source2 due-to)
abortion vote	(ABOUT active3 due-to causes)
cow phone	(intended PLACE2 have)
cardiff woman	(INHABIT made-of2 active2 trans-source2)

Figure 4: Sample results (normal predicates)

city employee	(PATIENT inhabit made-of2 trans-source2)
ocean study	(PRODUCT about possession trans-source1 due-to)
birth control	(ACT causes)
music critic	(AGENT generate)

Figure 5: Sample results (nominalised predicates)

of several different compounds (the correct/most suitable rule has been (manually) highlighted in each case).

The sample compound 'cow phone' was motivated by the Far Side cartoon depicting a farmer talking into a telephone mounted behind a flap in the side of a cow, entitled "The rural professional and his cowphone" (Larson, 1989). This is obviously a pun on carphone, but the (correct) interpretation assigned to what is essentially a nonsensical compound illustrates that the system can generate correct predicates corresponding to pragmatic sense-extension in some circumstances.

Overgeneration in the results generally is purely due to underspecificity in the lexicon and the qualia type system (and hence in the definition of the compounding rules). The qualia type system used for this study was an experimental one, and is terribly small and underspecified. There are only fifteen nodes in the whole system, as shown in Figure 3, specifying little other than animacy, physical composition and the natural/artifact distinction. With a properly specified system, the sets of results would be much smaller, possibly even atomic, which is what is ultimately desired.

Testing the nominalisation rules (examples in Figure 5), it

[1] This is due to the difficulties of implementing any morphological component in an inheritance-based system. More recent versions of the LKB, however, do have a morphological component.

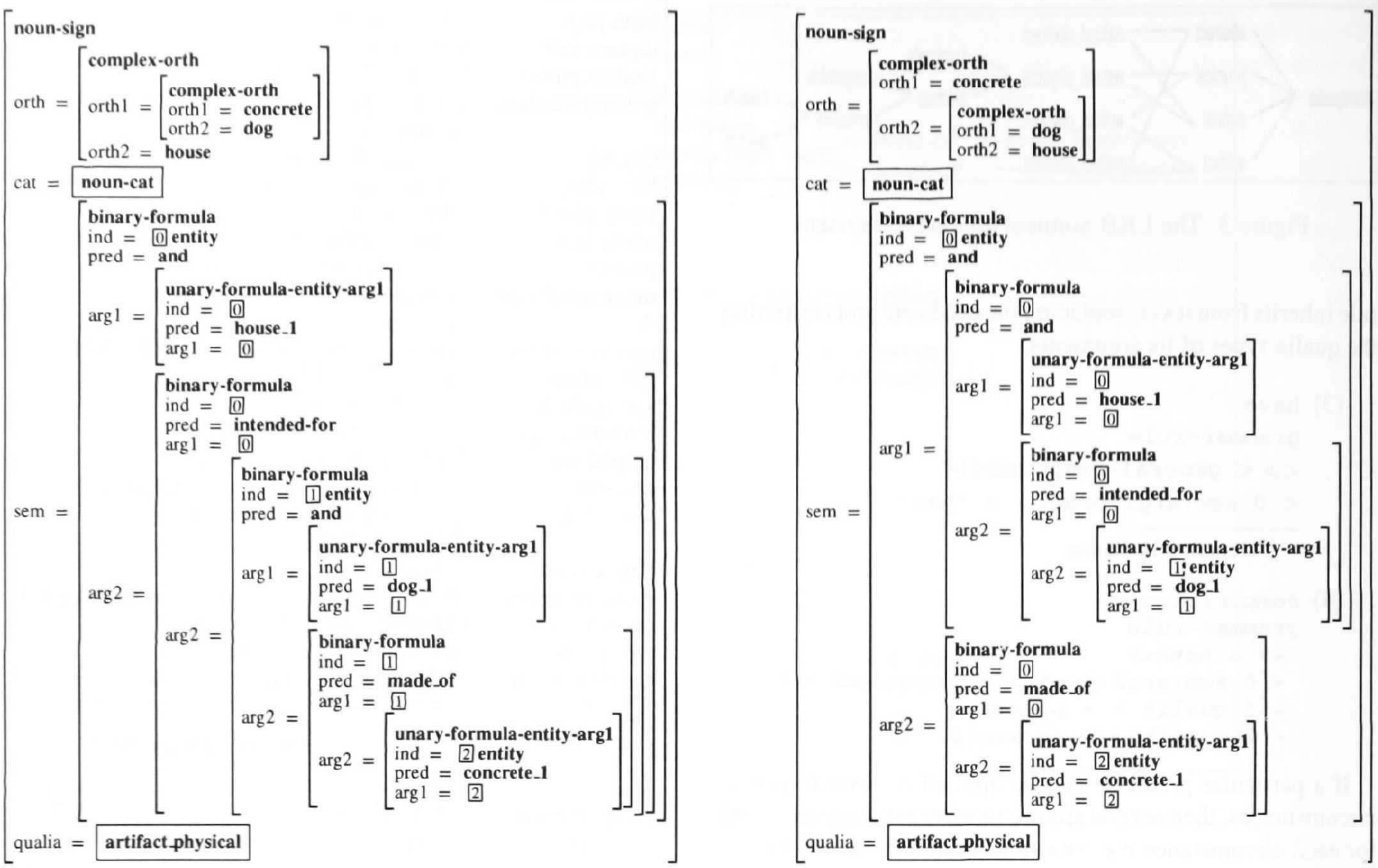

Figure 6: Feature structures for two interpretations of *concrete dog house*

becomes clear that the nominal predicates will most often be the most appropriate of the result set. In the set of results for *ocean study*, for example, it could be argued that ABOUT is as valid a rule as PRODUCT, and indeed there does not seem to be much to distinguish between a study about the ocean, and a study of the ocean. The latter, however, seems neater both linguistically and computationally since it involves just the predicate and a single argument, and so the nominalised predicate has been marked as the most appropriate in this instance. The other possibilities in the set are generated since some normal rules could also apply to the nominal elements. In the case of *city employee*, for example, the other rule applications are quite plausible interpretations for the compound, except for MADE-OF2.

Multiple compounds can also be interpreted, although the overgeneration problem becomes more pronounced. Since the ambiguities are multiplied, the compound *concrete dog house* was assigned 15 different interpretations. Figure 6 shows feature structures for 2 of the possibilities for the interpretation of this compound. The feature structures also demonstrate how the predicate is associated with its arguments in the semantics of the compound entity.

All together, some 60 different binary compounds were analysed, and none was assigned a set of predicates of more than six members. The average size of the predicate set was three for conventional compounds, and four for those compounds that involved nominalisation rules. There is the possibility that this latter figure could be reduced to unity by assigning more weight to nominalisation predicates in the set when they occur.

Since the compounding rules are implemented in the normal grammar of the LKB parsing system, it is very straightforward to include nominal compounds in normal parsing examples, e.g. the sentence *John admired the young Cardiff flower shop manageress* was successfully parsed to incorporate a correct interpretation of the compound, albeit with a high degree of ambiguity. This is of course due to the highly complex nature of the compound.

Discussion

The original requirements of the study have been satisfied: a system of general rules has been implemented that will generate a set of predicates to relate the elements of a nominal compound. For normal compounds, this set invariably seems to contain the most appropriate predicate for the compound, thus justifying the approach. The set of rules has been based upon those predicates given in Levi's work, but these have been greatly extended, improved and organised into a hierarchy.

When combined with more conventional parsing, an undesirable degree of ambiguity occurs, since every possible predicate will give rise to a seperate parse, or set of parses. Therefore there is an argument for carrying out the analysis of nominal compounds with suitable pragmatic disambiguation separately from the parsing process, either before or after the main parsing occurs. Complete separation of compound analysis from parsing could however be undesirable since there may be circumstances where there it is ambiguous whether two words are a compound or are separate.

No claims are made for the precise formulation of the compounding rules used in this study, or even their extent. The qualia structure in the LKB is extremely small, and so many rules have been made unnecessarily general. With a more precise semantic qualia system and lexicon, more precise rules would be possible, and hence the result sets would contain fewer general or inappropriate predicates. It might also prove possible and advantageous to enlarge the predicate hierarchy suggested here with more specific rules if a more precise qualia system were used. Note that rule hierarchies need not necessarily be based on Levi's (1978) predicates — these were used as a basis in the current study, since they were appropriate. If it proves advantageous to add to them, or to replace the functionality of one with other rules elsewhere in the hierarchy, then this should be done. The main purpose of the current study is to develop a novel methodology, not a canonical rule hierarchy.

There would obviously be fewer ambiguous or inappropriate analyses if the compounding information were lexicalised. The argument for not mounting the lexicalisation bandwagon, however, is that it introduces a huge overhead on lexicon size and complexity, greatly complicates the interaction of grammar and lexicon (since it is here that we have to relate information from the compound elements) and requires modifications to the system that the analysis in implemented in (since e.g. some complex disjunctive unification operations would be necessary). In addition it will prove impossible, in practice, to include all the possible compound relations in a lexical entry. However, lexicalisation should not be abandoned altogether. There will always be exocentric, metaphorical and unanticipated uses of compounding, and whilst it is possible to process some of these in a rule-based system (as the 'Cow-Phone' example showed) these instances are, in the main, better handled through lexicalisation. Thus a rule-based system such as the one described here is only more suitable for processing the endocentric nominal compounds, which constitute the majority of those encountered. For optimal performance, however, such a rule-based approach should be complemented by a modest number of lexicalisations for the more unusual compounds.

Whilst not a perfect implementation, the system described in this study has shown the advantages and possibilities of interpreting nominal compounds via a set of possible predicates relating the nominal elements, and has shown how such a set of predicates can be derived from the nature of the compound elements. This represents a very novel approach to the problem of extracting the meaning of a nominal compound from its constituents, and although the system's performance as described is less than perfect, the possibilities are almost infinite, given a larger and more specific qualia system and lexicon. Through judicious formulation of compounding rules, and use of a good qualia/semantic description, we should be able to interpret a very much more wide and general range of nominal compounds than has hitherto been possible, and derive more detailed and useful interpretations.

Acknowledgements

This paper is based upon a thesis submitted for the degree of Master of Philosophy at the University of Cambridge (Jones, 1992). The research was carried out under a grant from the (UK) Science and Engineering Research Council. Thanks for instructive and helpful comments to Ann Copestake, John Carroll, Andrew Fordham and anonymous reviewers. The author is currently supported by a grant from the (UK) Economic and Social Research Council.

References

Bouillon, P.; Bösefeldt, K.; and Russell, G. (1992). Compound Nouns in a Unification-based MT System. In *Proceedings of the Third Conference on Applied Natural Language Processing* (pp. 209–215). Trento, Italy.

Copestake, A. (1992). The Acquilex LKB: representational issues in semi-automatic acquisition of large lexicons. In *Proceedings of the Third Conference on Applied Natural Language Processing* (pp. 88–92). Trento, Italy.

Copestake, A. (1993). The Compleat LKB: Acquilex-II Deliverable 3.1. Technical report no. 316. Cambridge, UK: Cambridge University Computer Laboratory.

Downing, P. (1977). On the Creation and Use of English Compound Nouns. *Language*, 53, 810–842.

Finin, T.W. (1980). The Semantic Interpretation of Nominal Compounds. In *Proceedings of the First National Conference on Artificial Intelligence* (pp. 310–312).

Hobbs, J.R. and Martin, P. (1987). Local Pragmatics. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence* (pp. 520–523).

Isabelle, P. (1984). Another Look at Nominal Compounds. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the ACL* (pp. 509–516).

Jones, B. (1992). Predicting Nominal Compounds. MPhil thesis. Cambridge, UK: Department of Engineering, University of Cambridge.

Larson, G. (1989). *Night of the Crash-Test Dummies*. London, UK: Futura.

Levi, J.N. (1978). *The Syntax and Semantics of Complex Nominals*. New York: Academic Press.