# The Analysis and Acquisition of Proper Names for Robust Text Understanding

Sam Coates-Stephens
Department of Computer Science
City University

October 1992

This thesis is submitted as part of the requirements for a Ph.D. in Computer Science in the Department of Computer Science of City University, London, England.

# Contents

*Part I* LEXICAL ACQUISITION

*Part II* ACQUISITION AND ANALYSIS OF PROPER NAMES

*Part III* EVALUATION

APPENDICES

# List of Figures

# List of Tables

# Acknowledgements

Various people have been of immeasurable help to me throughout the course of this thesis.

First and foremost is my supervisor Lee McCluskey. He has been constantly available to offer help, advice, and encouragement. His comments on papers I have written, on the nature of the research project and manner of pursuing it have been invaluable.

Other members of the City University have offered helpful advice. In particular I would like to thank Alan Burton for reading through the entire first draft of the thesis and providing excellent feedback.

Working in an unexplored area of a relatively new field has meant that I have had to search far afield for researchers pursuing similar topics. I have been very fortunate in finding many such individuals with whom I have been able to engage in invaluable dialogues. Chief among these are David Lewis, Robert Kuhns, Robert Amsler, Marti Hearst, and Dave McDonald. Yorick Wilks, Don Walker and Geoffrey Leech have offered genial encouragement. I would particularly like to thank David Powers for early encouragement and for making an IJCAI travel award available for me to attend IJCAI '91.

Finally, I wish to thank my girlfriend Mary for not letting me forget the things which really matter.

# ABSTRACT

In this thesis we consider the problems that Proper Names cause in the analysis of unedited, naturally-occurring text. Proper Names cause problems because of their high frequency in many types of text, their poor coverage in conventional dictionaries, their importance in the text understanding process, and the complexity of their structure and the structure of the text which describes them. For the most part these problems have been ignored in the field of Natural Language Processing, with the result that Proper Names are one of its most under-researched areas.

As a solution to the problem, we present a detailed description of the syntax and semantics of seven major classes of Proper Name, and of their surrounding context. This description leads to the construction of syntactic and semantic rules specifically for the analysis of Proper Names, which capitalise on the wealth of descriptive material which often accompanies a Proper Name when it occurs in a text. Such an approach side-steps the problem of lexical coverage, by allowing a text processing system to use the very text it is analysing to construct lexical and knowledge base entries for unknown Proper Names as it encounters them. The information acquired on unknown Proper Names goes considerably beyond a simple syntactic and semantic classification, instead consisting of a detailed genus and differentia description.

A complete solution to the 'Proper Name Problem' must include approaches to the handling of apposition, conjunction and ellipsis, abbreviated reference, and many of the far from standard phenomena encountered in naturally-occurring text. The thesis advances partial and practical solutions in all of these areas.

In order to set the work described in a suitable context, the problems of Proper Names are viewed as a subset of the general problem of lexical inadequacy, as it arises in processing real, un-edited, text. The whole of this field is reviewed, and various methods of lexical acquisition compared and evaluated.

Our approach to coping with lexical inadequacy and to handling Proper Names is implemented in a news text understanding system called FUNES, which is able to automatically acquire detailed genus and differentia information on Proper Names as it encounters them in its processing of news text. We present an assessment of the system's performance on a sample of unseen news text which is held to support the validity of our approach to handling Proper Names.

# Chapter 1

# Introduction

In recent years the prospect of natural language analysis of unedited naturally-occurring text has become far more possible than anyone might have dreamed five or ten years ago. By analysis we mean the processing of a text to enable the processor to determine key facts within that text. A term that is becoming more frequent to describe such activity is information or fact extraction [93, 92, 11].

One of the major obstacles to such unrestricted textual analysis is lexical inadequacy. To perform well on a piece of text, any system utilising Natural Language Processing (NLP) style methods must have lexical entries for every word it encounters in the text. To attempt to fulfill this goal — to overcome the lexical bottleneck — has emerged as one of the great challenges for NLP communities in the 1980's and 90's. Despite the impressive work performed utilising Machine Readable Dictionaries (MRD's) [20, 47] it remains a fact that complete lexical coverage is simply not possible. The great number of new words arising, of new coinages, slang expressions, and especially of Proper Names, means that any lexical resource aimed at unrestricted text will always be inadequate. Therefore, a robust NLP system must consider ways of handling words not in its lexicon.

While work on coping with 'conventional' unknown words is relatively well-established ([69, 183, 111]), that class of words which is the greatest potential contributor to the problem — the class of Proper Names — has been relatively ignored. This is of particular concern, not only because Proper Names will so often be unknown and are so frequent in normal text, but also because they are so important in information extraction. In the analysis of news text many of the key items of information are given as Proper Names. In addition, much of the text is given over to describing the Proper Names that occur within it.

This thesis seeks to address the lack of work on Proper Names in text understanding. We present an in-depth description of the structure of Proper Names and the context which commonly surrounds them, and utilise this description in a simple news text understanding system. This system takes the analysis of Proper Names as the focal point in its analysis of news text. It is able to use the descriptions that occur within a text to produce lexical and Knowledge Base entries for the unknown Proper Names it encounters. This information emerges as a by-product of the text understanding task, its acquisition requiring adaptation of standard NLP-methods, rather than radically new approaches.

## 1.1 The Problem of Lexical Gaps

As NLP has moved from being a purely research-based field to encompass many applied aspects, it has had to begin to face up to the problems produced in analysing naturally-occurring, unedited text, as opposed to artificially constrained examples which are always within the lexical, grammatical and semantic capabilities of the system. The terms 'Robust NLP' [29, 88] and 'Partial Parsing' [120, 174, 82] have been used to describe the nature of systems which seek to analyse real text. As stated above, one of the major problems that such text produces is the problem of lexical coverage — real text demands a large vocabulary. Researchers in applied NLP have had to consider ways of increasing the tiny lexicons (of several hundred entries) of yester-year into the vast repositories (of many thousand entries) needed for processing real text.

The majority of NLP systems all use some sort of syntactic and semantic analysis stages, whether these be entirely separate or heavily inter-linked. The syntactic parse of a text requires that all the words in a sentence be labelled as to their part of speech, to enable the system's grammar rules to be applied. If a word is not known, (a situation Zernik [180] has described as encountering a 'lexical gap') a parse can not be completed. Even if a syntactic category can be created, the semantic representation produced for a sentence will be impoverished (or fail) if the semantic category of a word is not known.

A simple approach to overcoming this problem would seem to be to ensure it does not happen, by providing the system with a complete lexicon. A common approach to this has been to utilise Machine Readable Dictionaries to create large NLP-style lexicons. Other approaches [169, 23, 83] have attempted to extract lexical information from large text corpora. Although this work can serve to reduce the problem of encountering unknown words, it can never remove it, as no lexicon can ever be complete.

Therefore, an essential provision for a robust NLP system is the ability to cope with unknown words. This requirement has been stressed in many papers describing systems aimed at understanding real text, e. g. [117, 56, 182, 113, 29, 9]. Early approaches utilised grammatical constraints to derive a part of speech for an unknown, and semantic/contextual constraints to derive a semantic category. Although feasible in constrained domains, where a Knowledge Base provides complete information (as was the case for the Schankian script-based analysers such as SAM, PAM and BORIS), such an approach is less feasible in un-restricted areas. More recent approaches [183, 87, 123, 56] have relied heavily on morphology to relate an unknown word to a known root, or to provide a hypothesis as to its syntactic and semantic category. Coupled with a large lexicon such a strategy greatly reduces the problem of encountering unknowns. However, it does little to cope with the greatest source of lexical inadequacy — Proper Names.

## 1.2 The Problem of Proper Names

The majority of text understanding work looking at real text has focused on news items [96, 78, 104, 68, 51]. This is partly explained by its free availability, and partly by the interest of major funding agencies in the sort of information it contains (specifically business, political and foreign affairs information). A defining feature of this text is its event-oriented nature — it describes who did what, who they did it to, and where (and when) they did it. A large part of this information will be conveyed by Proper Names. Two of the major applications of work analysing news text have been topic classification and information extraction (or database creation). In both of these tasks the accurate and efficient processing of Proper Names is vital. [1] In topic classification, Proper Names provide a good index for classification, based on a company's field of activity, or a person's

[1] This contention has been supported by N. Sondheimer in his closing address at the 3rd Conference on Applied NLP [14]. Professor Sondheimer highlighted work on Proper Names as one of the areas in which more work should be done, stressing its important and under-researched nature.

group affiliation. In information extraction, Proper Names will convey the names of the main actors, and the location of the events described. For example, in scanning a story about a company takeover the main items to be extracted will be the two companies concerned (and maybe the price paid). In scanning a story about a terrorist attack, the main items to be extracted will be the perpetrator(s), targets and location (and maybe damage caused).

Even if one is not aiming for such a high level of understanding, the accurate parsing and semantic analysis of real text will demand that all the words in the input are known, or can be handled by effective unknown analysis procedures. Proper Names comprise a reasonable proportion of news text (up to 10%, [45]), and given their open-ended nature many of the Proper Names encountered will be unknown. Consideration must be given to how to cope with these if unrestricted text is to be parsed successfully.

The problem of lexical coverage of Proper Names is worsened by their absence (or low level of coverage) in the majority of conventional dictionaries, as shown in [172, 146, 130]. We therefore have a phenomenon that is :

- Very frequent in real text, especially news text.
- Very important for accurate text understanding.
- Very poorly represented in most lexical resources

These, however are not the only problems that Proper Names present. As a few researchers have noted ([30, 10]) Proper Names can occur in 'variant forms'. We can talk about 'the Atlantic' and 'the Atlantic Ocean'. We can talk about 'George Bush', 'Mr. Bush, 'President Bush' and so on. We can talk about 'Midland Bank plc' and 'Midland Bank' and 'the Midland'. The rules operating here are of interest for two reasons. Firstly they must be known to a text understanding system if it is to correctly process all these forms and to comprehend that they all refer to the same thing. Secondly, they describe a very neglected linguistic phenomenon, which up to now has received little detailed description.

The existence of these variant forms also demonstrates the impossibility of a 'super dictionary' solution to the Proper Name problem. A lexicon intended to cover all the different Proper Names that one might encounter in a large sample of news text would have to be immense (Smith, cited in [161], estimates the number of unique surnames in the US alone to be around 1.5 million, and the number of companies in the world will greatly exceed this). If it were then to cope with these variations, each entry would also have to have several alternative forms.

Further problems arise from lexical ambiguity (illustrated by words such as 'Bush' and 'Major'). Conventional lexical ambiguity is already a recognised problem in NLP, but the fact that in theory any word can be a Proper Name makes the situation many times worse. The simple solution of utilising initial letter case to choose between the Proper Name and the non Proper Name definition is not feasible, as many common words occur with an initial capital and yet retain their common meaning (e. g. the component words of the Proper Name 'Centre for Media Studies'), while others lose their common meaning (e. g. 'Terry **Butler**', 'the town of **Red Bridge**'). The resolution of this problem requires a sound understanding of the nature of Proper Names, and their surrounding context.

These phenomena have been neglected not only in computational linguistics, but also in linguistics as a whole. Although the age-old controversy over whether Proper Names have meaning in the same manner as other lexical items has been debated ad infinitum, there has been little inquiry into the syntax of Proper Names and very little work at all on any class of Proper Name beyond the personal name. ([30, 5, 115] provide a few exceptions). Yet real text abounds in all sorts of names (places, objects, companies, events, films, drugs, illnesses etc), which possess an immense variety of structure.

Above and beyond all the problems that stem from the Proper Name itself, there are also problems that stem from the syntactic context in which it occurs. This provides yet more obstacles to the 'super dictionary' position, in that even if we do know all the words in an input, the complexity of the constructions in which Proper Names can occur still renders their analysis problematic. For example, we might know the words 'Harold' and 'Wilson', and even the compound 'Harold Wilson', but this does not solve all our problems in attempting to analyse a phrase like:

> 'Harold Wilson, former Labour leader and British Prime Minister, now Lord Wilson of Rievalux, addressed the European Parliament on ... '

Proper Names commonly occur as, or in, a variety of extremely complex constructions, such as:

- Apposition, e. g. 'Azerbaijan's new Foreign Minister, Tovig Gasimov,'
- Complex Noun Groups, e. g. 'Popular Front leader Abulfez Elchibey'
- Conjoined Noun Phrases, e. g. 'Conference for Security and Co-operation in Europe', 'the war-torn Yugoslav cities of Sarajevo and Dubrovnik'
- Prepositional Phrase strings, e. g. 'the Press Syndicate of the University of Cambridge'

Apposition is a particularly under-investigated construction in Computational Linguistics. Although conjunction has received much investigation, the problems presented by Proper Names, in particular company names, in the analysis of conjunction have been almost totally ignored. Additionally, the interaction of conjunction and apposition involves many complicated attachment problems, as shown in the example below:

> '... Its performance contrasted with Vauxhall, the UK subsidiary of General Motors of the US, Rover, the British Aerospace subsidiary, and Peugeot-Talbot, the UK subsidiary of Peugeot of France.'

## 1.3 Solutions to these Problems

It is our contention that consideration of the problems of unknown words, and of Proper Names in particular, should be at the centre of any system undertaking robust NLP of real text.

We have constructed special purpose solutions for the handling of Proper Names, and general purpose solutions for the handling of 'normal' unknowns. In the event of the special purpose solutions failing, then the procedures for handling normal unknowns can equally well be applied to unknown Proper Names.

Our solutions to handling normal unknowns build on previous work in the area, utilising constraints from all levels of language processing (morphological, syntactic, semantic and pragmatic) to derive a syntactic and semantic category for an unknown word. The derivation of a semantic category is viewed as an incremental process, whereby a single category can eventually be derived over several encounters.

Our solutions to the handling of Proper Names form the heart of the thesis. It is consideration of the context in which Proper Names occur which provides the key to our solution. We have observed that Proper Names rarely occur in isolation in a text. Most of the time they occur together with a neighbouring word, or an apposition Noun Phrase, which provides a description of the Proper Name. Thus we might see:

- Bhoutros Bhoutros Ghali, the new Secretary-General of the UN,
- former ICI boss John Harvey Jones
- Olivetti, the large Italian electronics group,
- Ferranti Inc
- the Iraqi town of Zhako
- Southern Prisoners Defense Committee
- the Black Sea

We can see that the very text in which a Proper Name occurs provides a description for it. Thus the need for a large Proper Name lexicon is virtually obviated — the text under analysis can act as a highly-tuned lexicon, in that it will describe many of the Proper Names it contains. The beauty of this phenomenon is that the same mechanisms that analyse the text to discover its meaning can be used to produce a description for the Proper Names within the text. These descriptions are language, like any other part of the input, and thus they must be analysed by any system that is claiming to analyse the text in which they occur. In a way, therefore, it is the very complexity of the Proper Name that provides a solution to their analysis.

We advocate the use of these 'within text descriptions' to overcome the problem of poor lexical coverage of Proper Names. (If a Proper Name occurs unaccompanied then it will be processed by the general-purpose unknown handling heuristics.) The use of these descriptions, however, is dependent on the existence of syntactic and semantic rules to analyse them, and also on the existence of rules to analyse the complex nature of the names themselves, which can comprise several syntactic constituents. We have analysed seven major classes of Proper Name which occur in news text. Each class is described at both a syntactic and semantic level. Some classes of Proper Name (in particular personal names) have no meaning in themselves, and so a description is commonly provided by a separate Noun Phrase or a neighbouring common noun, as shown in the first five examples above. Other classes of Proper Name have a clear meaning (as shown in the last two above examples), challenging the prevalent linguistic view that no Proper Names have meaning. These names are self-describing, and so have less need of additional description. The syntactic and semantic description we have produced is utilised to create the rules needed for the analysis of the names and their surrounding context.

With these techniques we are able to analyse all the Proper Names encountered, and their accompanying descriptions, utilising the same basic mechanisms that are used to analyse the rest of the text. Proper Names are seen as varieties of complex noun groups (e. g. 'former ICI boss John Harvey Jones', 'Southern Prisoners Defense Committee'), or Noun Phrases ('the Iraqi town of Zhako', 'the League against Cruel Sports'). Apposition is analysed as an abbreviated copular sentence. Such an approach enables us to analyse even the most complex names and descriptive constructions, and to accurately extract agents, locations etc.

Finally, we present special purpose heuristics for handling variant forms, enabling all the alternative forms of a Proper Name to be understood as referring to the same thing, e. g. 'Southern Prisoners Defense Committee' and 'SPDC', 'Ferranti Inc' and 'Ferranti'.

All of the above work is implemented in an NLP system called FUNES (after Borges 'Funes the memorious' [22], but also an acronym for 'Figuring-out Unknown Nouns from English Sentences'). This system provides the context in which all of our work on unknown words and Proper Names takes place. FUNES is implemented in C-Prolog, version 1.5+, and runs on a Solbourne mini-computer. It analyses short news stories into a case-based semantic representation, and at the same time updates its lexicon and Knowledge Base with definitions for the unknown words, and in particular Proper Names, that it has encountered. The FUNES system demonstrates the viability of our approach in a very tangible way.

## 1.4 Outline of the Thesis

The thesis is divided into three parts. Part One (comprising chapters two, three and four) considers the general problem of lexical inadequacy, and describes the solutions to it adopted in FUNES. In chapter two we review previous work on word learning and other attempts to overcome the lexical bottleneck. This chapter describes lexical acquisition as a field in its own right, combining work from text processing, Machine Readable Dictionary research, corpus linguistics and Machine Learning of Natural Language. Chapter three introduces the FUNES system, and provides a brief description of its architecture and functioning. This is intended to make subsequent chapters which refer to aspects of the system more comprehensible. The final chapter in Part One describes our solutions to handling unknown words, and shows how FUNES makes use of all levels of processing — morphological, syntactic, semantic and pragmatic — to produce a syntactic and semantic category for unknown nouns.

Part Two, concentrating on the problems of Proper Names, is the main body of the work. Chapter five describes previous work on Proper Names, both linguistic and computationally linguistic. The overall conclusion is that there exists no sound theory of the syntax and semantics of Proper Names, and that the majority of existing work has been done very much in an 'as-needed' fashion, to cope with particular problems encountered by particular systems. In chapter six we present our solution to this problem, in the form of a detailed and formal description of the syntax and semantics of the major classes of Proper Name encountered in news text — people, places, corporations, legislation, information sources (such as newspapers and books), events, and objects. Chapter seven shows how this model is used and implemented in the FUNES system to enable it to successfully process a large majority of the Proper Names it meets in its processing of unedited news text.

The following three chapters look at the major categories of Proper Name on a category-by-category basis, examining particular problems they each give rise to, and the solutions we have produced. Chapter eight looks at personal Proper Names, considering the processing of apposition, its interaction with conjunction, and the differences in handling known and unknown personal Proper Names. Chapter nine looks at corporation and legislation Proper Names, considering the variant form problem, and difficulties raised by names that include conjoined Noun Phrases and strings of Prepositional Phrases. Chapter ten looks at the remaining categories of Proper Name, with particular focus on place names, again considering the variant form problem and problems of common noun/Proper Name ambiguity.

The final part of the thesis presents an evaluation of the methods described in the preceding chapters. This evaluation consisted of a test run of 200 unseen and unedited news stories through the FUNES system, with a view to assessing the performance on

Proper Name handling. This produced a best performance of 77% of unknown Proper Names correctly assigned to one of the seven above categories, and a worst performance of 66%.

We conclude with a comparison of our approach and related work, and a discussion of possible extensions and improvements to the system.

Extensive appendices give more detailed information on the FUNES system, and detailed examples of the major categories of Proper Name and the text used in the final evaluation.

Figure 1.1 gives a list of the abbreviations used throughout this thesis. Readers may find it useful to refer to this figure when an abbreviation is used with which they are not familiar.

| | |
|---|---|
| AmbN | AMBIGUOUS NOUN |
| CC | CAPITALISED CONSTITUENT |
| CF | CASE FRAME |
| Corp PN | CORPORATION PROPER NAME |
| Isource PN | INFORMATION SOURCE PROPER NAME |
| KW | KEY WORD |
| LAR | LEXICAL AMBIGUITY RESOLUTION |
| Legis PN | LEGISLATION PROPER NAME |
| MLNL | MACHINE LEARNING OF NATURAL LANGUA |
| MRD | MACHINE READABLE DICTIONARY |
| NDF | NAME DESCRIBING FORMULA |
| NG | NOUN GROUP |
| NLP | NATURAL LANGUAGE PROCESSING |
| Noun_comp/Ncomp | NOUN COMPLEMENT |
| NP | NOUN PHRASE |
| PN | PROPER NAME |
| PNcon | PROPER NAME CONSTITUENT |
| Pnoun | PROPER NOUN |
| PP | PREPOSITIONAL PHRASE |
| RC | RELATIVE CLAUSE |
| SR | SELECTIONAL RESTRICTION |
| TU | TEXT UNDERSTANDING |
| VP | VERB PHRASE |
| WFF | WELL-FORMED FORMULA |
| WTD | WITHIN-TEXT DESCRIPTION |
| WSJ | WALL STREET JOURNAL |

Figure 1.1: Abbreviations used in the Thesis

# Chapter 2

# Lexical Acquisition: Solutions to the Lexical Bottleneck

## 2.1 Introduction

Lexical Acquisition is a recent and still small topic, settling itself somewhere within Computational Linguistics, Computational Lexicography, and Text Processing. Over the past two years the term has started to emerge as describing a recognised field — that which seeks to provide lexical resources for NLP systems. This work could also be described as that which is attempting to overcome the 'lexical bottleneck', a term used by Wilks [177] to describe the problem which lack of lexical resources causes existing NLP technologies, and the problems of getting such resources into NLP systems. Figure 2.1 below shows the work and motivation which has lead to lexical acquisition emerging as a thriving and much needed topic.

This chapter presents an overview of this work and shows how it has led to, and how it is related to, the field as it now stands.

The term lexical acquisition is used to describe a variety of methods for overcoming the lexical bottleneck. Although all aimed at the same problem these do differ in their methodology, and can be usefully distinguished. Firstly there is work dedicated to building an initial lexicon. This creates the initial lexical resource that any NLP system needs to perform its task. For the creation of realistic and sizable lexicons this work is mainly associated with the use of Machine Readable Dictionaries, or MRD's. (However, in more limited domains semi-automated hand encoding is still used.) More recently such work has also been undertaken with the use of large text corpora. A good over-view of this type of work is provided in [181]. The second method is concerned with actual 'on the spot' word learning, equipping NLP systems with the ability to learn a new word as they encounter it in their processing. So, on one hand we have work directed towards the creation of an initial resource, and on the other work directed towards the acquisition of further resources to make good the deficiencies of the initial resource. The term I shall use for the former is lexical pre-processing, the latter will be called word learning, or more generally it can be seen as an aspect of robust NLP. The FUNES system is concerned with the second type of lexical acquisition.

There have been two major sources of motivation for research into word learning, one theoretical and one practical. On the theoretical side motivation has come from the study of human word learning. Various researchers have constructed models based on human word learning theories, to further explore this topic and assess the strengths

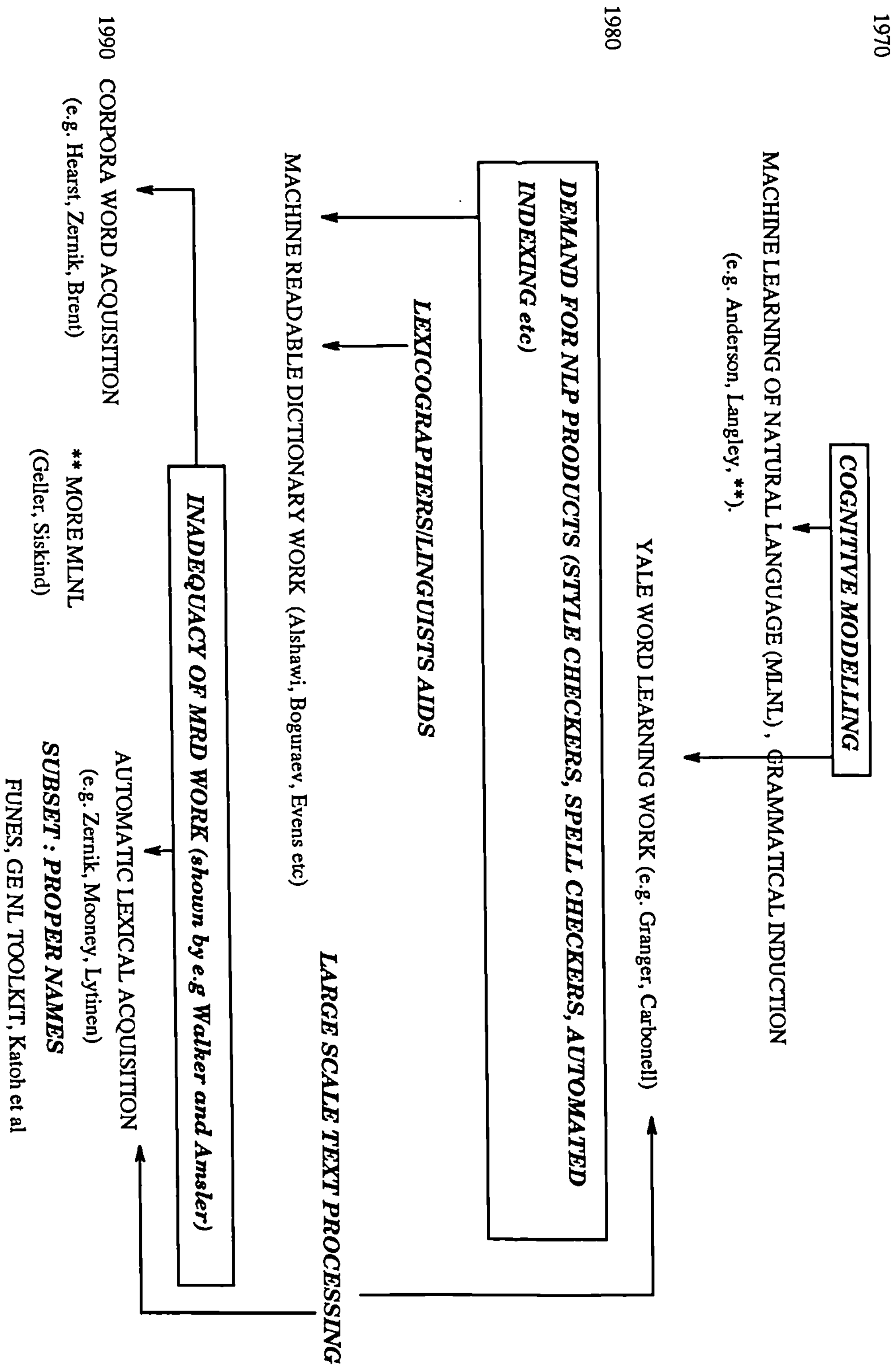

1970
COGNITIVE MODELLING
MACHINE LEARNING OF NATURAL LANGUAGE (MLNL) , GRAMMATICAL INDUCTION
(e.g. Anderson, Langley, **).
YALE WORD LEARNING WORK (e.g. Granger, Carbonell)
1980
DEMAND FOR NLP PRODUCTS (STYLE CHECKERS, SPELL CHECKERS, AUTOMATED INDEXING etc)
LEXICOGRAPHERS/LINGUISTS AIDS
MACHINE READABLE DICTIONARY WORK (Alshawi, Boguraev, Evens etc)
LARGE SCALE TEXT PROCESSING
INADEQUACY OF MRD WORK (shown by e.g Walker and Amsler)
1990 CORPORA WORD ACQUISITION
(e.g. Hearst, Zernik, Brent)
AUTOMATIC LEXICAL ACQUISITION
(e.g. Zernik, Mooney, Lytinen)
** MORE MLNL
(Geller, Siskind)
SUBSET : PROPER NAMES
FUNES, GE NL TOOLKIT, Katoh et al

Figure 2.1: The Background to Lexical Acquisition

and weaknesses of competing accounts of how this incredible feat is achieved. On the practical side is the need for robust NLP systems to be able to cope with new words, and to automatically expand their lexicons. We begin by looking at some of the work on modelling human word learning, work usually carried out in a field known as Machine Learning of Natural Language, or MLNL. We shall see that many of the approaches taken in this area cannot be practically applied by text processing systems. An approach taken at Yale in the later 1970's, however, was based on adult learning of unknown words in context and yet was successfully applied to enabling a script processing system to learn meanings for unknown words. The Yale approach is considered next, before we then proceed to look at work dealing with lexical pre-processing, both with MRD's and text corpora. While such work has had success it does not obviate the need for automatic word learning, as we show in the section on studies assessing the lexical coverage of MRD's. This naturally leads to a final examination of more recent word learning research.

## 2.2 Machine Learning of Natural Language

The majority of work in MLNL has been concerned with the acquisition of syntactic knowledge. The field of grammatical induction is extensive, and dates back to the 1960s; it is not, however, of relevance here, as we are concerned with word learning rather than learning of syntactic rules. However some of the work on general language learning has also looked at word learning. To achieve this, systems have commonly attempted to pair linguistic input or built-in concepts, with some sort of simulated visual input. Thus they seek to model the language acquisition of a human infant.

Some of the earliest such work is due to Larry Harris [76]. He presented a system which attempted to model the conditions under which a child learns Natural Language. It consisted of a simulated robot which walked around a room, attempting to acquire information about the room and its contents. It achieved this by matching built-in concepts with a linguistic description of its actions, given by a human 'teacher'. A system described by Anderson in [12, 13] accepted a string of words and corresponding scene descriptions encoded as associative networks. However, his program started at the point where links between words and concepts have already been established. Although Anderson claims the later system is able to acquire word meanings, it is not clear how it does this, and the emphasis is still very much on acquisition of syntax.

Robert Berwick produced a system for the acquisition of syntactic knowledge, [18], based on Marcus' deterministic parser Parsifal [114]. This was able to infer the grammatical category of unknown words based on the grammatical constraints of X Bar theory. An addition to the system, described in [17] was also able to learn word meanings, through a learning principle of analogical matching. This analysed stories into causal network descriptions (basically semantic networks), with objects, agents and qualities serving as the nodes and verbs serving as the links between the nodes. The learning was achieved by matching the part of the network which contained the unknown verb to a set of such networks derived from similar stories containing only known words.

Other important work in this area has been carried out by Selfridge [152], whose CHILD program modelled the language acquisition — learning both word meanings and syntax — of a human infant.

More recently Geller et al, [66, 62] have presented a rudimentary system that seeks to match simulated visual input with linguistic input to acquire word meanings. The work is notable due to its learning through two approaches - single channel and multi-channel (a similar distinction is drawn by Powers in [134]). The multi-channel approach uses both linguistic and simulated visual input to acquire the semantics of concrete nouns, and to ground this semantics in the real world. Thus a simple image of a table may be associated with the sentence 'this is a table'. Assuming all the words except table are known a semantics for table can be acquired. The single channel approach seeks to explain the learning of older children who learn solely through spoken or written explanations. Thus the sentence 'cats and dogs are animals' will lead to a rudimentary semantics for 'animal' if all the other words are known. Assuming that cats and dogs have been learned about through the multi-channel approach the word animals is also grounded in perceptual reality. We will discuss this work further in chapter 11 when we compare it to the FUNES approach to acquiring Proper Names (hence PN's), as there are some interesting similarities between the two.

Finally, Siskind has presented a series of papers [155, 156, 157] that describe a variety of learning, from a combination of linguistic and simulated visual input. The visual input is simulated by series of logical wff's which describe a visual representation of the associated sentence. A set of inference rules analyse the wff's into global 'Event' or 'State' primitives (based on Jackendoff's theory of Conceptual Structure, described in [91]). This input is then matched to a series of parsed sentences, and over a series of trials the words are matched to their correct semantic primitives.

The motivation of such work is the simulation of human learning, and not practical applications. Thus although it is impressive in its results and shows that much can be learned by computational methods, it appears that its utility is limited in practical situations. The needs of the text processing community are to rapidly acquire large lexicons, and to be able to deal efficiently with encountering lexical gaps. The above approaches deal mainly with learning from scratch, and thus require many examples or rely heavily on some kind of tutor. But if we attempt to learn from an existing large body of knowledge it seems feasible that more could be learned from fewer examples, and therefore it could be learned more quickly.

## 2.3 The Yale Approach to Automatic Word Learning

Such an approach was taken in the late 1970s at Yale, where Richard Granger and Jaime Carbonell constructed learning models based on ideas of adult learning, specifically on how an adult constructs a meaning for an unknown word encountered while reading a piece of text. The late 1970s at Yale saw some of the first attempts at deeper understanding of realistic language, based around the high level knowledge structures called scripts and plans.

The first such system was called SAM [48], for Script Applier Mechanism. This read short news stories and attempted a detailed understanding of them. It did so by identifying the story as fitting into one of its scripts. Scripts describe a stereotypical situation in terms of its actors, main events and results. If computers were to understand real language it was argued that they must possess scripts which described the context in which this language occurs.

Granger's FOUL-UP system [69] was written as an adjunct to SAM and it activated when SAM encountered an unknown word. It made use of the expectations scripts possess to guess at the meaning of an unknown word in a story. Scripts carry a list of subevents which happen during the course of the script, and restrictions on the actors in these events. If processing a story about a road accident and encountering 'the car struck an elm', FOUL-UP was able to analyse the unknown word as a noun, an object, and something that can play the role of an obstacle in a road accident. This was achieved through the use of syntactic, semantic and scriptal expectations. Syntactic expectations derive from the constraints of the grammar, and enable a word to be classified as to its part of speech. If, for example, an unknown word occurs at the end of a sentence and is preceded by a determiner, then the rules of grammar would constrain it to be a noun. Semantic expectations derive from meaning constraints within a sentence. Given a particular verb only certain classes of noun will make sense as its subject or object. In the above example the meaning restrictions of 'strike' constrain 'elm' to be an object. [1]. Scriptal expectations provide the final refinement on meaning, using the constraints a script places on the objects and actors within it. In this example, 'elm' fills the role of an obstacle in an automobile accident, so this information is added to its definition.

Although it may appear limited, Granger's approach has remained the predominant one in dealing with unknown words in text processing, and FOUL-UP is still one of the most frequently quoted papers in the literature. It may seem that the definition acquired for 'elm' is not at all what humans perceive an 'elm' to be. However, it is the best that can be formed **in this context**. If the sentence had read 'the car struck a boozan', we could form no better an idea for this word than that formed above for elm. The crucial point about the formation of such hypotheses is that they are context dependent. An ability to refine the hypothesis upon future encounters, like that included in the present work, greatly improves the method. As proof of the usability of this approach, the NOMAD system [70] developed by Granger et al in 1983, was used by the US navy for processing ship to shore telexes. NOMAD made use of FOUL-UP's word learning abilities in deriving a meaning for unknown words. It utilised expectations based on syntactic knowledge, and knowledge of the particular domain, to figure out meanings for unknowns. The restricted nature of the domain (ship and submarine movements and activities) was ideal for the FOUL-UP approach.

A year or so later, Carbonell produced a similar mechanism in his news story understander POLITICS [28], that was able to improve on FOUL-UP's limited script based hypotheses through its use of plan-based reasoning. This was the latest knowledge structure to emerge from Yale, and was aimed at overcoming the limited flexibility of script-based analysis. It attempted to consider the goals of actors in a story in order to better understand their actions. Without a flexible ability to monitor the goals and state of the actors in a story, any script would have to be impossibly large to account for all the possibilities that may arise.

Combining the script-based expectations with goal-based ones, Carbonell's system was able to arrive at richer hypotheses. The POLITICS system was a knowledge based natural language understander expert in the realm of U.S foreign policy. It used syntactic and semantic constraints in the same way as FOUL-UP. However it then applied plan-based analysis to infer how the actions described in its input fulfilled the goals of the actors. These inferences helped in deriving meaning for unknowns. The focus was again on nouns. The best way to see how POLITICS worked is to look at an example. The sentence below is taken from [27].

'Russia sent massive arms shipments to the MPLA in Angola'.

where 'MPLA' is unknown. Syntax reveals it to be a noun (as it is preceded by an article and followed by a preposition). Semantics reveal it to be a location or an agent (as it is the object of send). This is perhaps as far as an uninformed human reader could get.

[1] Presumably this very narrow sense of strike is enforced by the 'auto accident' script which is in operation at the time

[2] However, POLITICS then uses its knowledge of the context to try and work out how the action of sending arms can be related to the goals of the Soviet Union or any other possible agents involved. Angola is known to be in a state of civil war, i. e. a state where political factions exercise their goals of taking military and political control of a country. Possessing weapons is a precondition to military actions. Therefore the MPLA my be such a political faction, moreover as Russia is sending the arms it is likely to be a communist faction.

Carbonell also considered the use of multiple encounters with an unknown word to aid in refining the definition, and the learning of metaphorical usage. These facilities were not implemented however. The addition of plan-based analysis made for a more advanced system than Granger's, albeit suffering from the same problems induced by reliance on particular knowledge structures — that to add additional structures would require much reworking and rethinking.

The knowledge structure based systems that emerged from Yale in the late '70s were the first real attempts to deal with large bodies of text at a high level. Their decline was largely due to the realisation that to scale them up to attempt to process unrestricted information at the same level was almost impossible, due to the horrendous complexity of the knowledge structures required. [3] The 1980's saw a search for more unified knowledge structures that could be easily enlarged. This resulted in the work on marker passing which has yet to emerge as a significant field. Apart from the CYC project of Lenat [107], many researchers returned to more limited goals.

The latter part of the '80s has seen a resurgence of interest in text processing, characterised by two things — limited domains and shallow depth of processing. This is interesting in light of the fact that the most successful Yale processor — FRUMP [51] — was characterised by its text skimming as opposed to in depth processing. Although it scanned a variety of domains, its successful realisation as a product in the form of ATRANS [112], was restricted to a single domain, that of bank telex processing.

It is only in the past year or two, as work on realistic text processing has really matured, that the need for work on lexical inadequacy has become more widely appreciated, and the field of lexical acquisition has emerged. Systems prior to the '80s did not need to concern themselves with large lexicons for a variety of reasons. In the early part of the '70s NLP systems did not really exist outside the research labs of the larger universities. While the main concerns were with parsing, semantic representation and control of inference there was no need for large lexicons. When progress was made on these problems to the point where systems could be produced, these were characterised by their extremely limited domains. Again the need for more than a few hundred words at most was non-existent.

The first NLP systems to appear on the market were interfaces of one sort of another. These had no real need for large vocabularies, their domains being limited, and the expected discourse also restricted. Even given this restricted discourse, however, there was still the possibility that a user would type in a synonym which the developers had not anticipated. It did not seem to make sense to equip a system whose main vocabulary would be restricted to a few hundred items (and one of the main requirements of which was speed), with an entire 50,000 word lexicon, just to cope with a few esoteric users.

[2] Curiously, the system appears to make no use of the fact 'MPLA' is entirely in upper case. The FUNES system returns all such unknowns as corporations.

[3] This problem was clearly illustrated in Dyer's description of BORIS [54], a system which attempted to gain a deep level understanding of a few texts. To achieve this level of understanding it had to combine no less than 22 different knowledge structures.

So interfaces instead started to incorporate interactive word learning components, which when presented with an unknown would enter into a dialogue with the user to elicit information on the unknown. Interfaces that make use of this technique are described in [74, 16, 31], and the approach is summarised in [41]. The use of semantic grammars, made possible by the restricted nature of the dialogue, also permitted such systems to get by with limited lexicons.

However, this picture started to change in the '80s for two main reasons (explained below), and this led to the work with MRD's which has characterised much of the research on lexical acquisition.

## 2.4 Work with Machine Readable Dictionaries

The motivation for MRD work can be traced to two sources, one theoretical and one practical. On the theoretical side many post-Chomskian linguistic theories imparted the lexicon with a much larger role than did Chomsky, e. g. Word grammar [89], Generalised Phrase Structure Grammar (GPSG) [64], and Head-driven Phrase Structure Grammar [133]. With an increasing number of NLP systems being based around such theories (in particular GPSG and its derivatives), a lot of the knowledge that in a Chomskian theory resided in the grammar rules, now had to be based in the lexicon. While entering a lexicon of a few thousand words by hand was feasible when this was little more than a list of words, when each entry grows to the size where it almost resembles a small program such a task becomes much less feasible.

On the practical side, computational applications were requiring greater lexical resources. This applied to both those applications aimed at assisting linguistic endeavour (lexicographers aids, computer-assisted language learning) and NLP products (text understanders and summarisers, spell and style checkers, automatic indexers). It was realised that if these were ever to scale up and be useful across a variety of domains, they would require larger vocabularies. Whilst the task of building an NLP program capable of performing successfully across a variety of semantic domains was not feasible, the task of constructing general-purpose syntactic and morphological analysers was feasible, and attractive. One such parser could then be used as the syntactic unit for a variety of applications. The main obstacle was, that to be useful across a variety of domains, such a parser would require a very large vocabulary. Rather than just a few thousand words, tens or even hundreds of thousands of words would be required.

### 2.4.1 Building a Machine Usable Dictionary

The problem of how to get very large lexicons into systems emerged as one of the great challenges to Computational Linguistics in the 1980's. The attraction of MRD's was obvious. On the one hand they offered a ready made source of lexical knowledge — in theory this could save NLP workers the huge task of constructing their own lexicons. In addition, an MRD encapsulated the accumulated experience of many lexicographers over many years, and such lexical expertise is not something a worker on an NLP system might be expected to have. So they offered the prospect of providing lexical resources of vast size and of much higher standard than could be created by NLP workers.

However, as was soon realised, it is not the case that an MRD offers an immediately usable resource as a computational lexicon. Dictionaries are built for humans, and not NLP systems. Even the extraction of syntactic information has met with problems. It might be thought that such information is in a form ideally suited to extraction and immediate use by an NLP system, due to the fact that it requires no deep level of understanding in the way that sense definitions do. However the introduction to [20] provides many examples of the sorts of problems encountered in practice — intermingling of syntactic and semantic information, superficial syntactic information, and the different theories used in the construction of the lexicon which may represent similar information in diverse ways. Additional problems arise from errors on the dictionary type-setting tape.

Turning to the analysis of definitions, the fact that dictionaries are built for humans becomes even more plain. As such, they assume a considerable body of world and linguistic knowledge, which is necessary to understand the definitions they contain. To convert the definitions given in a standard dictionary into definitions usable by an NLP system requires significant processing of the definitions themselves. A common method is to use a pattern matching approach, whereby a number of lexical-syntactic patterns are derived for the dictionary definitions, each pattern having a set of actions associated with it which serve to build the Machine Usable (MU) definition. Below we show an example from [8]:

```
(n-135
        (n +0det +0adj &0noun +noun1
        +that-which *verb-pred &&))
```

This analyses a Noun Phrase (NP) + relative clause as consisting of zero or more determiners, zero or more adjectives, a head noun and optional complement nouns, plus 'that' or 'which' and a verb phrase. If a definition matches this pattern a particular structure-building rule is invoked to build the MU definition. The structure building-rule for pattern n-135 is:

```
(n-135 ((class +noun1
        (noun-mods &0noun)
        (properties &0adj)
        (predication *verb-pred)))
```

This specifies which elements in the pattern fill the slots in the MU definition, e. g. the properties slot is filled by the elements '&0adj'. The structure built using this pattern and rule from the definition for mug : 'a foolish person who is easily deceived' is :

```
((CLASS PERSON)
(PROPERTIES (FOOLISH))
(PREDICATION (OBJECT-OF ((CLASS DECEIVE)))))
```

These patterns work due to the nature of dictionary definitions, which have a specific type of format which the patterns exploit, e. g. nouns are often defined by giving the genus category as the first NP, with following differentia information given in relative clauses or Prepositional Phrases (PP's). The language of dictionary definitions can be viewed as a sublanguage [102], characterised by the consistent use of predictable syntactic and semantic constructions. Typically papers report reasonable success in extracting the genus information, leading to the effective construction of taxonomies, but poorer results in extracting differentia information. Alshawi [8] (in what is more of a feasibility study than a finished piece of work) reports the semantic head (the genus indicator) was identified correctly in 77% of the cases. From these 77% additional information was recovered for 61%, of which 88% was considered correct. Nakamura and Nagao [128] report various results about the success of their thesaurus-building program from Longmans Dictionary of Contemporary English (LDOCE, [135]). These results are somewhat obscure and very varied, for instance 98% of nouns marked as 'state' are 'correctly' identified as abstract, but only 39% of nouns identified as liquid by the extraction program have LDOCE semantic markers of liquid. They mention some particular problems for their program — compound nouns and disjunctive definitions being the main ones.

This brief account of some MRD work shows the clear difference in methodology between the acquisition of semantic information from an MRD, and an NLP system attempting to acquire information about an unknown word it encounters in its processing. In the MRD the word is clearly defined, and pattern matching techniques are applied to derive information from this definition. When an unknown word occurs in a piece of text it does not occur with a definition, and thus the system must attempt to use context and semantic restrictions to make a guess at the meaning. The system described by Alshawi is intended to obtain access to an MRD when a word not in the core lexicon is encountered, and retrieve a definition from there. An interesting point raised by Ludewig in [111] is that such an approach will be problematic for a system operating in real time due to the number of definitions provided for a word. Each of these must be analysed, and then the correct one chosen. Above and beyond such practical problems is the fact that no static lexicon can ever provide a perfect solution, as it will always be possible to meet an unknown that is not in the MRD (as the studies discussed below will show), and then the system must still make recourse to some sort of automatic word learning.

Although some may suppose that MRD's are now the dominant tool in creating large Machine Usable lexicons, a survey of the methods used to create such resources does not necessarily support this position. For instance, of the 15 text understanding systems in the MUC-3 [4] test [50], only 1 claims to have used an MRD to create the large (10-20,000) word lexicon required. Of the remaining systems that had to radically increase their lexicon size all used semi-automated hand encoding.

Whilst the acquisition of information on 'normal' unknown words from text is very different from MRD work, the method of handling PN's described in this thesis shows many similarities. These stem from the fact that PN's are frequently described/defined within the text in which they occur. This text can therefore be thought of as a PN dictionary. As the definitions occur as components of the text under analysis, a full analysis of this text should lead to an understanding of both the events it describes and the PN's within these events. The detailed analysis of PN's in news text has permitted the extraction of lexical-syntactic patterns which match a large number of the PN's encountered. Each pattern has a corresponding semantics which can be utilised to build the lexical entry for a particular PN. These lexical-syntactic patterns and their semantics are described in detail in chapter 6. [5]

The above results show that the current success in deriving semantic information from MRD's is reasonable, but far from perfect. This must be taken into consideration when we consider the adequacy of MRD's in covering free text. Even if it could be shown that MRD's covered 100% of free text, this would still not show that they are the perfect solution to conquering the lexical bottleneck, due to the fact that it has not been shown that anything like all their semantic information can be extracted. It is to attempts to assess the adequacy of MRD's that we now turn.

[4] MUC-3 (the 3rd Message Understanding Conference) is a US DARPA-sponsored initiative aiming to promote and evaluate text understanding of real texts.

[5] In some ways the language of news text could be thought of as a sublanguage, in the same way as can dictionary definitions. This view certainly applies to the constructions used to describe PN's in news text. The large number of PN's that occur in news text, and which do not so regularly occur in other types of text reinforces this characterisation, in that they can be viewed as a special purpose lexicon (cf. Sager [145] 'the lexicon of special languages is their most obvious distinguishing characteristic').

### 2.4.2 The Adequacy of Machine Readable Dictionaries

(Parts of this section were originally published in [44].). Work which assesses the adequacy of coverage of MRD's is important as it allows us to assess the need for automatic word learning. If MRD's can be shown to give 100% coverage, the need for automatic word learning is much reduced (given the corollary mentioned at the end of the previous section). If it can be shown that MRD's give considerably less than 100% coverage, the need for automatic word learning is clear. Unfortunately, as one might expect, the situation does not turn out to be as clear as this, since different studies give different results.

The two most quoted studies are by Walker & Amsler [172], and Sampson [146]. Walker & Amsler compared entries in Webster's Seventh New Collegiate Dictionary to a 3 month sample of stories from the New York Times newswire. They found 64% of the news wire words were not in the dictionary. Their breakdown of these results revealed one fourth to be inflected forms, one fourth were proper nouns, one sixth were hyphenated forms, one twelfth were mis-spellings and one fourth were unresolved, some of which were thought to be new words appearing since the dictionary was published.

That only a third of words from the news wire were in the dictionary is an extremely large discrepancy, and for those involved in MRD work a worrying one. The sources of this divergence can be split into three. Firstly there are the Proper Nouns/Names. This is unsurprising as these are not commonly found in dictionaries. It is their high occurrence in news text, coupled with their poor representation in dictionaries which makes them such an obstacle for NLP of real texts. This is a problem which we seek to overcome in this thesis. Secondly are the inflected forms, hyphenations and misspellings. A good morphological analyser and spell checker should overcome most of these. Lastly are the unresolved categories. Further analysis on these would have determined just how many new words were in this category, but was unfortunately not described. It is likely that some of the cases were caused by trouble with compound nouns, which Amsler discusses in his 1989 paper [10]. Compound words distort the comparison, for unless a compound can be correctly detected as such and removed from the text as a single 'word' each word in the compound will be treated as a single word, which is not the case. This problem is especially troublesome with PN's, where each element of the name will be treated as separate unless the whole name is known. We will review Amsler's work in more depth in chapter 5, as it deals specifically with PN's.

What this work seems to show is that raw text will need considerable pre-processing before lexical look-up from an MRD is undertaken. Although some of this pre-processing will be relatively easy (e. g. morphological analysis), some, such as that concerned with detecting compounds, will remain a problem. In addition, PN's present a problem that is not met by MRD's at present.

In a different study, Sampson [146] examined the convergence between a sample of the LOB (Lancaster Oslo/Bergen) corpus and a machine-usable version of the Oxford Advanced Learner's Dictionary of Current English, called the CUV2. This was created by Roger Mitton [126] as the dictionary for a spell-checker. While omitting definitions and examples, it included some 2,500 PN's (an addition Mitton described in a personal communication as a 'token gesture'), and listed each inflected form as a separate entry. Sampson's study was aimed much more at finding the base word-form convergence, and to this end many obstacles were removed. All sentence initial capitals were converted to lower case, all words with non-alphabetic characters were removed, as were enclitics (such as -n't and -'ll). In addition if a word was not found in the dictionary on the first run various changes were made and the search repeated. For example, all words not flagged as PN's had their initial capital reduced to lower case, the 'ise' type British endings were converted to the American 'ize', and all hyphens were removed. Thus Sampson's study went a long way to meeting the pre-processing shown to be necessary in the Walker and Amsler comparison.

The results were very different from Walker and Amsler's — only 3.24% of the LOB corpus sample were not in Mitton's CUV2. This provides a much more optimistic finding for those engaged in MRD research by showing that a conventional dictionary does give a very good base word coverage of words from free text. The difference in results is so striking that it is interesting to speculate on its causes. Obviously, much of the difference can be attributed to the pre-processing undertaken in Sampson's study. Another reason that I would suggest (which is supported by Sampson in [147]) is the subject matter of the different test corpora. It would seem reasonable that news text will contain more recent (and therefore unknown) words and more PN's than the more varied (and older) texts in the LOB corpus (which although it does contain news text also contains all sorts of other material — novels, scientific texts etc.) In addition, LDOCE contains few PN's, whereas Mitton's CUV2 contains 2,500 high-frequency PN's. It would seem, therefore that a lot of the divergence is due to the higher frequency of PN's in Walker & Amsler's study, and the greater number of PN's in the dictionary in Sampson's study. Despite the optimism of Sampson's results, it remains a fact that words from free text do not occur in the format he utilised. Many of the problems encountered by Walker and Amsler with compound nouns and PN's cannot be overcome by standard heuristics such as removing hyphens or changing British English spelling to American English.

Seitz et al [151] present several informative statistics from their work on constructing a large vocabulary for a spoken word recognizer. They started with a base vocabulary composed of Webster's Seventh New Collegiate Dictionary (W7), the 12,753 words from Francis and Kucera's 'Brown Corpus' which do not occur as top entries in W7, 376 common names from the appendix to W7 and 123 words from a 5000 word sample of newspaper stories. They then processed about four months worth of text from the Toronto Globe and Mail, some 10 million words, to see how many words from this sample were not in their base vocabulary. Twenty thousand 'new' words were found (i. e. words which were not in the base vocabulary) of which half had an initial capital (suggesting PN's and acronyms). This once again shows that conventional resources give very poor coverage of PN's.

Beyond the simple issue of lexical coverage there is the additional, and more complex, issue of the depth of lexical information present in a conventional MRD. The use of 'novel' language is characterised by more than just the use of unknown words. It can also consist of metaphor, idioms, unknown senses of existing words, compounds words and other such phenomena. In [21] Boguraev and Levin argue that the structure of existing computational lexicons is far too influenced by the structure of the MRD's that are used in their construction. This prevents computational lexicons from being able to deal gracefully with the productivity of language. Conventional dictionaries do not carry all the knowledge that speakers know about the words they use, and which enable them to deal with new words and new meanings. Slator [158] puts this point succinctly, in discussing the viability of utilising a 'super dictionary':

> '... it is difficult to imagine the result being other than a hopelessly static structure in a dynamic linguistic world.'

Boguraev and Levin outline the need for, and structure of, a 'Lexical Knowledge Base', characterised by its ability to use inferences based on linguistic generalisations to cope with the open-endedness of language.

It is hard to draw a definite conclusion from these studies. We can make the following judgements:

1. Work using news text seems to find poorer lexical coverage than that using a more varied test sample.
2. The base word form convergence between raw text and standard MRD's is quite good (if we ignore PN's). However, to capitalise on this a system must be equipped with a powerful morphological analyser, and rules governing spelling variations.
3. However good one's initial dictionary and morphological analyser, no system can afford to ignore the problem of unknown words. This seems especially true in news text.
4. PN's present the greatest problem for lexical coverage.

In chapter 5 we shall discuss more work on PN's, and consider these results further.

The fact that dictionaries are necessarily restricted in both their lexical coverage, and the amount of semantic information they can present, has not only led to work on automatic word learning, but also to work on lexical pre-processing from text corpora. These can offer a far greater variety of examples of word usage than a dictionary, albeit in a far less restricted form. In the next two sections we review some of the work in these fields.

## 2.5 Corpus-based Approaches to Lexical Acquisition

In the wider field of computational linguistics there is an ongoing debate on the utility of knowledge-intensive versus knowledge-independent methods. There has been a considerable resurgence of work on the latter over the past few years, particularly based around statistical approaches. Such methods work by finding common patterns in huge bodies of language text — they require little (or no) linguistic or world knowledge. They have been applied very successfully in part of speech tagging [63, 49], and with reasonable success in Machine Translation [24]. They can also be combined with traditional linguistic approaches to produce probabilistic grammars [37]. A good overview of the use of statistics in computational linguistics can be found in [39]. Knowledge-intensive methods are based on a linguistic model, and seek to accomplish their task of language processing through parsing, and semantic and pragmatic interpretation.

This debate has been mirrored in the smaller field of lexical acquisition. Approaches attempting to induce grammatical and semantic information from large corpora, using little or no linguistic knowledge have appeared in the last few years. They seek to provide a low-cost approach, but usually at the price of a lesser return than knowledge-intensive approaches. In a 1989 paper [179], Zernik proposed a knowledge-independent method on the grounds that, although he considered using context as the best method for word learning, it was unrealistic to expect an NLP system to be able to derive a suitable context across domains. Zernik's method acquired a rudimentary semantics for verbs using a large corpus of syntactically analysed sentences. The method required:

1. a set of verb categories (such as dative and beneficiary) each having associated semantic information and a list of all the different configurations of argument structure allowed, e. g. dative verbs permit argument structures like (NP1 verb NP2 NP3) and (NP1 verb NP3 to NP2).

2. a corpus of syntactically analysed sentences (15000 in all).

An input sentence is first analysed to obtain its syntactic argument structure. This is then compared to the argument structures of the verb categories to select potential categories for the verb whose meaning is to be acquired. The variant argument structures of the matching categories are obtained and the corpus is checked to see if an entry exists for the input verb with that variant argument structure. If entries exist in the corpus for all the variants of a given category then it is selected as the category of the input verb. The semantic information associated with that category gives an approximate semantics for the unknown verb. The semantic information acquired is necessarily very broad, as the verb categories share only very broad semantic features (such as verbs of sensation or communication).

This approach, while requiring much less analysis than context-oriented approaches, has several problems. Firstly its success depends on the coverage of the corpus, there is no guarantee that a corpus of just 15,000 sentences will have all the variants. In addition the method relies on a correctly parsed corpus, and to parse 15,000 sentences totally correctly is a task that would stretch the best existing parsers. Second, there are problems concerning the verb categories themselves, for example some variants do not apply to all verbs in the category.

Moreover, the rejection of use of context is unreasonable. Although to expect a present day NLP or text understanding (TU) system to have a broad knowledge across domains is unrealistic, this is not the way the field is moving. The dominant paradigm in TU is to have a core system that can be ported to new domains by the addition of much domain specific knowledge. The importance of such knowledge for effective language understanding is widely recognised and was borne out at the recent MUC-3 conference [50], where one of the main reasons given by teams for poor performance was lack of domain knowledge. In addition, the majority of the teams incorporated 'lexico-semantic' rules into their systems which carried strong expectations as to the type of case-frame slot fillers expected. It is just such expectations that can be exploited in the derivation of meanings for unknowns (as shown by the work of Granger and Carbonell). Moreover, the expectations provided by such rules would be very domain-dependent and thus provide an ideal context for the learning of unknowns. Therefore, we conclude the claim that knowledge dependent systems will not work as the context they require cannot be systematically provided is not supported.

In [141], Rayner et al present a simple method for acquiring word category information from grammatical constraints. The system has a very simple grammar and is presented with a selected sequence of sentences. The allocation of words to part of speech categories is constrained by the fact that the sentence must be parsable (learning is from positive examples only), and that each word has only one part of speech (although this constraint can be slightly relaxed). The approach can be best illustrated by looking at a simple example. Take the sentence 'The man died' and a single grammar rule 'S → Det Noun Verb'. The only allocation that would allow the sentence to be parsed would be Det(the) Noun(man) Verb(died). Thus the system has acquired part of speech information for three words. Given a more complex grammar, the assignment of a single part of speech to each word in the input will not be possible. If for instance we also had the grammar rule: 'S → Adj Noun Verb', as well as the above assignment of parts of speech we would also have to assign 'the' as an adjective. The order of sentence presentation is critical, in that an order where too many unknowns are introduced at once leads to a huge increase in the number of possible lexicons. Although the approach is appealing in its simplicity, and initial work with small grammars is promising, the applicability of this method to practical lexical acquisition remains to be shown.

Velardi et al [169, 168] present a system that builds a lexicon of Surface Semantic Patterns (SSP's) from a large corpus of free text. SSP's represent the meaning of a word through its genus, and restrictions on its cases, thus 'conference' is defined as a subtype of MEETING_ORGANISATION, with its participants restricted to be HUMAN, its theme to be a MENTAL_BUILD, its characteristic to be a EVENT_QUALITY. To automatically acquire the SSP's for a target word, all sentences containing that word are selected from the corpus. Then each of these sentences is parsed (in a shallow fashion) to derive all the phrasal patterns centred around the target word.

The next step is to interpret each phrasal pattern to derive the conceptual relation between the target word and the other words in the pattern. Finally all these semantic interpretations are generalised to produce SSP's. This is done by extracting common supertypes for the case roles in the semantic interpretations. For instance, from a variety of sentences containing the word 'agreement', an SSP is extracted with restricts the participants of an agreement to being human entities.

These SSP's are manually approved before lexical entry. This step is necessary because the limited number of example sentences containing the target word limits the generalisation process. Many of the semantic interpretations may not have common supertypes, this leads to the creation of specific SSP's which simply use the case roles of the semantic interpretations. The method appears to offer a realistic approach to lexical pre-processing from corpus. It is interesting in that it shows the combination of corpus use with a fairly rich linguistic analysis.

In [79], Hearst presents an approach much more akin to the pattern matching approach used in MRD work. Her program scans corpora searching for particular lexical-syntactic patterns that are indicative of the hyponym [6] relationship. The corpus used in development was Grolier's American Academic Encyclopedia. This sort of text can be seen as being part way between a dictionary and free text, in that it can be expected to contain definitions, yet not in so constrained a way as a dictionary. In addition the definitions must first be located in the main body of text. Given the nature of the text, the patterns must be quite restrictive if they are not to generate large numbers of false positives. An example is :

such NP as {NP,}* {(or—and)} NP

this finds text like '... works by **such authors as Herrick, Goldsmith, and Shakespeare.**' The actions associated with this pattern are simply :

hyponym("author","Herrick").
hyponym("author","Goldsmith").
hyponym("author","Shakespeare").

This work appears very promising for recovering coarse-grained information at a low cost. The program has also been run on 20 million words of the New York Times, this located 3178 candidate sentences, of which 46 precisely matched the above pattern, leading to the creation of hyponym assertions. The main drawback, which is illustrated by these

[6] Hyponym is a linguistic term denoting subcategory, e. g. 'football' is a hyponym of 'game'. This terminology can be confusing, as hypernym refers to a supercategory, e. g. 'game' is a hypernym of 'football'. Basically if A is a hyponym of B, then A and B can be linked by an 'isa' link, of the form 'A isa B'.

results, is the huge amount of text that must be scanned to obtain a reasonable number of matches.

Other important work in this field, which space does not permit us to describe has been carried out by Brent [23] on the automatic acquisition of verb subcategorisation frames, and Hindle [83] on determining the similarity of nouns on the basis of occurrence in similar linguistic contexts. The TU team at Bolt Beranek and Newman (BBN) is also investigating probabilistic methods based on large corpora. A 1990 paper [15] lends yet more support to the need for systems to handle novel words:

> ‘ However, even assuming a very large lexicon already exists, it can never be complete. Systems aiming for coverage of unrestricted language in broad domains must continually deal with new words and novel word senses.’

The use of corpora in computational linguistics and lexicography is one of the trends of the late 1980's and the 1990's. It is fast becoming a very large area, and is outside the scope of discussion here. A useful collection of papers in this field is contained in [167].

Next we move on to consider the most recent attempts at word learning. These have been heavily motivated by the need for robust text processing.

## 2.6 Recent Attempts at Word Learning

Much of the work below takes a similar approach to that of Granger and Carbonell (discussed above). The major difference is that these attempts are not heavily reliant on particular knowledge structures. One of the problems in gathering information on word learning for robust text processing is that much of the work is done in companies that are working on marketable TU systems. As such, published information describing the work is hard to come by.

The main centre for work on automatic word learning in the late 1980's was General Electric (GE). This work has developed under the umbrella of a text processing system called SCISOR which reads news stories in the domain of corporate finance with a view to summarising and updating a database of information on company takeovers [140, 138, 96]. In a 1988 paper [183], Zernik and Jacobs present work on the learning of meanings for single unknown words. The programs TRUMP and RINA aimed to combine information from four different levels:

- morphology
- syntax
- semantics
- context

The use of syntax and semantics was very similar to Granger's work, utilising constraints on the part of speech and semantic category of the unknown. The original aspects of this approach were the focus on morphology and the lack of dependence on predictions derived from an in-depth contextual analysis. The morphology of an unknown word can provide a rich source of information as to its word class and even its meaning. The TRUMP system has a base lexicon of 10,000 roots, so morphological analysis will often reveal a known root. A simple example is affix stripping, e. g. the meaning of ‘unchanged’ can be derived from its root and prefix if ‘change’ is already in the lexicon. A word like merger can be revealed to be either a comparative adjective, a simple root (like ledger) or a noun like 'shopper' describing the actor of some hypothetical action 'merge/merg'.

The system also makes use of surrounding words that may be associated with the unknown. Looking at the unknown word 'merger' again, in the example described in [183] this occurred as 'merger offer'. Therefore it may describe a type of offer or something being offered.

The use of context is different from previous work, not being heavily dependent on knowledge structures. Much emphasis is placed on key words like 'another' which provide clues to the previous occurrence of the unknown, presumably under a different name. In fact, the example given in [183] seems overly reliant on this and it is not clear how such detailed definitions could be arrived at without this crucial clue. The context used is that created by the previous input, so in the corporate finance realm this would be a list of previously mentioned companies, takeovers etc. In the merger example, the full sentence was 'Warnaco received another merger offer, valued at $36 a share'. This indicates that the 'merger offer' was the object of an abstract transfer event (receive) in which Warnaco was the recipient. The context is built from previous sentences, and contains a previous offer in which Warnaco was the acquirer. The system refers the new phrase 'merger offer' to this previous phrase, based on the head noun 'offer', the word 'another' and the existence of the previous offer in the context. As the previous offer was an acquisition offer, merger can be classified as subtype of company_acquisition.

The process of deriving a meaning is illuminatingly viewed as a two stage process of hypothesis formation and refinement. The hypothesis is the meaning for the unknown and it is formed upon the first encounter. Subsequent encounters are considered to refine the meaning until a workable definition is arrived at. This is a more realistic approach than assuming one encounter is sufficient to give a workable definition.

Work on text processing at GE has progressed under the MUC-2 and MUC-3 initiatives, and will be discussed further in subsequent chapters.

Dejong & Mooney [52] and Mooney [127] present a system that actually learns words for which no concept existed previously. Their work, like that of the Yale systems, is partly based on the modelling of human learning. However it also intends to deal with the processing of unknown words. It combines similar techniques to the FOUL-UP system with an explanation-based learning mechanism that acquires new schemata (similar to scripts) which explain new concepts. The newly-acquired schema can then be used to fill in more detail on the unknown words, if these actually feature in the schema.

The LILOG project [81] of IBM Germany is a large project started in the mid-eighties aimed at creating a robust text understander for a reasonable segment of the German language. The system is equipped with several mechanisms for coping with unknown words, something which the authors stress is of prime importance in designing a system aimed at real text [111, 56]. Emde, in [56], describes the actually implemented methods for coping with unknowns. If a word occurs which is not in the core lexicon, the following methods are utilised (applied in the order shown):

- Search for the word in the temporary lexicon, i. e. the lexicon which contains newly-learned words not yet entered into the core lexicon.

- If this fails search in the synonym lexicon. This specifies synonyms for 346 word roots (nouns, verbs, adjectives, and adverbs).

- If this fails utilise morphological and compound word formation rules to attempt to derive a known root.

- Finally utilise interaction with the system user to obtain a syntactic and semantic category for the unknown

Automatically acquired words are stored in a temporary lexicon before being manually transferred into the core lexicon. This is something we have considered in the FUNES system, to prevent the automatic entry of incorrectly-acquired material into the core lexicon. The system pays particular attention to compounds ending in 'strasse' (street), 'platz' (place) and 'allee' (avenue), as these indicate street names. This is a similar approach to the FUNES use of place 'key words' (such as street, square, and avenue), except that in German street names form a single word. [111] is a more theoretical account of the possible approaches to word acquisition, which considers the use of 'contextual specification' to acquire an unknown word. This is equivalent to the use of context, seen in the systems of Granger and Carbonell. However, it has not yet been implemented in the LILOG project. We shall return to [111] in chapter 4 as it contains some interesting similarities to unknown word acquisition as implemented in FUNES.

In [113, 77] Lytinen and Hastings discuss ongoing work with the NLP system LINK. This is a unification-based processor that is able to infer syntactic and semantic information on unknown words from syntactic and semantic restrictions or expectations. The former paper describes the use of syntactic rules to determine the part of speech of an unknown word, and selectional restrictions to infer a global semantic category. Thus from the example 'John ate a sandwich', if sandwich were unknown it could be hypothesized to be a noun due to its position in the sentence and the grammar rule 'NP → Det Noun'. The selectional restrictions of 'eat' would assign it a category of 'food'. This approach is identical to that of Granger, albeit implemented in a much neater unification-based framework. This is able to use exactly the same mechanism — unification — to infer both semantic and syntactic information.

Like Granger, the authors discuss how such an approach is best suited to the derivation of meaning for unknown nouns. A context or expectation-driven approach to word learning is obviously dependent on the strength of expectations for a particular word. A verb will typically carry quite strong expectations on the nouns that may accompany it. Embodied as selectional restrictions these can provide hypotheses for the meaning of an unknown subject or object. However, unknown verbs are harder to define within this approach as they are the central element of the sentence, and thus the source of any expectations the sentence provides. For instance, the number of things a human being may actually do is vast, while the number of things that can be the subject of thinking verbs (for example) can be restricted to just one — a human being. Verb meanings must be derived from contextually-based expectations, e. g. in a breakfast script the occurrence of the verb 'scoff' could be restricted to a verb of cooking or eating. The most difficult class of word to derive a meaning for based on expectations is the adjective, as these are not expected – they lack predefined slots in a sentence case-frame.

The approach taken in [113, 77] for acquiring verb meanings is one of concept refinement. The first assumption for any unknown verb is that it represents an Action or a State, the two top categories in the verb hierarchy. The process of concept refinement moves down the verb hierarchy to more specific verb concepts, guided by the types of slot filler in the unknown verb's case frame. For instance, if the subject slot were filled by a noun known to be human, the verb category would be refined to the more specific

'Intentional-act'. If the object slot were filled by a noun known to be food, the verb meaning could be refined still further to 'Ingest'.

In [77], this approach is applied to a specific domain, that of an automobile assembly line. This benefits from having a narrow domain that can be rigidly specified in a concept hierarchy. Such an approach was also taken by Keirsey [101] in the domain of naval ship to shore messages. Both are able to use the narrowness of the domain, and the idea of a concept hierarchy, to infer relatively specific meanings for unknowns. In [77] a facility is also introduced for refining a hypothesis over multiple encounters with an unknown.

The main reservation about such work is the necessity of it. Although an ability to cope with unknown words is a necessary requirement for a robust parser, the need for coping with unknowns in a limited domain is much reduced. In a domain as limited as those described the vocabulary will not be large, especially that used to describe the objects involved in automobile assembly. Although an ability to cope with unknowns is never wasted, in that however constrained the domain there will always be some cases of unknown words, it would perhaps be more appropriate to apply the method to a wider area, where the likelihood of meeting unknowns is that much higher.

## 2.7 Summary and Conclusion

This chapter has given an outline of work in lexical acquisition. The following areas have been discussed:

- Machine Learning of Natural Language (MLNL)
- MRD work
- Automatic Acquisition from Corpora
- Automatic Word Learning

We have seen that lexical acquisition work has been motivated by both theoretical and practical considerations. The theoretical side has been involved with MLNL, systems intended to model infant and adult language acquisition. On the practical side it has been shown that the need for large computational lexicons has only arisen relatively recently. This is due to theoretical changes in linguistic theory attaching more importance to the lexicon, and to the practical need of modern systems for broad coverage lexicons. MRD's have made a great contribution to overcoming the lexical bottleneck. In addition, the ever-increasing availability of MRD's and large text corpora mean that the creation of large MU lexicons is becoming a more feasible task. It is now common for large NLP systems to have lexicons of 10,000 words or more, whereas only five years ago such lexicons were a great rarity. In 1987 Whitelock [176] reported on an informal poll to establish the average size of system lexicons at a workshop on linguistic theory and computer applications. Excluding one MT system that had a lexicon of 5000 words, the average size was 25 words.

However, it has become apparent that dictionaries are limited both in their lexical coverage, and in their depth of semantic information. This inadequacy has driven the rise of work on lexical acquisition from corpora and automatic word learning.

The present thesis attempts to address a major problem (if not the major problem) in lexical inadequacy — the analysis and acquisition of PN's. From the point of view of their 'unknownness', PN's can be viewed as a subset of the class of general unknown words.

Therefore, rather than approach this topic in isolation we have approached it as an aspect of the field of automatic word learning.

It is the handling of 'general' unknown words that we consider in this first part of the thesis. The methods we have produced for this can be applied to PN's as well as other types of unknown. Our approach builds on the work of Granger, Carbonell, and Zernik and Jacobs by applying constraints on a word's syntactic and semantic class at all levels of linguistic analysis. In addition the meaning acquired for a word can be refined upon subsequent occurrences.

To facilitate an understanding of how unknowns are handled, in the next chapter we present a brief description of the FUNES system. This is intended to enable the reader to place subsequent descriptions of unknown and PN handling in their appropriate context, i. e. as components or aspects of the general text understanding task. Having outlined the overall architecture of the FUNES system, in chapter 4 we describe the suite of methods used for processing unknown words, with particular focus on unknown common nouns.

# Chapter 3

# A Brief Overview of the FUNES System

## 3.1 Introduction

In this chapter we give a short over-view of the FUNES system. Description is kept to a minimum as the architecture and functioning of the FUNES system are not a central element of this thesis — they are simply a means to an end. Subsequent chapters will fill in detail where it is required for an understanding of the handling of unknowns and PN's. In appendices A-F each section of the system is described in more detail. Any reader wishing to know more about a particular aspect can therefore refer to the appropriate appendix.

FUNES can be viewed as comprising four separate modules that interlink. These are :

- Lexicon and Knowledge Base
- Pre-Processor
- Syntactic Parser
- Semantic Analyser

The input to FUNES is a stream of raw text, composed of ASCII characters. This is first tokenised, that is, split into a list of words and punctuation characters. Each word is then looked up in the lexicon, with words not found being classified as unknown. The lexical look-up returns one or more parts of speech for each known word. Lexically ambiguous words are resolved, and the part of speech of unknown words derived, before the parser is called. This receives a list of terms, where each term is a word indexed by its part of speech. The parser's main task is to locate the major syntactic constructions dealt with by FUNES — Noun Phrases (NP's), Prepositional Phrases (PP's) and Verb Phrases (VP's). When a NP-VP group is located, it is passed to the semantic analyser which derives a case-based semantic representation. Control then returns to the parser to continue with syntactic analysis, if this has not been completed. Co-routining of the syntactic and semantic modules allows partial analyses to be returned if the parser should fail on a later section of a sentence. The top-level architecture of FUNES is shown in figure 3.1. Below we describe each of the modules in turn.

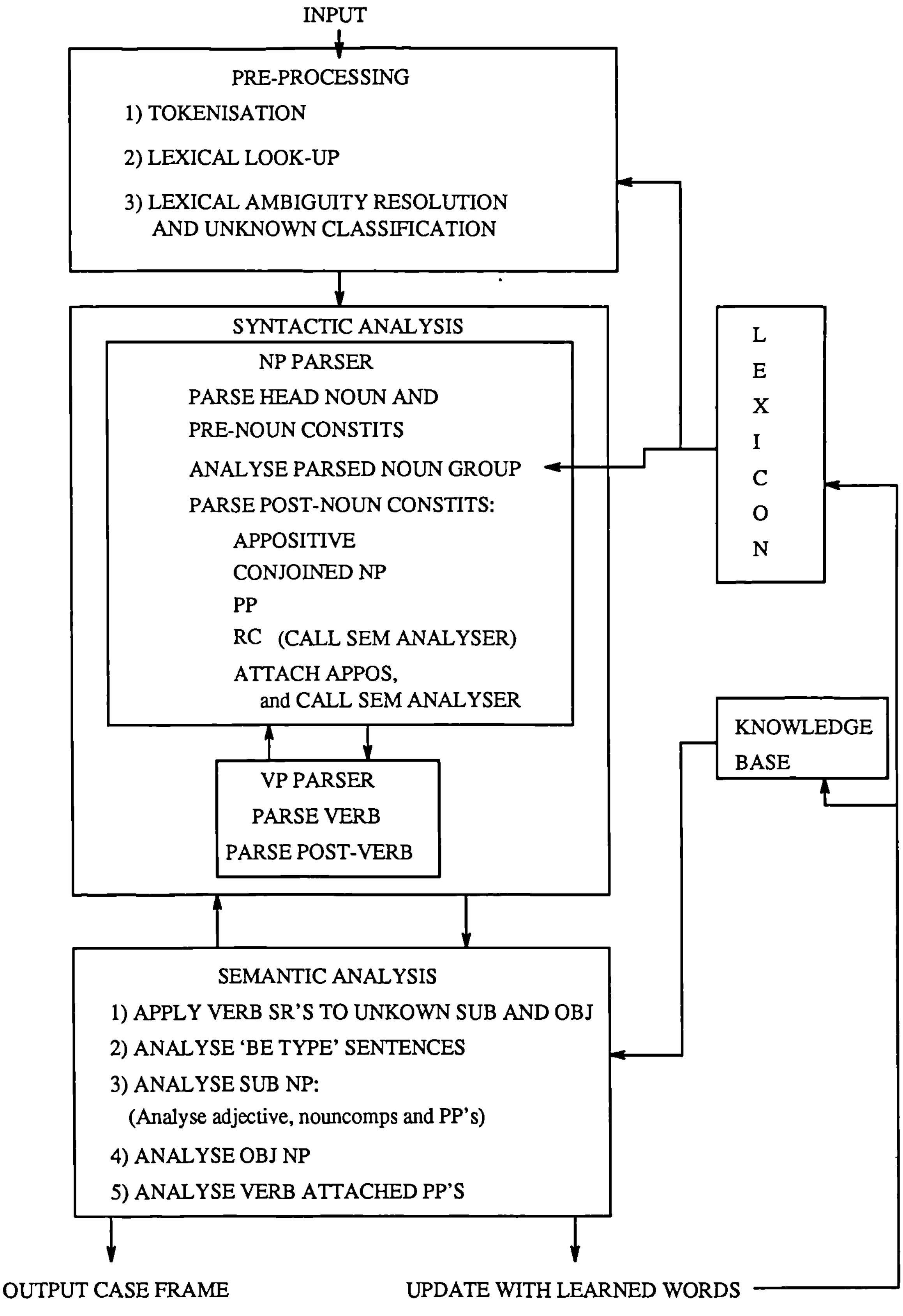

INPUT
PRE-PROCESSING
1) TOKENISATION
2) LEXICAL LOOK-UP
3) LEXICAL AMBIGUITY RESOLUTION AND UNKNOWN CLASSIFICATION
SYNTACTIC ANALYSIS
NP PARSER
PARSE HEAD NOUN AND
PRE-NOUN CONSTITS
ANALYSE PARSED NOUN GROUP
PARSE POST-NOUN CONSTITS:
APPOSITIVE
CONJOINED NP
PP
RC (CALL SEM ANALYSER)
ATTACH APPOS, and CALL SEM ANALYSER
VP PARSER
PARSE VERB
PARSE POST-VERB
LEXICON
KNOWLEDGE BASE
SEMANTIC ANALYSIS
1) APPLY VERB SR'S TO UNKOWN SUB AND OBJ
2) ANALYSE 'BE TYPE' SENTENCES
3) ANALYSE SUB NP:
(Analyse adjective, nouncomps and PP's)
4) ANALYSE OBJ NP
5) ANALYSE VERB ATTACHED PP'S
OUTPUT CASE FRAME
UPDATE WITH LEARNED WORDS

Figure 3.1: Architecture of the FUNES system

## 3.2 The Lexicon and Knowledge Base

FUNES has a lexicon of about 2000 common word roots. In addition inflected forms for all 560 verbs are listed. Morphological processing, dealing with plural noun forms and comparative and superlative adjective forms, extends the vocabulary considerably. The lexicon also contains 37 collocations such as 'nothing short of' and 'more and more'; 47 phrasal verbs such as 'call off' and 'work out'; and 44 compound nouns such as 'venture capital' and 'prime minister'. The Knowledge base for the existing form of FUNES is compact, simply consisting of a semantic hierarchy of the major semantic categories covered in the FUNES domain of news stories.

Lexical entries differ for each part of speech. The entry for nouns (with which we are most concerned in this thesis) consists of the noun itself, a semantic category (such as human or location) and gender, e. g. 'noun(attack,[event],n)'. Semantic categories are held in a semantic network in the Knowledge Base. The entry for verbs consists of the verb, its transitivity, its semantic category, and subject and object selectional restrictions, held as lists of semantic categories.

Inflected verb forms are held in the lexicon as 4 or 5-tuples, depending on the existence of an 'en' form. These are automatically pre-processed upon consultation of the lexicon into a format indicating the tense and number of that particular form, utilising a method described in [171].

## 3.3 The Pre-Processor

This module accepts as input a stream of raw text, composed of ASCII characters. It outputs a list of terms consisting of each word, its part of speech and associated syntactic and semantic information. Pre-processing is most easily viewed as consisting of three stages :

- Tokenisation
- Lexical Look-up
- Lexical Ambiguity Resolution and Unknown Part of Speech Classification

### 3.3.1 Tokenisation

The process of tokenising an input stream for subsequent linguistic analysis is an established one. In FUNES the input is read character by character from the input file. Words are considered to be terminated by carriage returns, punctuation characters or spaces. Each word is returned as a triple, consisting of the word itself in lower case, an indication of its case, and a third variable which indicates whether the word was hyphenated, or whether it was preceded or followed by a bracket. Case can be any of three types : word (indicating a normal word), init_cap (indicating a word with an initial capital), or abb (indicating a word with two initial capitals, commonly an acronym).

### 3.3.2 Lexical Look-up

The input for the lexical look-up stage is the output of the tokeniser described above. Here every word is looked-up in the lexicon to obtain syntactic and semantic information, which is subsequently held with that word. If a word is not found in the lexicon, various morphological changes are carried out and the result is once more searched for in the lexicon. If it is still not found the word is returned as unknown. The output of this stage is a list of lists, where each sublist contains one or more terms, indexed by the part(s) of speech of a word, and containing various items of information. This information is dependent on the word's part of speech.

The lexical look-up stage also handles hyphenated words (described in the next chapter), compound words (such as 'home office', 'nothing short of'), bracketed words, and known PN compounds (described in chapter 9).

### 3.3.3 Lexical Ambiguity Resolution and Unknown Part of Speech Classification

Lexical ambiguity is one of the most awkward problems facing NLP systems. In English, a large percentage of words are lexically ambiguous, in that they can be included in more than one syntactic category. The majority of these cases can be resolved by examining the parts of speech of neighbouring words. In addition, in some cases, the morphology of a word can help in the decision.

Any term returned from lexical look-up that is unknown or lexically ambiguous is analysed in this stage. PN's are detected by their initial capital. Other unknowns, and known ambiguous words, have their correct part of speech derived by procedures which inspect the syntactic category of neighbouring words. In cases where a definite decision cannot be reached the morphology of the word is used to make a decision. This process is described in the next chapter (as the same procedures are used for deriving an unknown's part of speech and for disambiguating ambiguous noun/verbs).

At the end of the Lexical Ambiguity Resolution stage each ambiguous word will have been resolved, and the list of lists passed in from the lexical look-up stage is returned as a simple list. The elements of this list are functor-argument structures, wherein the functor is a part of speech, and the arguments depend on the particular class of functor. All the ambiguity resolution heuristics utilised in FUNES are contained in appendix D.

The list returned from the LAR stage is passed onto the parser, to which we now turn our attention.

## 3.4 The Syntactic Parser

The FUNES parser operates with an overall top-down control algorithm. This hypothesis-driven approach is balanced by a data-driven look-ahead facility which makes finer grained decisions on rule application. The parser works in a logic grammar type formalism, the list of unparsed words being handed from predicate to predicate. This formalism is augmented with Augmented Transition Network (ATN) like registers for holding NP's, VP's etc. We can see that the FUNES parser combines elements of three of the dominant parsing 'schools':

- Logic/Prolog Parsing (see [132, 65]).
- ATN Parsing (see [178]).
- Marcus/deterministic parsing (see [114]).

FUNES incorporates no elements from chart parsing (see [98, 65]) — it delivers a single syntactic analysis and does minimal backtracking. When it does backtrack this does not involve complete destruction of built constituents, but rather a re-arrangement.

The parser does not attempt to derive a complete syntactic analysis of each sentence. To attempt to do so with real text is an extremely difficult, if not impossible process. This observation is supported by many of the system designers taking part in the MUC-3 challenge [50]. Systems which relied on complete syntactic analysis were forced to incorporate a variety of recovery strategies for dealing with failed parses (some of which are described in [88, 71]). Many of the more successful systems utilised a partial parsing approach. McDonald [120, 121] has pointed out that the large number of unknown words that will occur in unedited real text, coupled with the frequent grammatical violations, and the fact that many stretches of a text will simply be beyond a parser's capability, mean that anything beyond such a partial parsing approach will not succeed. This methodology has also been incorporated in Hindle's Fidditch parser [82] and the BBN TU system [174].

The FUNES parser basically looks for NP-VP constituents. Upon encountering relative clauses, appositives and embedded sentences it analyses each into NP, PP and VP and passes straight to the semantic analyser. Upon completion this passes back to the parser to continue the parse of the main sentence. Such an approach enables semantic information to be recovered despite subsequent parse failure. The focus of syntactic analysis is on NP's, as it is these that contain any PN descriptions, and such descriptions have been the central element of this thesis. If FUNES should encounter a construction it cannot handle it simply ends its analysis at that point and proceeds to the next sentence.

The FUNES grammar (shown in appendix E) is implemented as a logic grammar. The main parsing predicate is handed the list of words and their parts of speech which is output by the pre-processing stage. This is handed to the NP parser which processes the initial NP constituents, and hands the remaining list to the VP parser which processes the VP constituents. Normally it would be insisted that a successful parse consumed all the words in the list, but due to the complexities of real text, and the fact that FUNES grammar only covers a subset of the news text encountered this is not enforced. Within the NP (and VP) parser the parse is controlled in the same way, the predicates which process each sub-constituent are handed the input list of words, and return that list minus the parsed constituent which is returned in a separate variable.

Each of the predicates except that for parsing the Noun Group and main verb will always succeed. However if the constituent that a predicate is meant to parse is absent it will return no value, and the slot for that constituent in the appropriate register will remain empty. The main syntactic constituents (namely NP, PP and Verb) are held in registers once parsed, rather than being returned as arguments in the appropriate predicates. The main reason for this is the large number of arguments that quickly proliferate in such a grammar when it is being used to process complex sentences. In FUNES it may be necessary to examine any previous constituent at various times, and it is far easier to hold these in globally available registers than passing them all along to each predicate.

One of the most important concepts in the FUNES parser is that of syntactic levels. Parsed constituents are stored in registers, and each register is indexed to a specific level. A level can be seen as an NP-VP unit. The main sentence is held to be at Level Zero. Whenever a sentential constituent — a relative clause, an appositive or an embedded sentence — occurs, the Level is increased by one, and a recursive call to the NP or VP (or both) parsers is made. Thus a whole new set of registers is made available. At the end of parsing such a constituent, the contents of all the registers at that level are handed to the semantic analyser. When this has derived the case-frame, control returns to the syntactic parser which continues parsing at the original level. Movement is handled by copying of registers between levels.

In the next section we look briefly at the NP and VP parsers.

### 3.4.1 Parsing of Noun-Phrases

The process of parsing a NP can best be viewed as a pass through a list of parts of speech to locate the head noun. During the course of this pass various actions are taken, dependent on the words encountered. Each head noun is stored in a noun register, together with any accompanying adjectives and noun complements. When the head noun has been located and the noun register filled, the parse continues by looking for appositives, conjunctions, PP's, and relative clauses.

Pronoun reference is handled in a standard fashion [86, 4] utilising a history list of previous nouns, which are then matched on number and gender.

A NP register has the following form :

Noun : (the head noun)
Case : (the letter case of Noun)
Sem_Cat : (the semantic category of Noun)
Adj : (any adjectives in the NP)
Noun_C : (any noun complements in the NP)
Det : (the type of determiner )

Any of these slots except the first two can be empty. The Adj slot would be empty for example if there were no adjectives in the NP, and the Sem_Cat slot would be empty if the head noun were unknown, and a semantic category had not yet been derived.

The parsing of post-noun constituents makes extensive use of look-ahead. For the most part these constituents are all clearly signalled by marker words such as prepositions or relative pronouns. The utilisation of look-ahead leads to more efficient parsing, preventing the pointless calling of high level predicates to parse constituents that are not present. It can be viewed as a bottom-up process, as it makes the firing of rules sensitive to the presence of the words required for their successful application. This provides a data-driven adjunct to the main top-down or hypothesis-driven control mechanism.

Appositives will be discussed at length in chapter 8, as they are most frequently used to provide descriptions of personal PN's. Once detected the NP parser is called recursively at an incremented Level to carry out the parse. Upon completion the semantic analyser is called to derive the case-frame for the main NP/appositive NP pair. When this exits the main NP parse continues, searching for a conjunction such as 'and'. FUNES concentrates on NP conjunction, and if a conjunction is detected calls the NP parser to parse the conjoined NP. Again this process will be discussed at length in chapter 8, as conjunctions and apposition can interact in a complex manner in the description of PN's.

After the parsing of any conjoined NP's, the main NP parser continues by searching for a preposition. If found this indicates the presence of a PP. The notorious problem in processing PP's is that of attachment — does the PP attach to the preceding noun or to the verb. The literature on PP attachment is extensive. A succinct account of the problems of attachment ambiguities can, however, be found in [85].

In FUNES a few general attachment heuristics are used to decide on attachment. These heuristics utilise semantic information to decide if the PP should attach to the current NP. As many PN's can include PP's, the heuristics make use of knowledge of the structure of certain classes of PN to aid in the decision. This decision is made before the PP is actually parsed. If it is decided that it does not attach to the present NP, then the PP is not parsed, and will instead be attached to the verb. (If parsing a subject NP, then any following PP is always parsed and attached to the subject). Upon completion of PP analysis the main NP parse continues, searching for a relative clause.

The relative clause is an ubiquitous construction in news text. FUNES can handle many different types as discussed in more detail in appendix F. A relative clause is usually flagged by the occurrence of a relative pronoun — 'who', 'which' or 'where'. We also treat as relative clauses constructions where a sentential complement marker (e. g. 'to', or 'for') and a verb phrase follow a noun (as in 'who devised a plan **to infect leading politicians**'). A relative clause can also occur with no marker, in which case it is known as a reduced relative clause. This can either occur with a progressive verb, as in 'Gunmen opened fire, killing at least three', or with a past tense verb, as in 'for a film series based on their files'. FUNES can only process the first type.

Relative clauses are parsed by calls to the NP or VP parsers (or both), which parse the constituent at an incremented Level. When the relative clause has been parsed, the 'missing' constituent (either a subject or object NP or a PP) is located, and the semantic analyser called. When this has derived the appropriate case-frame, the relative clause has effectively been removed from the sentence, and the parse of the main sentence continues.

This completes our short account of the parsing of NP's. Appendix F gives considerable more detail on the process of parsing NP's, and how it it carried out in FUNES. We next consider the VP parser.

### 3.4.2 Parsing of Verb Phrases

As can be seen from the grammar for the VP (see appendix E), the parsing of this constituent is a complex process. This is due to the great variety in the structure of the VP, which stems from the variety of post verb constituents — NP's, PP's, embedded sentences, and any combination thereof. However, apart from the verb and any preceding auxiliaries the main content of the VP is actually made up from NP's.

As with NP's, VP's can most easily be thought of as composed of pre-complements and post-complements. The Pre-verb structure is comprised of the verb auxiliaries — modals, have, be and do. Any of the auxiliaries can be followed by a negative or an adverb. The main verb is parsed in a similar fashion to the auxiliaries, and tense, number and transitivity information returned. The transitivity value may later be overwritten depending on the presence or absence of an object. As with auxiliaries the main verb can be followed by an adverb or negative.

There are four main types of post-verb structure:

1. Embedded Sentences

2. Noun Phrases

3. Adjective Phrases

4. Prepositional Phrases

In addition intransitive verbs can occur with no post-verb constituents, e. g. 'the car exploded'. We examine each of the above components in turn.

Only verbs held in the lexicon as COMP can take embedded sentences. However to cope with unknown verbs we also test for such a structure whenever the main verb is followed by 'that' or 'modal(to)'. Embedded sentences are usually signalled by the presence of a marker such as 'whether', 'to', 'for' or 'that'. An embedded sentence can also be signalled by the presence of a progressive verb with no auxiliary directly following the main verb. The problems of parsing a VP are worsened by the fact that very many COMP verbs can also be used transitively. So we can have 'see' (for example) used with a sentential complement, as in 'we saw the plane explode in mid-air', or with a simple object, as in 'we saw the plane'.

Look-ahead is used extensively here to make processing more efficient by deciding which post-verb structure is present before calling the relevant predicates to parse it. Embedded sentences are parsed, when detected, in the same way as relative clauses, by calls to the VP parser, or to both the NP and VP parser. When the parse is complete, the semantic analyser is called to derive the case frame for the embedded sentence. Before this can be done, however, if the embedded constituent was a simple VP, rather than an NP + VP, the missing subject is located. This will be either the main sentence subject or object, and is located by calling up the appropriate NP register, and copying the contents into the subject NP register at Level+1.

For verbs which are not marked COMP the parse is simpler. If the verb is marked ADJ it may take an Adjective Phrase (AP) complement, e. g. 'they were angry'. The situation is, however, complicated by the fact that all verbs that take an AP complement can also take an NP complement. If present the AP is held in an object NP register, and will be analysed in the same way as the NP object in a copular sentence.

If the verb is neither COMP nor ADJ, it is assumed to be a simple transitive or intransitive case. This is determined by the presence or absence of an object NP. If an object NP was parsed, it is also possible that there may be a second NP, in which case the previous object was in fact the indirect object. Such situations are handled by re-arrangement of registers.

Passive constructions are also handled through register rearrangement. If, after the main verb and its auxiliaries have been parsed, the VOICE variable is set to 'pass', the subject NP register is cleared and its contents copied into an object NP register.

After parsing of the post-verb constituents, the verb is entered into the verb register. This contains slots for:

Trans: the verb's transitivity (trans or intrans)
Sem_cat: the semantic category of the verb (e. g. mbuild or mtrans)
Tense: past, present or prog
Inflected form (needed as the verb itself is held in root form)
Subject selectional restrictions (held as list of sem_cats)
Object selectional restrictions (held as list of sem_cats)

The selectional restrictions are looked up in the short-term lexicon. [1] This can only be done after processing of the VP, which reveals the precise TRANS (comp, transitive, intransitive) of the verb, as the SR's depend on the TRANS setting.

This completes the brief account of the parsing of VP's, and the account of the syntactic parser. Appendix G expands on the description of the VP parser, and also provides details of other syntactic constructions handled in FUNES. We now turn to a description of the semantic analyser.

## 3.5 The Semantic Analyser

Semantic analysis consists of various processes:

- deriving the case (agent, theme etc.) of each NP passed to it.

[1] This is a temporary store into which all the words in the current sentence are copied, together with accompanying lexical information, in order to facilitate any subsequent inspection.

- deriving the case relationship of adjectives and noun complements to the head noun.
- deriving detailed case relationships for PP's.
- Detailed analysis of unknown words and especially PN's. This will not be considered here, but in subsequent chapters.

The analyser produces a Case-Frame (CF) for each 'sentence' handed to it, which represents the meaning of the sentence. The nature of the CF is loosely based on case grammar [58], and follows the semantic analyser outlined by Allen in [4]. The CF is anchored around the verb, which appears in its root form with a tense indicator. The Subject and Object NP's and PP's fill the 'cases' of the verb, such as Agent, Theme, or Loc. In turn each NP has its own case frame in which its accompanying qualifiers (adjectives, noun complements, and PP's) fill slots anchored around the head noun. The variety of cases used is greater than in Fillmore's original theory. Appendix H contains a specification of all the major cases and how they are derived. In subsequent chapters the terms 'case' and 'case label' are used interchangeably, to refer to cases such as 'agent' and 'origin'. The term 'Case-Frame' is used to refer both to a single case and its filler (e. g. [agent(president)]), and a list of such structures appended together to represent a whole NP or sentence (e. g. [agent(president),origin(france),property(former)]).

In the following sections we describe the various different stages of semantic analysis, at a fairly superficial level.

The semantic analyser receives as input the registers created by the parser. These are re-indexed in accordance with the constituent they hold, rather than the Level they were created at. The analysis proceeds in various stages, as shown below:

1) Processing of unknown subject and object nouns.
2) Processing of subject and object NP's
3) Processing of noun attached PP's
4) Processing of verb attached PP's

We describe each of these stages below, except the first, which is dealt with in the following chapter.

- Subject and Object NP Analysis.

  The first step in analysis is to check that the semantic category of the noun is consistent with the selectional restrictions of the verb. When the SR's have been checked, the next step is to analyse any PP's attached to this particular noun. We consider this in the next section. Then the case of the subject or object noun is derived. This is done by examining the transitivity and semantic category of the verb, and the semantic category of the noun. The choice is essentially between AGENT, THEME, or INSTRUMENT. The heuristics used are described in appendix H. Then the adjectives and noun complements (if present) attached to this noun are processed.

- Processing of Adjectives and Noun Complements

  Certain types of adjectives receive special processing. The default case for all other adjectives is simply 'property'. Digits, Measures, Locations and Durations are analysed first to derive the cases Day, Age, Number, Distance, Size, Origin, Superpart, or Duration.

The CF for adjectives is returned as a list of case labels and their arguments (the adjectives themselves), e. g. 'the 10 former hostages' is returned as [number(10),property(former)]. This would be appended to the CF for the head noun 'hostages', to produce, for example, [theme(hostage),number(10),property(former)].

Noun complements are handled in a somewhat similar fashion, only the emphasis is on detection of PN's, in particular corporation names. In FUNES we do not consider the problem of compound common nouns, unless they are held in the lexicon. Basically any noun complements which cannot be analysed as organisation names, or which were not included in the analysis of PN's done within the parser, are simply returned as 'property'.

- The Analysis of Noun Attached PP's.

  Analysis consists of deriving the case of the PP by calling the appropriate PP module, and then analysing the accompanying adjectives and noun complements of that PP as described above. Each preposition has its own module, which derives a case for the PP (such as Superpart, Works_for or Field) representing the relationship between the PP and the NP to which it is attached. This case is derived through consideration of the preposition itself, and the semantic categories of the PP head noun and the head noun of the NP to which it is attached.

  Considerable attention is also paid to detecting corporation PN's (hence corp PN's) that may run over several PP's, and to detecting relationships between PN's that may be given through PP's, such as origin (for a corp/person and a country), works_for (between a person and a corp), and name (between a location and its name). These are described in subsequent chapters. The PP modules also derive a semantic category for any unknown word, as described in the next chapter.

- The Analysis of Verb Attached PP's

  The analysis of these constituents is virtually identical to the above except that we need not consider the semantic category of the preceding noun. Instead the verb's semantic category is considered. It is important to note that verb attached PP's can have PP's attached to them, just as they are attached to their verb. Thus in the sentence '... that he had been too involved in the Iran-Contra affair during an earlier stint as deputy director of the CIA', the PP 'during an earlier stint' is attached to the verb, but 'as deputy director' is attached to 'stint' and then 'of the CIA' is attached to 'deputy director'.

  Thus, after each verb attached PP has itself been analysed we must check the NP-attached PP register to see if any of the PP's therein are attached to the verb PP we have just analysed.

  Verb attached PP's fill verb cases, mainly temporal and locational ones, such as During, Before, At_loc, or From_loc.

The CF's for each NP-VP unit are stored as lists and output after each sentence has been fully analysed. CF's for embedded sentences are slotted into the CF for the main sentence in a COMP slot. CF's for relative clauses and appositives are simply output alongside that for the main sentence.

## 3.6 Summary

In this chapter we have given a brief overview of the architecture and functioning of the FUNES system. The sole purpose of the chapter is to facilitate the understanding of the subsequent chapters which describe the heart of the thesis. These chapters should serve themselves to clarify and add detail to the above description.

It is to the question of unknowns that we now turn. In the next chapter we describe the handling of 'normal' unknown words, i. e. unknown common nouns, verbs, adverbs and adjectives. This will provide the backbone to the following chapters which will show how PN's can be handled by a combination of 'normal' word learning techniques and PN specific techniques.

# Chapter 4

# The Processing of Unknown Words and Acquisition of Common Nouns

## 4.1 Introduction

In this chapter we describe methods for processing unknown words, and for acquiring a meaning for unknown common nouns. We describe each level of processing — pre-processing, syntactic and semantic — and show how different information at each of these levels can contribute to a meaning for an unknown word. Each of these stages can be thought of as adding restrictions to the possible meanings an unknown can have. At the start of processing an unknown could be any open class word with any meaning. By the end of processing all incorrect or irrelevant information will have been filtered away, leaving only the correct information. We also show how pragmatic information can be used for the same purpose. An earlier implementation of FUNES utilised context, based upon goal/plan analysis, to 'fill out' the broader semantic information derived in earlier stages.

We conclude with a discussion on the utility of these approaches, and the differences between dealing with PN's and ordinary words.

The information sources used to construct lexical entries are shown below.

- Morphology. This gives information as to the syntactic class of an unknown word, and in some cases its meaning. Morphological processing takes place in the pre-processing stage.

- Syntactic Context. This is conveyed by the word class of surrounding words, and is used to derive the syntactic class of an unknown. The method of dealing with unknown words in FUNES is the same as that for handling known lexically ambiguous words, and is again carried out in the pre-processing stage.

- Semantic Context. This is conveyed by the selectional restrictions of the verbs and prepositions surrounding an unknown word. These constrain the semantic class of a word. For example, the object of an ingest verb is very likely to be of semantic class 'food'. Selectional restrictions are applied in the semantic analysis stage.

- Pragmatic Context. This is the highest level of understanding in the processing of Natural Language, consisting of the incorporation of the meaning of a sentence into the reader's conceptual framework as represented by the context produced from an understanding of previous sentences. Its use in the learning of unknown words is to supply possible hypotheses for their meaning based on an understanding of actor's needs and goals, and possible ways of achieving them.

Each of these information sources is considered in turn, and it is shown how FUNES brings each to bear to derive a syntax and semantics for unknown words. Most of the processes described here apply equally to both common nouns and PN's. However, as we will show in subsequent chapters, with PN's an explicit definition is usually provided in the text itself, and the initial capitalisation gives a clear indicator as to part of speech. Thus some of the methods described here will not usually be needed for the processing of PN's.

We begin with a discussion of Morphology.

## 4.2 Morphology

This is an area that has been relatively neglected in FUNES, and one in which more work could have been done, had the focus of the thesis not been on the acquisition of PN's.

An excellent account of the power morphological information can bring to NLP systems is given by Byrd in [26]. This shows how many multi-syllabic words are analysable as a combination of roots and suffixes. The rules by which morphemes are combined must consider various restrictions, such as subcategorisation and selectional restrictions. For instance Byrd describes a rule for deriving 'draftee' from 'draft'. This will only apply to transitive verbs which take animate objects, so the rule works for 'draft' and 'draftee', but not for 'sing' and 'singee'.

The aim of much work on word formation in NLP seems to be the removal of lexical redundancy, in that if a system is equipped with a good set of word formation rules, then the separate listing of morphologically related words is not necessary. As pointed out in [159], the decision on whether to include inflected words as separate entries, or to exclude them and allow syntactic and semantic information to be derived by morphological rules rests on the store vs. compute trade-off.

As Byrd points out though, such rules can also help in coping with new coinages and other unknown words. Many of the TU systems in the MUC initiative (reviewed in the next chapter) make use of morphology for just this purpose. In FUNES, some use is made of the ending of an unknown word in deciding on a syntactic and semantic class.

Initially, if a word is not found in the lexicon, various morphological rules are applied, and the derived root is then looked-up. These rules handle plural nouns, inflected verbs, and superlative and comparative adjectives. They lead to the discovery of a known root, and thus their successful application provides complete information on the inflected word, in terms of knowledge of the known root plus a number or tense indicator. Ludewig [111] has used the term inflectional morphology to describe this type of morphological knowledge.

If a word is still not found after their application, it is flagged as unknown, and passed on to a set of procedures which derive its part of speech. Here, PN's are detected by their initial capitalisation. If a word is not capitalised [1] various heuristics are applied to examine

[1] Henceforth we use the term 'capitalised' to refer to the convention in English of spelling PN's with an initial capital. If we wish to convey that a word is completely capitalised (e. g. BBC) we will state this clearly.

the word's ending. Although looking at the structure of the word, this is not morphological processing in the strict sense, as the aim is not to derive a known root, but merely to arrive at a syntactic or semantic classification. This process has more in common with work on the use of suffixes to indicate part of speech, as described in [116, 97]. Ludewig (previous citation) has used the term derivational morphology to describe this type of morphological knowledge. The endings considered are:

- 'ism/tion/ness/ship'. This leads to classification as an abstract noun, with gender n (neuter). (Many non-abstract nouns that end in 'tion', such as organisation, are already in the lexicon, and so will not be classified by this heuristic.)
- 'ist/ists'. This leads to classification as a 'role' noun, with gender 'e' (for either, i. e. male or female). Role nouns are nouns which primarily describe what people do, i. e. their occupation, for example 'pianist' or 'scientist', but the class also includes less definite roles or characteristics, e. g. 'philanthropist' or 'monarchist'.
- 'ment/ments'. This leads to a simple noun classification, with gender n. No semantic classification is attempted. The semantic class of such nouns is varied, and although many are connected in some way with the verb from which they derive, the nature of this connection is very variable.
- 'er/or' (or 'ers/ors'). In this case the root of the word is derived, and looked up to see if it is known as a verb. If so, the word is returned as a role noun, with gender 'e'.
- 'ly'. Leads to classification as an adverb.

A final use of morphology/word ending is made in cases where syntactic context cannot decide between a noun or a verb reading. In this case, the ending of the word is examined, and if it is 'ed' or 'ing' the word is returned as a verb, if not it is returned as a noun.

Morphological rules can be of great benefit to an NLP system in helping it derive information on unknown words. Few of the earlier automatic word learners described in chapter 2 make use of morphological constraints, relying instead on syntactic constraints to return a part of speech. Those systems aimed at robust text processing (such as the MUC-3 contenders), however, use it extensively. The reason few rules are provided in FUNES is due to constraints of time and focus. The focus of work has been on the use of higher level source of information and on handling PN's, to which conventional morphological rules do not apply. [2] In a complete NLP system the inclusion of a large number of morphological rules would be essential.

## 4.3 Syntactic Constraints

The acquisition of syntactic or part of speech information for a unknown word is achieved partly through the use of morphological knowledge as described above, and partly through the use of syntactic context. This is conveyed by the parts of speech of surrounding words.

It is an interesting question whether an emphasis should be placed on morphology or context in the classification of unknowns. FUNES places most emphasis on the latter. This involves fairly complex rules for the examination of the word class of surrounding words. It might be argued that a simple, purely morphological approach, classifying anything

[2] Although in restricted domains, such as that of Latin American terrorism used in the MUC-3 challenge, heuristics specifically for the identification of Spanish personal names were developed.

ending in 'ing' or 'ed' as a verb and anything else as a noun would succeed in the majority of cases, and be very inexpensive to implement. Such an approach is adopted in the SRI text processor, [87], where its success depends on the size of the lexicon employed, and the relatively closed domain, which means that very few unknown verbs will be encountered. This approach would not work in a more open domain, or with a smaller lexicon, both of which are the case with the FUNES system. It would, for instance, miss all tenseless verb forms, and would not cover those unknowns which are ambiguous between verb and noun, e. g. 'shooting'. It is in cases like these that the use of context is crucial to make a correct decision, to differentiate for example 'the shooting was horrific' vs. 'they were shooting at the protesters'. However there are cases where context is ambiguous, and here morphology is required, for example in differentiating 'he was CONVICTED of the murder' and 'he was GOVERNOR of the province'. The context being identical, a decision can only be made by use of the global 'if ed classify as verb' rule. It seems, therefore, that a combination of morphology and syntactic context is required to successfully handle unknowns.

After a word has been returned as unknown from the lexical look-up stage, and if it is still unclassified after application of the morphological heuristics described above, it goes through a series of rules that examine the word class of the preceding and following words. The same rules are used both for the classification of unknowns and for the resolution of known verb/noun or noun/adjective ambiguous words. This approach works as unknown words can be treated as verb/noun/adj ambiguous. It is an example of our attempt in FUNES to utilise general purpose mechanisms for the handling of unknowns. We view a facility of unknown word handling as an intrinsic and essential part of any NLP system, which should be built in from the start, rather than constructed as an add-on and ad hoc facility. In this we are in agreement with McDonald [120].

A full description of all the ambiguity resolution and unknown word classification heuristics is provided in appendix D. Here we just describe a single heuristic, to give the reader a feel for how the process works. In many cases, context can provide a definite classification, but in others it cannot, and FUNES must make recourse to more arbitrary heuristics, such as the use of the 'ed/ing' ending, or the likelihood of the syntactic construction a particular part of speech (POS) would imply. It will be noted that the only POS possibilities considered for unknowns are adjective, verb or noun. Closed class words can be ruled out for obvious reasons. Adverbs are not a closed class, but if the initial lexicon includes the majority of 'irregular' types (such as 'quite', 'rather' etc) then the simple heuristic of returning anything ending in 'ly' as an adverb works well.

In Table 4.1 we show the heuristics for handling an unknown word preceded by a relative pronoun. In this case the unknown could be an adjective, a noun or a verb [3]. If the next word is an adjective it is almost impossible to tell the POS of the unknown without having parsed constituents to examine, due to the number of words one might have to examine, and the ambiguities each might contain. It could be a verb, e. g. 'who **monitored** federal transport parking bay restrictions', or it could be an adjective, e. g. 'where **extortionate** foreign banking charges do not apply'. As the subject missing relative clause is far more common than the object missing type we return the unknown as a verb. Similar problems arise if the next word is a noun. The unknown could also be a noun (a noun complement to the RC subject noun) or a verb. Again the choice is complicated by the possible length of the phrase that follows the relative pronoun, e. g. 'who mercenary Tamil rebel leader Vellupillai Prabhakaaran had killed'. Even if we

[3] However, as FUNES does not recognise noun/adjective ambiguity, simply returning all such words as nouns due to the fact that even if they were adjectives they will still parse as noun complements, we only consider the noun/verb choice.

```
if unknown is last word in sentence
  then return as verb
else if unknown followed by an adverb or have, prog or modal auxiliary
        (but not 'to' or 'from')
        then return as a noun
else if it is followed by a preposition/determiner/pronoun/adjective
        then return as verb
else if next word is noun
        then if unk ends in 'ed/ing'
                then return as verb
                else as noun
```

Table 4.1: Heuristics for Classifying an Unknown following a Relative Pronoun

skipped through all the following nouns (hoping they are all known), the presence of the verb 'had' could not definitely indicate the unknown 'mercenary' was a noun, as if the unknown was a verb 'had' could be the main sentence verb or auxiliary. So the decision must again be resolved on a fairly arbitrary basis, using the heuristic 'if ends in ed/ing classify as verb, else noun'.

The contextual heuristics used in FUNES have, like much of the program, been derived from the analysis of news text. In this domain they have worked very well, mainly due to the relatively restricted style of the sort of stories analysed. Moreover, the majority of unknowns encountered have been nouns, and the heuristics are weighted towards a noun verdict in cases of uncertainty. The general nature of nouns appears to be more open-ended than verbs in news text. The lexicon currently holds about 560 verbs, which, while by no means adequate, has given quite reasonable coverage of the texts processed. I would estimate that a lexicon of around a thousand verbs would cover the vast majority used in normal news texts. Such a resource would make the processing of unknowns much easier, as they could almost automatically be returned as nouns (an approach largely adopted by Kuhns in [104]). Moreover, virtually all the unknown verbs encountered have been tensed, making their detection more straightforward than it would otherwise have been.

The approach taken for deriving the POS of an unknown in FUNES is based on previous word learning work (e. g. Granger [69], and Zernik and Jacobs [183]), and also on work in POS tagging (e. g. Garside and Leech [63], and Marshall [116]). The sort of rules utilised in POS tagging cannot always narrow down a word's possible POS to a single category by looking only at immediate neighbours. This is because they are developed to run on totally unrestricted text, and because they often use a large number of POS categories. We have developed our heuristics specifically for (and from) news text, and only consider three (or even two) possibilities for an unknown's POS. Coupled with morphological information, the heuristics have performed remarkably well.

### 4.3.1 Hyphenated Words

Hyphenated words present a large source of potentially unknown words, even for the largest of lexicons, due to their highly productive nature. There are an almost endless number of possibilities formed by the combination of common prefixes such as 'pre' and 'anti' with other words. In addition people are prone to coining all sorts of hyphenated words to describe particular situations, e. g. 'blue-painted', 'haggard-looking', 'HIV-resistant'. A proper approach to the problems such compounds present would take a thesis in itself. In FUNES we merely scrape the surface.

Hyphenated words (hence h-words) are returned from the tokeniser as two (or three) separate words, with the hyphenation indicated by a 'hyph' variable on the first word. So 'vice-president' would be returned as [(vice,word,hyph),(president,word,_)]. The first step in the lexical analysis is to look the h-word up in the lexicon. If it is known then all is well, and from here on the h-word will be handled as a single word.

If not, various types of h-word are considered, and checked for as follows:

- Check for prefix hyph, e. g. pro-Arab
- **else** check for plural hyph, e. g. hold-ups
- **else** check for name hyph, e. g. al-Huq
- **else** check for triple hyph, e. g. hand-to-hand
- **else** check for number hyph, e. g. fifty-five
- **else** return as unknown

Below we refer to the first word in the h-word as word-one, the second word as word-two. Prefix hyph deals with h-words where word-one is a common prefix such as 'pre', 'post', 'anti' or 'all'. FUNES checks for thirteen such prefixes (which could be extended at any time). If word-one is indeed a known prefix, the h-word is returned as a known adjective, in the form 'prefix(word two)', e. g. 'anti-Nazi' would be returned as 'anti(nazi)'. The prefix 'mid' indicates a noun compound, such as 'mid-term' or 'mid-air'.

If this is not the case, word-two is inspected to see if it is plural, if so the singular form is derived and the whole h-word looked up in the lexicon. If it is found then the relevant information is returned, so 'vice-presidents' would be returned as above, only with number set to plural. If not, the next possibility is that the h-word is in fact a name. This is signalled by both words one and two being capitalised, (as in 'Coates-Stephens'), or else by word-one being a known name prefix, such as 'al', or 'van'. If it is indeed a name it is simply returned as an unknown noun.

If not we check for the possibility of a triple h-word, such as 'hand-to-hand', or '55-year-old'. Examples like the second are not held in the lexicon but detected from the pattern 'Number-year-old'. Finally, if none of the above has yet been found, we check for number type h-words, such as 'twenty-two', 'five-star' or 'ten-day'. These are returned as known adjectives. Where word-two is of semantic category '[time]' the adjective can be given the semantic category 'duration'.

If none of these patterns are found, the h-word is just returned as unknown, and will be analysed by the procedures which look at syntactic context to derive a POS. Here also, h-words receive special handling. Some of this will be described in the following chapters on PN's, as certain types of word-two indicate an origin PN. These cases are checked for first, for instance if word-two is 'speaking' it indicates word-one is an origin word. If these checks fail, we check to see if word-two is a preposition, e. g. 'check-up', 'check-out'. If so the h-word is returned as an unknown noun, with gender set to 'n'. Such words could also be verbs, but if so tend to be written without the hyphen. If word-two is not a preposition we check to see if it ends in 'er/or' ('ers/ors') and has a known verb root, as described above for normal unknowns, e. g. 'gold-miners', 'sharp-shooters'. Finally, if still unknown the word is returned as an unknown adjective.

This completes the account of pre-processing of unknowns. At the end of the pre-processing stage a unique part of speech will have been found for all unknowns (if all the above heuristics fail the default is to return the unknown as a noun). The performance of the unknown/ambiguous word classifier was found to be very good in the evaluative tests described in chapter 11.

We next describe the use of semantic context to provide a semantic classification for an unknown noun.

## 4.4 The Use of Semantic Context

Semantic context is used to constrain the semantic category of an unknown noun. It reflects both word and world knowledge, which constrain the nature of events and situations conveyed by language. We know, for instance, that things which are killed must be animate (excluding metaphorical language), and that things which talk are also, for the most part, animate. Such knowledge can be included in the lexicon with the appropriate verb as a selectional restriction on the semantic category of its subject or object.

In a much broader way, prepositions also constrain the semantic category of a word that they govern. These restrictions are much tighter when the word in the PP is a PN, as explained in chapter 10. When it is not, they usually only permit the assignment of several possible semantic categories.

When an unknown reaches the semantic stage, it will already have been allocated a syntactic category in the pre-processing stage. The only POS for which FUNES attempts to construct a semantic category is the noun. As was discussed in chapter 2, the use of expectation-based heuristics (which is essentially what SR's are) is not applicable to the derivation of meanings for adjectives and adverbs, which are not expected in a sentence. We will discuss the possibility of acquiring verb meanings below. The first step in the semantic analysis of a sentence is the application of verb SR's to unknown subject and object nouns.

### 4.4.1 The Application of Verb Selectional Restrictions

Verb SR's are held with the verb itself in a 'Verb Register'. If a noun is detected as having no semantic category [4] at the start of semantic processing, then the SR's are used to create one. Each verb has two sets of SR, one for subject nouns and one for objects. These have been hand-coded in the lexicon.

Before the SR's are applied some global checks on compatibility between them and the gender of the noun, if it is known, are necessary. The unknown noun could have had its gender set due to morphological processing, or by being located as a pronoun referent. If the gender is m(ale), f(emale) or e(ither male or female) the SR must be a sub-category of [animate]. If the gender is n(euter) the SR must be a sub-category of [inanimate].

After this, the SR's of the verb are applied. This is a straightforward process of setting the semantic category of the unknown noun to the list of semantic categories held with the verb.

The main data structure for handling unknown common nouns is the Genus DataBase (DB). This is used for holding all the unknowns that have occurred in a particular story, together with the semantic categories derived for them. When a word is entered into the Genus DB, a check is made to see if an entry already exists for it. For PN's, this process can be quite complex, as humans can be referred to by any part of their name, and companies by various abbreviated forms (this variant form problem is discussed in

[4] An unknown may already have gained a semantic category from from an accompanying key-word (if it is a PN) or from one of the morphological rules described above.

detail in chapters 9 and 10). For normal nouns it is a simple matter of seeing if the same word is already entered. This can be seen as a simplified form of NP reference.

If the noun has been found to have occurred previously, the hypothesized semantic categories of the previous entry are compared to those of the new entry, and any common types retained. This process permits the acquisition of an unknown word to occur as an incremental procedure, with each new occurrence providing new hypotheses. Carbonell [27] discussed the desirability of such a facility, but his POLITICS system did not implement it. Zernik and Jacobs [183] lay great emphasis on the necessity of being able to refine a hypothesized meaning over multiple encounters. As the information that can be gathered on an unknown from a single exposure is limited and fairly vague it does not often suffice to provide an acceptable lexical entry, (this problem of the broadness of information provided by semantic context is also noted by Ludewig in [111]). However when the word has occurred several times, in several different contexts, there is more chance of arriving at a single semantic category.

As SR's are composed of fairly high-level semantic categories the comparison is mostly just a simple test of equality. However if this fails, we also test if one hypothesized category is a sub-type of the other, or vice-versa. We will look at an example, utilising the news extract below. [5] The unknown word we are concerned with is 'dispute'.

> 'President Particio Aylwin of Chile will meet President Carlos Menem of Argentina in Buenos Aires today to settle a territorial dispute first arbitrated in 1902 by King Edward VII, writes Alisdair Ross. The presidents will sign agreements on 23 other border disputes as part of a 1984 friendship accord...'

The first occurrence of the unknown is as the object of the verb 'settle'. The object SR's for this verb are 'human/abstract/loc'. These correspond to the examples :

1) 'The immigrants were settled in the poorest districts.'
2) 'They settled the dispute.'
3) 'The land was first settled in 1802'.

This occurrence will lead to the word being entered into the Genus DB as [[human], [abstract], [loc]]. The second occurrence of the word is within the PP 'on 23 other border disputes'. This is handled by the PP module for 'on', which returns a hypothesis of '[abstract]', due to the nature of the preceding phrase 'agreements on ...'. When 'dispute' is entered into the unknown register for the second time, the previous occurrence is detected and the two sets of hypotheses compared. The only common element is '[abstract]', and so this is retained as the new entry, and the old entry removed.

This approach works reasonably well, and is a good method of combining information from different contexts. Occasionally it will backfire, in the situation where we have one occurrence which actually gives the correct category, among others, and a second occurrence which does not have this category but has one of the others. For the most part though it serves to reduce the number of hypotheses that are entertained.

The above example shows the necessity of allowing automatically acquired words to have their meaning or POS updated. 'Dispute' is a verb as well as a noun, but this information could not be derived from the above example. If the lexicon is updated with a noun POS, a subsequent occurrence as a verb will cause the parser to fail. A compromise would seem to be the best solution to this problem, whereby if an unknown has occurred with the same POS on X occurrences, then its POS should be considered stable. Otherwise,

[5] This extract, and all the extracts presented in the thesis, are taken from the FUNES Development and Test Corpora. They thus illustrate the sort of text which FUNES is able to process.

a system will be forever monitoring the status of words in a text, which will lead to a very slow performance.

The use of a temporary lexicon can help with this problem. In FUNES, all acquired words are initially entered into a temporary store, which it is always considered possible to update. Words can be transferred from here into the permanent 'core' lexicon manually. Ideally this process should be automated utilising the above approach, whereby when a word has been used correctly on enough occasions it is automatically transferred to the core lexicon. The same method is used in the LILOG project [56]. Here automatically acquired words are entered into a temporary lexicon, from where they can form the basis for a corresponding entry into the core lexicon, either manually or automatically.

In FUNES, when the unknown has been entered into the unknown register, and comparisons made if necessary, the semantic processing of the NP-VP unit can continue as if all the words were known.

At the end of processing, all the words in the Genus DB are extracted and, if a single semantic category has been arrived at, the word is entered into the temporary lexicon.

### 4.4.2 The Application of Prepositional Selectional Restrictions

As mentioned above, when the PP noun is a PN, selectional restrictions can provide quite a specific semantic category for the unknown, e. g. 'in' plus a PN almost invariably indicates a location. However, when the PP noun is a common noun, it is much harder to specify a unique semantic category. The fact that this work has taken place in the sublanguage of news text does mean that many possibilities can be ruled out purely on the basis of probability. For instance 'with' can indicate various cases when it occurs in a post noun PP, but those for 'Wearing', 'Manner' or 'Feature' are far less likely than that of 'Works_for'.

FUNES uses special preposition-specific 'modules' to carry out the analysis of PP's. These deliver a case label for the PP (such as AT_LOC or FROM_TIME), and also derive a semantic category for the PP head noun if it is unknown. Appendix H contains specifications for the major case labels used in the FUNES system, showing exactly how they are returned.

Each module returns a case and a semantic category after considering the following items of information about the PP in question:

— its position, i. e. whether it was post noun or post verb.
— the semantic category of the verb
— the transitivity of the verb
— the preceding noun's semantic category and letter case
— the PP head noun's semantic category and letter case

The basic procedure in each module is to test for all the indicators of standard cases, mainly using the semantic category of the PP noun or the preceding noun. If the PP noun has no semantic category (i. e. it is unknown), its letter case is checked to see if it is a PN or not. If it is a PN, the semantic category returned is far more precise and accurate. If not, it is assigned the default semantic categories for that preposition. The defaults are shown below for various prepositions :

BY : [[loc],[human]]
AS : [role]
WITH : [[object],[abstract]]
TO : [[loc],[human]]
BETWEEN : [[loc],[human]]

These defaults are, for the most part, quite vague. As FUNES emerged from running on artificially devised examples to real world text, it became apparent that the number of cases flagged by many prepositions were numerous, and thus the various possible semantic categories for an unknown head noun were also numerous. Specific patterns around a PP can restrict the possible semantic categories, as we show below:

- If the preposition is 'for', the verb sem_cat 'poss' and the PP contains a number, the unknown noun is returned with sem_cat [money].
- If the preposition is 'from' and the PP is sentence initial and the PPnoun is unknown, the unknown is returned as [loc]
- If the preposition is 'from' and the PP noun is not a PN, it is returned as [human]
- If the preposition is 'on' and the PP is postnoun and the preceding noun sem_cat is corp/abstract/event the unknown is returned as [abstract]
- If the preposition is 'during/since/until/after/before' the unknown is returned as [event]

However, it remains a fact that prepositional selectional restrictions alone are not a particularly good way of deriving information on unknown common nouns.

### 4.4.3 The Acquisition of Verbs

FUNES does not attempt to acquire a meaning for verbs. It does, however, derive a part of speech for unknown verbs. At present this is not used for lexical update. Provision of morphological heuristics to derive the correct verb root from an inflected form, and then to derive the other inflected forms, would produce a lexical entry giving part of speech information and the inflected forms. As FUNES has performed well in deriving the correct POS for unknown verbs, such a step would be a viable extension of the system.

However, this would only provide one part of the information required for a verb entry in the lexicon. A full entry would also need transitivity information and selectional restriction information. (Although semantic categories are listed for verbs, this information is not used extensively, and for many verbs, the semantic category is simply the verb itself). The acquisition of this information is possible, as has been shown in [169, 168, 23, 77]. Indeed, acquisition of such information from textual occurrences is in many ways more desirable than entering it initially in the lexicon, as real text may present examples which were not anticipated by a lexicographer.

Initial entries could be created from the context in which the verb was encountered. If it occurs with a direct object, then transitivity is transitive, if not then it is intransitive. Obviously such entries must be capable of adaptation as so many verbs are both transitive and intransitive, and can also take embedded complements. Below we illustrate this process with an an example input containing the unknown verb 'traced':

> 'Mr Vernon Mwaanga, Zambia's new Foreign Minister, told the Daily Telegraph yesterday that Cebekhulu had been traced to Lusaka Central Prison, where he has been held for several months.'

'Traced' was correctly returned as a verb by FUNES, due to its following the known word 'been'. Its occurrence in this context would also provide the information that it can be transitive. In addition an object Selectional Restriction could be formed of [human], as in this example the object of 'trace' is a human. The passive construction would prevent the formation of a subject SR on this occasion.

Such an approach would be similar to that described in [169, 168], in that both seek to build SR's from examination of textual examples. However it would differ in that Velardi et al describe a pre-processor which is fed a large number of examples, and which automatically produces a lexicon, independently of any text processing system. Here we would be considering acquisition within a text processing environment. Brent [23] also describes a lexical pre-processing tool, but this is solely aimed at producing subcategorisation (transitivity) information. It works in a very different manner, but the principle of utilising textual examples is the same. Finally Hastings et al. [77] utilise the context of occurrence to place an unknown verb in a hierarchy, with particular reference to the semantic category of subject and object.

Therefore we would consider such an approach promising. It has not been developed in FUNES due to the focus of the work being on PN's. In addition it is felt that the need for work on verb acquisition is lower than the need for work on handling PN's. This is because PN's represent a far more open class of lexical entity than do verbs, particularly in news text. Moreover they have received much less research, further increasing the motivation for concentrating on them rather than other word classes.

So far in this chapter we have described heuristics and methods that are utilised in the present implementation of FUNES for handling unknown words. In the final section we describe an approach that is no longer implemented, but which represents an interesting exploration of the use of higher level knowledge sources in the processing of unknown words.

## 4.5 Application of Pragmatic Knowledge

This section considers the use of high-level understanding of basic human goals and plans to aid in the acquisition of specific semantic categories for unknown nouns. This work was carried out in the early stages of the project, before PN's and their accompanying descriptions had been identified as the main focus of the work. At the present time therefore, it remains a relatively un-explored avenue, albeit one with interesting potential.

As can be seen from the previous sections, although acquisition of syntactic information on unknown nouns is a possible and now proven process, the acquisition of anything but a very broad semantic category remains problematic. This is due to the great flexibility and ambiguity of language, and to the fact that the syntactic and semantic levels do not regularly supply enough information to produce a detailed semantic description for an unknown word.

This realisation lead to a quest for higher level information sources. Granger had utilised scripts for this purpose, although they did not really produce much of an improvement on SR's. Carbonell had used plan and goal information to much better effect. A similar approach was adopted in FUNES. The original application in which this approach was explored was the processing of simple 2 and 3 sentence stories describing everyday events, such as buying and selling, eating, and travelling.

The basic approach was to utilise a rich knowledge base containing detailed information on goals, plans, preconditions and results. This would be used to create a context for the story being read, in which the actions described could be understood in terms of the goals of the actor in the story, and the plans needed to bring about successful resolution of these goals. Each new sentence is connected to the ongoing context in various ways. It could be by the sentence fulfilling an earlier precondition, by its being a plan for an ongoing goal (or a goal of an ongoing plan), or by it being identified as part of an active script. The basis of the method was, after the processing of an input, to see if any predictions made during processing remain unsatisfied. If so, and if an unknown occurred in the correct situation to potentially satisfy the prediction, it was assumed to do so and assigned the meaning required to fulfill the prediction.

Some examples of how precondition, plan and goal information can derive meaning based on unfulfilled predictions should clarify the process.

A precondition for a purchase event is that the purchaser is likely to be located in a shop. Given an input

'On saturday I drove into town. I went to Dillons where I bought some books'

in which 'Dillons' is unknown, the precondition of 'bought' is 'at(shop,I)'. This however is not fulfilled in the above input. However an inference rule will have been activated by the occurrence of a PTRANS [6] verb (drove) with a 'to' type preposition. The effect of this rule is to infer that the agent of the verb is now located at the location flagged by the 'to' preposition. So FUNES infers 'at(dillons,I)'. In the absence of any further input, an attempt is made to satisfy the precondition by mapping 'Dillons' to 'shop'. Moreover, the sentence also supplies the differentia information that it is a shop that sells books.

In this implementation FUNES goes beyond the semantic stage to a pragmatic stage, in which it makes inferences from the sentence it has just read. These inferences are of various kinds. 'Immediate' inferences are like those connected with the PTRANS verb above, many verbs have such immediate inferences connected to them. These inferences deal with the immediate effects of verbs. Other inferences are concerned with preconditions, plans and goals associated with common verbs and adjectives. In the above example we saw how the precondition of 'buy' is that the buyer be in a shop, this was fulfilled by the result of a PTRANS verb.

Whereas existing verbs in the lexicon have associated preconditions, existing nouns have associated activities or functions. The noun 'shop' is associated with the primitive act 'ATRANS', the noun 'gun' with the act 'SHOOT', and so on. Consider an input subsequent to the above like

'At Dillons I got the new Bukowski novel'

in which 'got' is unknown (or at least its purchase sense is unknown). A predicted act for the person located at Dillons is an ATRANS, the mapping of 'got' to this act will permit the correct analysis of the sentence (or the resolution of the anomaly produced by the use of an unknown sense of a known word). The above sentence also provides some grammatical information on the word 'got', that it can be used as a transitive verb and takes an animate subject. This information could be gained by inheritance from ATRANS, but the above example provides clarification. From this example it can be seen that this approach enables the learning of alternative meanings for known words, as well as the learning of entirely new words. The learning process is the same in both cases. It is the detection of the unknown which differs.

Knowledge about human goals also provide much information as to meaning. For example a goal of someone who is ill is to get better. Plans to do so are to visit a doctor or to take some medicine. In an input like

'Jim had a bad headache. He took four aspirin.'

[6] In this implementation some of the primitives from Schank's theory of Conceptual Dependency [148], and the inference rules connected with them [149] were adopted. PTRANS refers to verbs of movement (Physical TRANSfer), ATRANS to verbs of exchange of goods (Abstract TRANSfer) .

where 'aspirin' is unknown, the plan 'ingest(medicine)' remains unfulfilled. However there is an 'ingest(aspirin)' event described, and with no further input specifically describing an 'ingest(medicine)', the plan can be fulfilled by assigning aspirin as a type of medicine. Use of plan/precondition chains can also cope with inputs like

> 'Jim had a bad headache. He went to the store where he bought some aspirin.'

Here there is no 'ingest(medicine)' nor is there any ingest event at all. However a precondition of ingest is 'have' and a plan for have is ATRANS. The object of the ATRANS is 'aspirin' which can again be mapped to medicine, the object of the original plan for curing the headache.

Scriptal information is used in a similar manner. For instance the 'get_takeaway' script is indexed as a plan for the 'relieve(hunger)' goal, which is activated upon processing a sentence describing a hungry state. Given an input like

> 'I was starving when I got in from work. I phoned Pizza_hut.'

various plans for relieving hunger will be activated. A plan which describes a phone event is 'get_takeaway' which has 'phone(restaurant)' as a possible step. As above, in the absence of explicit explanation of the second sentence it can be explained by assigning Pizza-hut as a restaurant, which then enables the second sentence to be explained as a step in a takeaway script.

The use of these techniques in deriving meanings does not require radical new procedures, or a huge amount of additional processing beyond that required in an understanding system without the ability to learn new words. The attempt to connect sentences through the use of preconditions, consequences, goals and plans is common to many natural language understanding systems. The only additional action required is that taken when an input cannot be connected to the ongoing context. This action consists of checking for the presence of an unknown and seeing if it occurs in the right form (e. g. it is the LOCATION of a buy event or the THEME of an ingest event) to enable the unexplained input to be explained. If it does then it is given the meaning required to explain the input. This not only enables a richer understanding of the text as more sentences are connected but it also enriches the lexicon with new words.

The next section discusses in more detail how the methods discussed above are implemented. Although the system described below works on a marker passing framework the approach advocated should be applicable to any implementation of inferential processes.

### 4.5.1 A Marker-Passing Framework for the Inference of Word Meanings

The pragmatics stage described above attempts to fit the current sentence into the existing context by seeing if it fulfills any existing predictions, and then deriving predictions from the sentence itself. This stage is implemented as a marker-pass through a hierarchically structured knowledge base, in a manner similar to [85, 80, 35, 55].

Marks are passed from all nouns and verbs in the sentence to concepts related to them. The problem of limiting the spread of marks is solved via the use of Norvig's marker state concept, as described in [129]. This assumes the existence of pre-defined 'meaningful' path shapes and a finite state network controlling the spread of marks. When a mark is passed to a node that already has a mark on it an intersection occurs signalling a possible inference. Marker passing was chosen as the method of implementing 'understanding' because of its ability to handle many different forms of knowledge (such as scripts, plans and goals) in a unified way. The advantages of weak methods of inference are discussed in [106, 129].

Charniak [34] has pointed out that marker-passing can be done in parallel with the rest of the processing as soon as a word is read in, as it makes no use of functional structure. However this is not the case for unknowns. As they are unknown they do not exist in the knowledge base and therefore there is nothing from which to pass marks. Therefore they must wait until the semantic component has derived a category (or categories) for them and then marks are passed from this.

Making inferences is a matter of tracing the paths of nodes involved in intersections back to their origins. As passing is constrained by the use of a marker state concept the intersections that occur are always meaningful, although they do require the checking of variable bindings as functional structure is lost in the marking process. The assignment of meaning to unknowns is carried out when a path from an unknown intersects another path. This will be illustrated with reference to some of the examples described in the previous section. In the input

> 'Jim woke up with a splitting headache.He took four aspirin.'

the following paths are obtained

> [headache(a23) inst headache,headache isa ill,ill has_g relieve(ill),
> relieve(ill) has_plan ingest(medicine),ingest(medicine) arg medicine,
> medicine isa edible – aspirin(a12) inst aspirin,aspirin isa edible]

> [headache(a23) inst headache,headache isa ill,ill has_g relieve(ill),
> relieve(ill) has_plan ingest(medicine),ingest(medicine) isa ingest –
> took(a13) inst take,take isa ingest]

The first path leads to an intersection at 'edible', (aspirin is classified as 'edible' due to the selectional restrictions of the verb 'take'). Intersections involving unknowns are allowed on 'isa plateaus', otherwise they are prevented. The path to 'edible' which did not involve an unknown is traced to find the most specific category, i. e. that at the root of the first 'isa' link, and the unknown is assigned to that category, here aspirin is assigned as medicine. In the second path variable checking reveals the Theme of 'took' to be an unknown, this is then assigned the meaning of the argument of the known triple involved in the intersection (i. e. 'aspirin' is again assigned as medicine).

A second example comes from the the British paper the Independent. It is intended to show the viability of this approach in a more realistic area. The extract is:

> 'Paruyr Ayrikyan,a former Russian political prisoner,was elected to the Armenian parliament on monday. He had spent ten years in a gulag. His release was ordered by the prime minister.'

The unknown word is 'gulag'. As this occurs after the preposition 'in' and has an indefinite determiner it is given the category LOC (if there was no determiner it could also be TIME, as in 'in July'). This leads to several intersections, with 'Armenia', 'Russia' and 'prison', giving two possible meanings — country or prison. As the word occurred with a determiner, the country meaning is less preferred because we do not commonly talk about countries and use an indefinite determiner (e. g. 'a US', 'a Russia'). This leaves the prison definition. The relevant path is:

> [prisoner(a12) inst prisoner,prisoner inhabit prison,
> prison isa building,building isa loc,
> – gulag(a34) inst gulag,gulag isa loc]

As the unknown occurred in a story about Russia a flag could be attached to the lexicon entry indicating 'gulag' is possibly of Russian origin. Further examples of this implementation are contained in [41, 42].

### 4.5.2 Limitations of the Approach

The above method was almost entirely developed and tested on toy-world examples. This is commonly the case with marker-passing approaches. This is because the major problem with the approach is limiting the spread of marks. Even with a marker-state concept, when the input is a realistic and long sentence (such as that from a news story), there is a real danger of the whole Knowledge Base becoming activated. This is increased by the presence of unknowns, which can only pass marks from the high-level semantic categories derived for them from the semantic processing stage. In one way, therefore, the success of the marker-passing framework is dependent on the success of the semantic stage in producing a correct (and preferably unique) semantic category for an unknown. As the previous sections described this is not often possible.

The other problem in moving from toy domains to real domains is the proliferation of goals, plans and preconditions. Real texts do not occur in the pre-arranged format of toy domain examples, which permit an easy construction of precondition chains. However, this is not to dismiss such an approach — if higher level understanding of text is ever to be achieved, a goal and plan-based analysis of text is a crucial step. Texts are structured so that each sentence fits into the context created by the previous sentences, in a similar but more complex manner to those examples described above. Such methods, however, need considerably more development before they can be viably used in unconstrained domains.

Finally, as regards the derivation of meanings for PN's, extensive study of real world news text has shown that very often these are described/defined within the text in which they occur. It was concluded therefore, that a better way to handle PN's was to capitalise on these 'ready-made' definitions. It is this endeavour that is described in the rest of the thesis.

## 4.6 Summary and Conclusion

This chapter has described the use of different sources of information to derive meanings for unknown common nouns. We have shown how the final meaning can be thought of as evolving through the different stages of processing which occur in the task of Text Understanding. This meaning evolves from:

- Morphological Information: the ending of a word can provide clues as to its part of speech, and sometimes its semantic category.

- Syntactic Information: the syntactic class of surrounding words can provide information as to the syntactic class of an unknown word.

- Semantic Information: the verb (or preposition) which governs an unknown noun can contribute restrictions on its semantic class. Over several encounters these can be narrowed down to given a unique class.

- Pragmatic Information: the context into which an unknown word fits can provide a more detailed meaning in terms of its role in an active plan.

All of these levels except the last are utilised in the current implementation of FUNES, both in the processing of common nouns and PN's. However, without the detailed 'within text description' which a PN receives, the meaning arrived at for an unknown common noun can often be quite vague. It may take several encounters to arrive at a single meaning. These restrictions lead us to conclude that the above methods are an adequate and reasonable solution for coping with unknowns, but they in no way obviate the need for an initial lexicon, which is both large and detailed. (This conclusion does not apply to PN's). Such methods are an important and necessary part of any system intending to cope with real text, but they should only be used to cope with those (infrequent) cases which are not in the lexicon. A large lexicon, coupled with a powerful morphological processor, should cover the majority of words encountered. Methods like the above can be used to cope with those cases which still remain unknown. However, it does not seem viable to attempt to use such methods for large-scale lexical acquisition. In addition to the above reservations, it should also be noted that the quality of the definition produced degrades as the number of unknowns in a sentence increases. As long as a sentence only contains a single unknown, the chances of a reasonable syntactic and semantic class being produced are good, but if it contains two or more unknowns we lose a base on which to build any semantic representation.

These reservations do not apply to PN's. Firstly, there are few large PN lexical resources available for use. In addition the number of PN's, and the number of new PN's occurring each day, renders the use of static lexical resources untenable. Finally the complexity of their structure, and the structures in which they can be embedded, mean that even if the PN is known, considerable analysis is still needed to produce a reasonable analysis. These factors all indicate that PN's require special handling. The remaining chapters in this thesis show how we feel this should be carried out.

# Chapter 5

# The Proper Name in Linguistics and Computational Linguistics

## 5.1 Introduction

In this chapter we discuss previous work on the subject of PN's. One of the most distinctive feature of the PN is that it has virtually been ignored in Computational Linguistics until very recently. We account for this by the fact that it is only in the past few years that research has been seriously directed at unrestricted, naturally-occurring text. It is only in this sort of text that the PN becomes noticeable as a problem, both because of its ubiquity and complexity.

Other areas of Artificial Intelligence/Computer Science have looked at the 'PN Problem' more closely, in particular speech research. We shall not examine this work in any detail as it is not of great relevance to the problem of handling PN's in textual input. We begin the chapter with a review of the work on MRD adequacy that has touched on PN's.

## 5.2 The Inadequacy of a Static Lexical Resource for Coverage of Proper Names

The work of Walker and Amsler [172], Sampson [146], and Seitz et al [151], discussed in chapter 2, clearly showed that MRD's give poor coverage of PN's.

This conclusion is supported by Walker and Amsler's finding (related at at the 7th University of Waterloo Conference) of the non-boundedness of news text vocabulary. Walker related how he and Amsler conducted an experiment based on their 10M word New York Times sample. They processed large sections of this sample at a time, deriving all the words within each sample as they went, and then comparing these to the words in the next sample and adding any new words. They found that the growth curve for the dictionary never flattened out — new words kept on appearing. Brian Slator in [158] also discusses this finding, saying (my italics):

> '... another part of us may be quite willing to accept that the trickle (of new words) will never stop — if only because *there are always new proper names in the news, new people, new companies, new countries.* '

Sampson concluded his assessment of MRD's by stating that the process of adding PN's to a dictionary would have to be taken to

> 'heroic and impractical lengths before one would come within sight of covering all names encountered in practice'.

When analysing news text even this is an understatement. The number of people, place and company names (to identify but a few of the more common PN categories) in the world is immeasurably vast, and in addition they are constantly changing as new figures and companies appear, and old ones disappear.

One of the conclusions of Walker and Amslers study was that lexical coverage would be greatly improved by the use of a lexical resource for PN's. In [10], Amsler reports on a study which directly compares PN's occurring in news text to those listed in 'The World Almanac and Book of Facts' (referred to as WA85). A program was developed to detect PN compounds in text, and those extracted from one month's worth of the New York Times were compared to the listings in WA85. The results were very surprising (if not alarming) — the WA85 contained only 10% of the PN's from the newswire. Closer analysis showed much of this was due to a problem of 'near misses', where a PN can have several different forms only one of which is contained in the dictionary, but all of which are used in the news text. This work seems to shows that even the provision of vast lexical resources of PN's will not provide a simple solution to the PN Problem.

Researchers in the field of speech have highlighted the problems people's names present for lexicon construction. Church [40] shows that while the most common 2,000 names can account for 46% of the Kansas City phone book, it would take a huge jump to 40,000 names to obtain 93% coverage. Liu & Hass [110] conducted a study on 75 million US households, in which they showed that the first few thousand names accounted for 60% of the sample, but it would take 175,000 names to cover 88.7% of the sample.

These studies tell us several different things. Those looking at the adequacy of MRD's clearly show that standard dictionaries provide little or no coverage of PN's. Thus the high occurrence of PN's in news text particularly will provide problems for systems attempting to analyse that text. In addition, the work of Amsler et al. has shown that even a dictionary resource comprised entirely of PN's does not overcome this problem, due to variations in the form of PN's. Static lexical resources will also be hampered by the constantly changing nature of the PN's in the world, with new people and companies constantly appearing. Finally work in the field of speech processing has shown that to obtain high coverage of personal names (and by high we only mean around 90%) will take vast lexical resources.

The conclusion we draw from this is that any approach which seeks to overcome the PN problem through provision of a large lexical resource will work poorly at best, and will also hamper the rest of the system's processing through the size of the lexicon required.

In the next section we begin our review of previous work on PN's by looking at linguistic accounts of the nature of the PN.

## 5.3 Proper Names in Linguistics

Work in Computational Linguistics is commonly based on work in conventional linguistics, and so some discussion of linguistic work on PN's is in order. Unfortunately very little of relevance is to be found. Although there is a huge literature on PN's, it is almost all concerned with the question of reference and meaning, and the long-running debate on whether PN's have meaning in the same way as normal words (see [124, 60, 144, 103, 150]). An excellent summary of this work is given in Allerton's paper (see below), pages 69-73. However this debate is of no relevance or help to the present endeavour. In particular such work refuses to consider any sort of PN beyond the personal name (such as 'Aristotle'), a restriction which seriously undermines its worth. As the authors below point out, it is this concentration on simple Proper Nouns that is partly responsible for the conclusion that Proper Names have no meaning in the same way as do normal words. While this may be true of a word like 'Aristotle', it does not appear to be true of a Proper Name like 'Centre for Media Studies'.

We will examine the work of three researchers who have done more than simply consider the meaning/denoting/connoting debate.

The work of John Carroll at IBM is the earliest example of a wider-ranging enquiry. This work was actually the original inspiration for Amsler's reflections on PN's described in section 5.5. Carroll's book 'Whats in a Name' [30] was a psychologically oriented investigation into the naming process. It also included consideration of the grammatical structure of place names and the variant forms they can take. Carroll appears to be the first researcher to really look at the structure of place names. He analysed such a name as consisting of a category word and a name-stem. The category word is that which classifies the name into a particular type, e. g. 'square' in 'Trafalgar Square'. The name-stem is the non-classifying part of the name, e. g. 'Trafalgar' in 'Trafalgar Square'. If the name-stem can be used to stand for the whole name, it is called a namehead. What principally interested him were the rules which governed the formation of nameheads, and why it is that some names do not have nameheads while others do.

Carroll presented evidence to show that the namehead phenomenon was neither entirely lexical nor grammatical in nature. He proposed that it was governed by a system of rule-schemas. These are approximate rules or strategies. It is clear that namehead formation rules apply across categories rather than on a name by name basis. Thus squares, harbours, beaches and straits (to take a few of his examples) do not have nameheads, while bridges, newspapers, mountains and canals do. (The inclusion of bridges here appears to apply only to American names, names of bridges in London, for example, do not have nameheads.) Two other important factors Carroll mentions in the operation of these rule-schemas are the presence of the definite article and the position of the namestem in relation to the classificatory component of the name. The presence or absence of the definite article can also be looked at across categories of names, so rivers, canals, museums and newspapers require it, while airports, lakes, parks and prisons do not. The position of the namestem can affect the formation of a namehead within a category, thus 'Mount Everest' has the namehead 'Everest', but 'Bear Mountain' does not have the namehead 'Bear'.

Carroll observed that the shortening process is 'pragmatically constrained' in that it must provide the reader/listener with a clear referent. Two properties only touched on in the book, which also have strong pragmatic influences, are familiarity and uniqueness. These can override the rules derived from the category of place word. We shall consider these in chapter 6. Here it suffices to state that familiarity enables one to shorten a PN that would otherwise not permit it, whereas uniqueness prevents shortening of a PN that would normally permit it. If the namehead does not provide a unique identifier the shortening will not take place, e. g. 'Arctic' is not a sufficient variation for 'Arctic Ocean' (unless one is listing all the oceans), but the category of Ocean will allow such a variation if a unique namehead is left (as in 'Atlantic' and 'Pacific').

Allerton [5] considers several aspects of the PN, in particular how can the class be delimited, what are the subcategories within it, and is it a valid part of the language. He points out that PN's differ from normal words in that their structure is fixed, whereas normal words exhibit a potential for structural variety (e. g. plural forms, presence/absence of determiners and quantifiers). He proposes four syntactic categories of PN, shown in Table 5.1. This appears to be the first work to identify and discuss the whole range of PN's, clearly showing that many PN's can be composed entirely of common nouns and

| | |
|---|---|
| Pure PN's | Basically proper nouns, with or without titles<br>e. g. the Hague, Michael, Paris |
| Mixed PN's | A combination of proper nouns plus common nouns<br>e. g. Hyde Park, Charles the Second, the River Thames. |
| Common-based PN's | Composed entirely of common constituents<br>e. g. the Black Sea, Central Park |
| Coded PN's | Acronyms, combinations of letters and digits<br>e. g. M25, BBC |

Table 5.1: Allerton's Proper Name Classification

adjectives. The next chapter of this thesis describes the nature of such PN's in more detail. Allerton also outlines a semantic categorisation of PN's, along very similar lines to a categorisation presented by Choueka [38] (described below) and to that presented in the next chapter. He contends that the main semantic test for PN membership is that PN's only refer to individual entities (corresponding to count rather than mass nouns). As regards their position inside or outside the language, Allerton concludes that some PN's, in particular those which are known to the whole population and which exhibit polymorphy (i. e. which have translations in other languages), are, but others are not. His semantic categorisation is graded accoring to these criteria, with temporal PN's and organisation PN's being the most definitely within the language, and personal and vehicle PN's least within the language.

Work by Marmaridou [115] is also of interest. This is an investigation into the nature of PN's and their relation with other lexical items. It is notable for considering the whole range of PN's — place names, company names, object names etc. — as well as the typical class of personal names. Marmaridou begins by considering two criteria for inclusion as a PN — uniqueness of referent and capitalisation. She concludes that these are insufficient and consideration of the meaning of the name and its social function is also required. Consideration of the social and communicative function of the PN leads her to conclude that PN's do have a meaning, one that is 'constituted in metalinguistic knowledge about the category itself'. PN's are integrated into Fillmore's theory of Linguistic Semantics, and are thus considered to activate an 'experiental scene' when heard or read, just like any normal word does in Fillmore's theory. But the PN activates two kinds of scene:

1) Contains encyclopediac information about the specific referent of the name
2) Contains knowledge about the category of the PN itself, including knowledge about prototypical instances of members of this category

So when someone says 'I met George this morning', even if we do not know who they are referring to, the name activates the second type of knowledge, so we can still understand the sentence.

The theory is then expanded to include work from Prototype theory [143] and Lakoff's work on Idealised Cognitive Models (ICM's). Marmaridou proposes that membership of the class of PN's is not absolute but graded. The personal PN (e. g. 'Aristotle') is the ideal prototype PN, maximally distinct from the prototypical common noun, whereas less prototypical PN's (such as 'Ministry of Defense') are more similar to common nouns. Those PN's which are most protoypical are held to be most central to the 'PN ICM' and to have the simplest linguistic form, while the less prototypical types, which are less central to the ICM, require more complex linguistic forms (e. g. company and book and film names). Allerton's categories of PN are recast in this most prototypical—least prototypical spectrum.

In our view this theory of PN's is the most convincing proposed to date, and is one that contributes far more than much of the previous meaning/denotation/connotation debate.

A recent paper by Evans and Wimmer [57] has also highlighted some of the problems discussed above. The thrust of the paper is to highlight the lack of linguistic work on the form of PN's. They point out some serious obstacles to a linguistic theory of PN's, stemming from ignorance of the following areas :

1. the syntax of PN's, particularly the relationship between Proper Names and Proper Nouns.
2. the semantics of PN's, especially the problem of PNs that do not have the semantics of their constituents (e. g. Red Bridge, Three Bridges).
3. the relationship of PN's to the normal lexicon, e. g. the entry of PN's into the normal lexicon as normal words (shanghaied), and the use of normal words as PN's (names like Green, Thatcher etc).
4. the act and function of name-giving.

Subsequent chapters of this thesis seek to address some of these shortcomings.

In the next section we briefly describe work in other areas of Artificial Intelligence and Computer Science that has concerned itself with PN's.

## 5.4 Proper Names in other areas of Computer Science

In speech, personal names have been considered as they pose a problem for both understanding and generation. In understanding, the problem is the sheer size of the lexicon required to cope with every possible name. Studies looking at this problem have been mentioned in section 5.2 above. In generation, the problem comes from the violation of normal pronunciation rules that personal and place names produce. This is largely due to their geographic origin. A country like the US has people from almost every country in the world, and their names must be pronounced according to the rules of their mother language. Establishing the correct language group is a major problem. Work addressing this problem is described in [170, 161].

This problem also besets database search for personal names, for example public record or police databases. Foreign names are prone to misspellings, and in addition possess variant forms that are often used. If such variants are to be corrected it is important to elicit the language the names belong to. In [131] Oshika et al. use Hidden Markov Models to determine a names nationality, and then language specific rules are able to generate plausible spelling variations.

This variant form problem is also present in the problems of automatic museum cataloguing of artists names. The format and spelling of personal names varies greatly from one institution to the next, as the form and spelling have been changed over many years. If historical information is to be shared these variants must be matched successfully. The variations are numerous — in spelling, in name order, in name form, in number of middle names, in punctuation and so on. An impressive attempt to produce an automatic matching algorithm is described in [154, 72].

Interesting as such work is, it is not of direct relevance to processing PN's in a computationally linguistic environment. In the next section we describe more directly relevant work.

## 5.5 Proper Names in Computational Linguistics

Until very recently (1991-92) PN's have been very under-researched in this area, having been considered in passing by only a few researchers. It is interesting that the only work which has been done on PN's has been carried out in the context of text understanding of un-edited text. This first occurred at Yale in the late 1970's, and more recently at General Electric, and NHK (Japan). This observation strongly supports our contention that it is only in work on real, un-edited text that the PN becomes noticeable as a problem.

One of the first papers specifically mentioning PN's, and the analysis of the syntactic structures in which they can occur, is due to Gershman in 1977 [67]. He describes a complex noun group analyser that is partly aimed at dealing with newspaper stories and headlines. The analyser acts as a component of the previously discussed SAM and PAM natural language processors. SAM deals with two classes of noun groups, divided by the type of conceptual structure they produce - Picture Producers (PP's) and Concept Producers. It is notable that a large number of these deal with PN's. Such an approach *differs very much* from previous (and future) NLP work which concerns itself very little with PN's.

The 7 classes of PP recognised by SAM are #PERSON, #PHYSOBJ, #ORGANISATION, #LOCALE, #ROAD, #GROUP, and #POLITY. All of these except #PHYSOBJ and #GROUP will be mainly PN's. The noun group analyser is expectation-based — each word carries with it expectations of what words will follow it and how they relate to the present word. Like all the Yale work it is a semantically oriented parser, having no separate syntactic stage. Much of the systems knowledge is contained in the dictionary entries rather than the grammar rules. The problem with such an approach is the huge effort needed to extend it to cover various domains, as every new word added requires a huge amount of detail. Conceptual Dependency lexical entries are not the sort that could be compiled from MRD's.

Gershman gives examples of how the processor handles very complex PN appositive constructions, but it seems largely assumed that the majority of the PN's are known. Although some consideration of dealing with unknown PN's is given, it is very hard to tell how general are these facilities. For instance, examples are given using certain key-words such as 'Dr', 'Avenue' and 'Ambulance' to classify neighbouring unknown words as names of the corresponding syntactic category. But it is not clear whether these are special-purpose rules for these particular words, or if similar rules apply to all such words. Beyond the use of key-words, context is used to classify unknown's by handing them over to Granger's FOUL-UP program.

Gershman's paper is impressive in being the only paper I have seen (prior to 1990) to deal extensively with the problems PN's present for NLP of real text. It was unfortunate to be tied to a paradigm that came to grief due to the complexity and incompatibility of its knowledge structures, and the huge task of extending these to cover new domains. Unfortunately he provides no results to show how general are the PN handling abilities of the system. The rules he does show for splitting surface noun groups will fail in some cases, e. g. the rule that a number can not be preceded by a Name in the same surface noun group will fail on cases like Edward VII, or Saturn V. Little consideration is given to lexical ambiguity, and especially PN/other part of speech ambiguity (for instance how to prevent the known definition of 'tornado' when analysing 'Tornado fighter-bomber', and yet allow the existing definition of 'Civil Liberties' in 'National Council for Civil Liberties'). Finally, no mention is made of conjunction handling. PN expressions containing conjunctions are very common, e. g. 'the Vietnam and Korean Wars', 'X and Y, the presidents of A and B,' or 'the planets Neptune and Pluto'.

A more recent paper that has some consideration of PN's is the TICC project of Allport [6]. This is a text understander of real text — that output by the local police force describing traffic accidents. It produces short summaries of text describing a traffic accident and outputs these for local motorists. The input text is highly abbreviated and very agrammatical. Consideration is given to words that are not in the system's lexicon, some of these are identified as PN's through the use of local words that clearly describe the unknown as a name. This is very similar to Gershman's approach. Allport describes the use of Title words, such as 'pc' and 'sgt' to classify the following word as a human name (in this case a policeman) and location words such as 'lane' and 'avenue' to classify a previous unknown as a lane or avenue name. This approach appears to be used in all of the few systems that have considered the PN problem, but it is usually mentioned in passing, and seems to be considered an ad hoc mechanism for coping with a few awkward cases. There is a lack of a general theory describing PN's, both from a syntactic and semantic angle.

In the area of Machine Translation, Wheeler [175] has also considered the PN problem. Much of the work in modifying the Systran system, bought by the EC in 1976, has involved dealing with unknown words, what Wheeler calls Not Found Words or NFWs. A standard approach in MT is to leave any NFW in the mother tongue, presumably partly from the assumption that it is a name of some sort. The new Systran is more advanced that this, making use of much morphological processing, both to derive word roots and to discover morphological correspondences between languages. Wheeler states that at present any NFW's (given the morphological processing and the size of the Systran lexicon — 70,000 single word entries and 35,000 multi-word entries) are most likely to be highly specialised technical words or PN's. This once again shows that however large the lexicon, PNs will remain a problem.

The Systran approach resembles those we have already discussed — if a NFW occurs next to a title word, another name, or some other capitalised noun or PN, leave the NFW in the source language. If all the words in a sentence are capitalised, it is assumed to be a title or heading so the system must ignore any inferences it would otherwise draw about PNs from the capitalisation. It also uses a heuristic that if any sentence has more than 50% of NFW's, it is left untranslated, as given the size of the lexicon and the facilities for NFW handling, such a high number of NFW's is not considered possible, and such a sentence must in fact be in another language, e. g. a report or publication title.

Such heuristics would seem to cope well with people's names and publications. The size of the lexicon enables Systran to utilise the risky heuristic of leaving a NFW next to a capitalised noun in its mother tongue. If any of the words in, say, 'National Union of Public Employees' were not known, they would be erroneously left in the mother tongue — however with a large lexicon this is unlikely. In addition, PN's present a less serious problem to MT than to TU, in that the output of MT is aimed at people. Thus the detection of every PN as a single unit is not vital, for if it is translated word for word, the reader can use their own understanding of the text to realise that those words are in fact a PN. Thus if the above example were not known as a PN, and were translated

| Category | Examples |
|---|---|
| Persons | Ronald Reagan |
| Titled names | President Reagan |
| Geographic | countries, cities, mountains, lakes, etc. |
| Geopolitical | Latin America |
| Places | Tour Eiffel, Lincoln Plaza |
| Organisations (non-English words) | Procter and Gamble |
| Organisations (English words) | Security Council |
| Creations | books, movies etc. |

Table 5.2: Choueka's Proper Name Classification

as 'Union Nationale des Employés Publics' a French reader could presumably work out this was a single entity. A TU system is processing text for summarisation or database creation, and thus must itself be able to realise that the above words constitute a single entity. Moreover, lacking an overall theory of PN formation Systran would not realise that something like 'the Nine Mile Point nuclear facility' should not be translated into 'Neuf Mile Endroit', as it is a place name.

In the area of corpus research, Choueka [38] has described a program for locating collocations in text. Collocations are groups of words that commonly occur together. The original impetus for such work was to derive collocations whose meaning was not derivable from the words involved in the collocation, and thus which would need separate handling in the dictionary, e. g. 'foreign minister'. But as one might expect the work also extracted a large number of PN's. In fact these can be considered as collocations, for the 'meaning' of 'Los Angeles' or 'Ronald Reagan' is not derivable from the meaning of each of the constituents. Choueka presented a brief classification of PN's which was hoped would cover most of the PN collocations extracted. This is shown in table 5.2.

Some of the language used in this classification is a little strange. By non-English words Choueka presumably means words that have no meaning beyond their naming function, i. e. proper nouns (as opposed to Proper Names). This is a useful distinction, as will be shown in chapter 6, but it is not as simple as that above. Many organisations have names comprised of normal words and naming words, e. g. 'Barclays Bank', 'Likud Central Committee'.

The program was run on 10M words of New York Times news wire, and produced 16,000 2 word collocations, 4,800 3 word collocations, 1,000 4 word collocations, 200 5 words, and 10 6 words. The program did nothing else but extract collocations, no information on the nature of the collocations was provided, as the results were aimed at human lexicographers rather than NLP systems. No statistics were provided on the number of PNs found, or how they fitted into the above categories. The program is thus a very cost effective way of producing large numbers of PN's, but with no information provided on them is of little use to an NLP system. In addition as letter case information is not used the collocations are not divided into common nouns and PN's, and would need a human to do so.

We have reviewed some of Amsler's work on PN's in section 5.2 above, but here we describe it in more detail, with particular reference to [10]. This paper discusses the problems in matching raw text to dictionary text, and particularly problems caused by compound words. The PN extraction program required to enable a comparison of news text PN's to those in the World Almanac and Book of Facts is described, and the problems it encountered highlighted. The use of capitalisation in detecting PN's is hampered by the fact that any word is capitalised when starting a sentence. In addition, as periods frequently occur after abbreviations any word followed by a period can also be a PN. Lastly PN collocations can contain function words such as 'and' and 'of' (e. g. 'American Telephone and Telegraph Company', 'United States of America'), but these can also join two separate PN's (e. g. 'France and Spain').

The poor results achieved in the comparison were described in section 5.2. Closer inspection of these showed that much of this problem was due to the hitherto unexpected occurrence of variant forms of PNs. One source may have an abbreviated form while the other uses the full form (US, United States), or one may provide a shorter form while the other uses the full form (Rocky Mountains, Rockies).

The conclusion drawn was that PNs have a semi-grammatical structure, possessing variant forms that appear to follow rules. They can be understood in part from stored lexical entries and in part from using the common noun meaning of what appear to be PNs. Thus when a human encounters a collocation like 'Grand Forks Energy Research Center' they can understand this having never met it before, as a single establishment, which is a centre for energy research located at a place called Grand Forks. The words 'Energy Research Center', although capitalised must be understood as common nouns. But the words 'Grand Forks' must lose their common noun meaning and be understood as a PN meaning a place. The existence of a common rule for the creation of 'building_PN's' presumably helps in this process. This rule has the form

Building_PN → Location + Description + Building_noun.

Thus this particular PN can be understood from the application of rules. Amsler maintains that some PN's, however, must be understood from stored lexical entries. This may be true for a minority of PN's, especially those which seek to be noted by violating standard formation rules, but the majority of PN's we have observed in news text follow the standard rules we have derived (and which we describe in the next chapter).

### 5.5.1 Directly Relevant Work

In this section we introduce other TU work that has directly focused on PN's. Description is purposefully brief here, as we will return to this work in chapter 11, where we compare and contrast it to our own approach to handling PN's.

McDonald [120, 121] is currently working on a partial parser for unrestricted text. This parser is being tested on a Wall Street Journal corpus of news text. McDonald's stated aims in the development of the parser were to make it as robust as possible, so that it could cope with text that exceeded its grammar, and contained words not in its lexicon. McDonald states that one of the most important goals for such a parser is that

' unknown proper names must be categorised from context'.

However none of the cited papers contain descriptions on how the parser achieves this. McDonald has stated (personal communication) that the similarity between his approach and that taken in the FUNES system is 'quite striking'.

Katoh et al. [99, 100] have described work on an English to Japanese news translation project at NHK Research Labs in Japan. The problems that PN's cause a system attempting to process free news text are clearly highlighted here. The authors say:

> 'As news requires such a large number of words to deal with changing world events, it is impossible to list all the proper nouns into a dictionary. Most proper nouns remain undefined. Undefined proper nouns, however, can be treated essentially as defined words if they are identified as compound words with words preceding or following them'.

The approach is similar to FUNES, attempting to identify PN's from the surrounding defining words. When identified (a process carried out after morphological analysis and before syntactic analysis) the PN's and their defining words are grouped together into single units to facilitate the subsequent parse. The analysis of PN's assumes additional importance in MT, as in Japanese different words are used for the same English word, dependent on the surrounding context. For example the word 'President' has four translations, depending on whether it is preceded by a place name, a council name, a company name or the word 'Chinese'. Unknown PN's can be acquired as in FUNES. However, various different forms are entered into the lexicon, for example the entire compound 'Exxon Corp President Lee Raymond' is acquired when encountered, as well as 'President Lee Raymond' and (presumably) 'Lee Raymond'.

The SCISOR system developed at GE [140] has already been mentioned in chapter 2. This was originally developed as a TU system in the domain of corporate takeovers. It has since developed into the GE NLToolset [94] a more domain independent TU system, capable of being ported to various domains. The realm of corporate finance and acquisitions contains a large number of PN's, mainly company names. The problems these present have been considered at length by the system developers (Rau and Jacobs), and are described in [137]. Rau states:

> 'One of the major problems in the accurate analysis of natural language is the presence of unknown words, especially names. While names account for a large percentage of the unknowns in a text, they can also be the most important piece of information in a text ...'

The paper points out that to attempt to construct a complete DB or lexicon of company names is simply not possible, due to the transient nature of companies. In a test on financial news text it was found that 8% of the text was comprised of unknown words. 4% of the words in the text were constituents of company names, over a quarter of which were unknown words. [1] The recognition of company names is considered critical for accurate topic analysis, and thus some means must be found to cope with new company names as they occur in the news text being processed. The paper presents methods very similar to those used by the FUNES system for detecting word strings that are company names, for matching these to existing entries, for creating variant name forms and for updating if unknown.

In-depth discussion of these systems are postponed until chapter 11. The major difference that can be noted is that they are all morphologically/syntactically oriented ap-

[1] These percentages seem rather small, this is because Rau is only counting single unknown words. For example if 'Olivetti' were unknown it would only be counted as one word, even if it occurred as 'Olivetti Electronic Corp', whereas really this unknown compound consists of three words. In addition, unknown PN's composed of entirely known words (such as 'National Development Council') are not detected at all.

proaches, and do not extend their PN analysis into the semantic stage, as does FUNES. Thus the definitions they produce are more limited.

The work discussed in this section suggests several things. Firstly it seems that the majority of the work which has addressed the PN problem has done so in a relatively ad hoc manner, lacking a sound theory of PN's to base itself on. The work of Carrol and Amsler indicates that there are rules at work in the formation of PN's, and in the derivation of their variant forms. To successfully deal with PN's in a computational context requires that these rules be formally described, and embedded in a language processing system. We must discover how often PN's occur, what are the main categories of PN used, how general or specific are the rules in applying to each category, how are PN's formed (syntax) and how do neighbouring words, and component words, contribute to the meaning of the PN (semantics). As PN's are relatively common in many areas of real text, a global theory would be of help to many different projects.

Having given a brief account of previous work dealing directly with PN's, the final section of this chapter will look at systems that have actually tried to process real world text, with a view to seeing how much of a problem PN's were.

## 5.6 Work in News Analysis and Text Processing

One of the main contentions of the thesis is that news text is extremely rich in PNs, and thus should give problems to systems seeking to process such text. To test the validity of this contention we must investigate previous (and current) work of this nature.

Until recently there has been little work undertaking text understanding from real world text. The main exception to this has been the previously discussed work of Schank and his associates at Yale. More recently in the US there has been a major DARPA-sponsored drive towards development and assessment of text understanding systems, culminating in the recent MUC-3 evaluation, [50], and continuing in the current TIPSTER and MUC-4 programs. In addition, there have been several isolated attempts at news text processing.

The first of these was the famous FRUMP [51], produced by Gerald Dejong. This departed from the previous Yale programs SAM and PAM in that it did not attempt a deep understanding of the text being processed. It still used a script-based approach, but these were Sketchy Scripts, carrying much less detail than the detailed scripts of SAM. Its application area was raw news wire text, which it skimmed in an attempt to work out which sketchy script the story best fitted. When it had decided upon a best candidate it continued processing in a style called prediction and substantiation — the script put forward expectations about the sort of information to look for and how it might occur, and a parser would scan the text for the relevant information. This information would vary depending on the script, e. g. for a kidnap scenario it would be target, kidnapper, location, demands etc.

The approach that FRUMP took meant that it did not have to be concerned about unknown PN's within the text. This was because the predictor would predict where the looked for concepts would be, and unless FRUMP found something to contradict this it would just take that bit of text. Nevertheless lack of vocabulary (of any sort) was cited as the main reason for FRUMPs failures.

The very successful CONSTRUE system of Hayes et al [78] is also able to overcome the problems presented by PN's. CONSTRUE is not a text understander, it is a text categoriser which scans news stories to decide which of various topics they describe, and routes the story off to the appropriate Reuters department. It uses a large number of painstakingly hand-crafted pattern-action rules, which essentially look for keywords in specific grammatical patterns, which are indicative of a particular story. For instance the word 'war(s)' when it is a noun and not preceded by the word 'ratings' is indicative of a war topic. Thus the system does not concern itself with any lexical items beyond the keywords contained in its patterns, and the problems PNs present from the point of view of being unknown do not arise. (As an aside, as has been pointed out by Rau [137] and Kuhns [104], PN's do themselves offer a good means of topic classification, e. g. determining the specific topic of a business story from knowledge of the industry a company operates in).

In CONSTRUE a final topic is decided upon by comparing the various topics hypothesised (from the occurrence of the patterns in the pattern-action rules applying to those topics), and choosing that topic above a certain threshold. Then a confirmation search is made to check that the language used in hypothesizing the topic was not used in a misleading way. Such an approach has produced very impressive results, but is not usable for anything beyond topic determination. No attempt is made to understand the stories being processed. (Hayes points out the aim of the system is to do a specified task faster and better than a human, to this end the system is highly successful).

A different, and more NLP-like approach to news story topic classification is described in [104, 105] by Kuhns. The NAS (News Analysis System) parses each story using a Government-Binding parser, and matches the parsed story to a concept rule base. This produces a general topic area, and more specific details, e. g.

```
Terrorism,
          location     : Kuwait,
          instrument   : Bombings.
```

As it is based upon a syntactic parse the system must be able to determine the word class of every word it encounters. To this end it has a lexicon of around 15,000 words. As Kuhns states, despite having many common names in the lexicon :

> 'Names, especially of individuals, corporations, and geographical locations, not present in the lexicon are found in news reports regularly'.

Thus a facility is needed to cope with unknowns. The only feature mentioned is the use of suffixes like 'inc', 'corp' or 'co' to classify the preceding words as a corporation. This is the same approach we have seen in Gershman and Allport. It should also be pointed out that the frequent use of such suffixes after company names is a practice peculiar to American papers. In the absence of such suffixes the unknown is simply classified as a noun. This process works if the system can assume that all other word classes (such as adjectives, verbs etc) used in news text are contained in the lexicon.

Such an approach is a very effective error-handling procedure. However, if a system is capable of analysing a text to a sufficient level to produce some kind of semantic representation, it seems wasteful not to make greater use of the surrounding information to create a detailed description of any PNs encountered, especially as this information is conveyed through language in just the same way as the rest of the test. As Kuhns claims that ideally NAS should make use of specialised databases of company names, a facility for automatically acquiring such names from text would seem of use. In addition, any accompanying information about the companies' sphere of operation would also help in text classification. This is stated clearly by Kuhns who gives several examples of the specific detail indicators of a story being derived from knowledge of the industry of the companies involved.

The work of the GE team has been referred to already in this chapter. Further consideration of the problems which PN's have presented to the GE TU system is given in [95, 93]. This presents work on the creation of lexico-semantic patterns for aiding text understanding. Some of these patterns are specifically for dealing with PN's. These are considered difficult to process, even when they are in the lexicon, as they can be very long, contain preceding and following descriptive information, and appositive information. The GE system has specific patterns for recognising human names and place names. Unfortunately these patterns are not described, mention is simply made of the use of looking for combinations of spatial prepositions and known locations for detecting location patterns and isolating these in a pre-processing stage. The conclusion of [137] mentions similar mechanisms, including the use of titles and large lists of known first names in the detection of personal PN's. Although no deep description is given, the fact that one of the most developed news text understanding systems has given so much consideration to PN's sits very nicely with our contention stated at the start of the chapter, that it is only when one starts to process real text that the PN problem becomes apparent. When it does become apparent it is a major obstacle to accurate text understanding.

Until very recently the above was the only available work on text understanding systems that processed news text. With the publication of the MUC-3 Conference proceedings [50] many descriptions of systems that had gone unreported have become available. The reason for the lack of reporting appears to be that the majority of these have been developed within commercial companies as potential products. The MUC-3 conference describes the performance and nature of the 15 TU systems that took part in the MUC-3 evaluation. This was a task involving the processing of 100 news stories describing terrorist activities in Latin America. The systems were required to fill templates for all stories describing recent terrorist attacks on civilians. These templates consisted of slots for such things as type of incident, perpetrator, target, location and date. A 1300 story development corpus was made available to all the participating sites some months before the final test. This corpus could be used for lexicon and Knowledge base building and grammar and semantic rule development.

One of the major problems with the MUC-3 task in assessing the abilities of systems to process raw unseen text was the use of the development corpus in constructing domain specific lexicons. Thus all systems were able to extract all the PN's (and any other vocabulary) from the DC prior to the test run. The percentage of words in the test run that were not in the DC was only 1.6%. From the studies we have reviewed on the coverage of MRD's this can be seen to be an unrealistic degree of lexical coverage. In addition large lists of PN's were made available to all the participants. Given that the task was aimed at assessing the language analysis abilities of the systems this may have seemed advisable, but it does prevent us gaining an accurate picture of the problems unknown words, and especially PN's, can present to TU systems.

Despite the low level of unknown words in the test stories, most of the participants equipped their systems with facilities for coping with unknown words. This was probably due to the fact that the participants were mostly existing TU systems, which were intended to be useful across a variety of domains, and thus an ability to deal with unknown words was crucial.

The typical approach taken in handling unknown words was :

1. use spell checker to determine if unknowns are really mis-spelled known words. If

this fails.

2. use specialist module for detecting Spanish names (not all systems). If this fails
3. use general morphological processing to either reveal known root, or to guess at unknown's part of speech. If still fail
4. classify as noun

We shall look briefly at some of the systems that explicitly mention the handling of unknowns or PN's. Each system report being fairly brief the information given was not detailed.

The PLUM (Probabilistic Language Understanding Mechanism) of BBN (Bolt Beranek and Newman) incorporated a statistically-derived 5 letter model of words of Spanish origin. This was first run on the list of unknown words produced by applying a part of speech tagger to the DC, and all words classified as Spanish origin were added to the lexicon as names. This special module was deemed necessary because the tagger used performed poorly on all upper case input. The BBN part of speech tagger (described at length in [123] ) was not ready at the time of the evaluation, but has subsequently been able to achieve 85% accuracy on classifying unknown words. The tagger that was used, and the Spanish name checker, pre-processed the test input and classified any unknown words as to part of speech using morphological information.

The ITP Interprettext system mentions the use of special facilities for recognising and building up PN's, but does not describe these at all. They do point out that the failure of these facilities was one of the major reasons for their poor performance. This once again shows the importance of effective PN handling procedures.

The DBG system of Language Systems Inc. places much emphasis on two modules for dealing with unexpected (i. e. unknown) words. The LUX module detects such words, and attempts spell correction. If this fails to identify a known word, the unknown can either be sent to a user for definition, or this can be done automatically using morphology. The system also has a Template Unexpected Input Module. This searches for expected information that is missing in the final output — the expectations come from the unfilled slots in the template used to represent the current event. The search is conducted among material that could not be analysed during the main analysis. For example if the perpetrator slot of the template is unfilled, leftover material will be searched for anything that could possibly be a terrorist group name. This module thus provides a mechanism for recognising unknown PN's and specifying their function.

The PRC Praktus system follows the methodology outlined above. Like BBN it includes specialist routines for detecting Spanish names, although these are much simpler than the BBN module. The system includes 24 heuristics for dealing with unknown words whose roots are also unknown. One such heuristic classifies an unknown word ending in 'z' or 'o' as a Spanish name. If all these heuristics fail, use is made of syntactic and semantic context during parsing. This is not described, although given the similarities of the overall system to those produced at Yale, it seems likely it makes use of selectional restrictions and case filler expectations.

The TACITUS system of SRI also follows the above methods, likewise including a facility for handling Hispanic names. The morphological handling routines are very simple, and very similar to the ones used in the FUNES system. Unknowns ending in 'ly' are classified as adverbs, those ending in 'ing' or 'ed' as verbs, all other unknowns are classified as nouns. As these heuristics are so simple they will obviously produce erroneous misclassifications. However as the authors clearly show, in the majority of cases these can be judged harmless as they still permit a parse to be obtained. A particular case that has also been noted in the FUNES system is that any adjectives that are classified as nouns will simply be handled as prenominal noun complements. (However without a large lexicon the automatic classification of unknowns ending in 'ing' or 'ed' as verbs would surely lead to problems in that many adjectives also have this ending, and such a misclassification is likely to disrupt a parser. As the TACITUS lexicon has 12,000 entries, including 2,000 personal names, this problem would be largely avoided, in that most common verbs and adjectives would be covered).

In summary, virtually all the systems described in the MUC-3 proceedings stress the importance of handling unknowns. Even though they are equipped beforehand with the majority of PN's that occur in the test corpus many use special PN handling procedures for dealing with locations, human names and terrorist groups. It appears that PN's are held to be a considerable problem even if they are already known. None of the systems, though, seems to possess a sound theory of the syntax and semantics of PN's, in the way they do for normal words. In view of the fact that names are vital to the MUC-3 task, this is a large handicap. This view has been supported by Dave Lewis, one of the MUC-3 program committee, who observed (personal communication) that PN's were one of the major stumbling-blocks for systems in MUC-3. The major limiting factor felt by all the MUC participants was time, so this perhaps explains the ad hoc nature of some of the name handling. Another problem affecting the handling of PN's in the task was the fact that the text was all upper case. This obviously makes detection of PN's more difficult (although the fact participating systems were given the majority of PN's should have compensated for this). It will be interesting to observe the performance of the systems participating in MUC-4, in June 1992.

## 5.7 Summary and Conclusion

This chapter has described work on PN's in all areas of Computational Linguistics. Discussion of work with MRD's and Text Understanding systems has highlighted two important problems connected with PN's. The first is their poor coverage in conventional dictionaries. The second is their high occurrence in such areas as news text. The combination of these two factors means that PN's present great problems for TU systems. Moreover, this is not a problem that can be cured by the addition of large PN lexicons. Firstly, these will never be adequate, and the variant form problem described by Amsler will hamper their efficiency even if they are. Secondly, as was shown in the MUC-3 evaluation where PN lexical resources were adequate, the complexity of PN structure and that of the descriptive material that accompanies them, means that they are difficult to analyse even if they are known.

All these factors would not cause such problems if PN's were unimportant in text understanding. However, far from being unimportant, PN's are one of the most important elements in a piece of text. Effective analysis of these elements is held to be crucial both in text classification and in text understanding.

The little work that there is looking at the structure of PN's has indicated that their effective processing requires knowledge of the semi-grammatical rules that they follow.

We can summarize all these points into three major issues, which together form the motivation for the present investigation:

- The shortage of dictionary resources on PN's, and the infeasibility of a static source ever being able to give complete coverage.

- The problems PN's have given to news TU systems, and the importance of their effective handling for understanding of the overall text.
- The lack of theoretical work on the subject of the computational processing of PN's.

We have developed an approach, implemented in the FUNES system, which seeks to address the PN problem by automatically acquiring PN's as it processes the text they occur in. This thesis claims that as PN's make up a large and important part of news text they merit being one of the major focuses of processing in an NLP system. Whereas previous systems that have realised the problem PN's represent have merely dealt with them as a problem or a sideline, the FUNES system makes their processing central to its analysis. As we stress over and over, the description of PN's makes up a large part of the very text being analysed. Rather than just detecting this and ignoring it, FUNES attempts (as do humans) to process this text as it processes all the text of a story, and extract the descriptive information from it. Such an approach enables the system to increase its lexical coverage as it runs, automatically updating its lexicon with the PN's (and other words) it learns.

In the next chapter we present a description and classification of the major categories of PN that are encountered in news text. This description underpins our entire approach.

# Chapter 6

# The Grammatical Nature of Proper Names and their Surrounding Context

## 6.1 Introduction

In this chapter we discuss the grammatical nature of PN's and their accompanying descriptive information. The discussion is from a purely grammatical viewpoint. In subsequent chapters we show how an NLP system can handle the various phenomena described here. First we describe the taxonomy of PN's that has been created for categorisation of PN's in the news text domain. Then we describe the structure of each of the most common categories of PN observed in this domain. This description is aided by the introduction of a model of the syntactic patterns in which PN's occur. Each pattern has a corresponding semantics, which describes the information conveyed about the PN in a frame/slot formalism.

The description of PN's contained in this chapter is based on the in-depth examination of news text undertaken during the thesis. We have tried to base the analysis solely on examples found in this examination. Indeed, the analysis has only been made possible by the wealth of examples found in the text studied. However, in some situations the number of examples of a particular type has been extremely limited. In such situations, rather than attempt no analysis, we have made recourse to well known examples from other sources (history and reference books). As the emphasis in this work has been on a data-driven analysis, the fullest analysis is given to those types which occur most frequently in the text examined.

The news text we have studied over the past two years has been comprised of:

- The UK dailies The Times, The Daily Telegraph, The Independent, The Guardian and the Financial Times, and their Sunday equivalents.

- A computational corpus of US news from the AP newswire, comprising home and foreign news, and sports and arts news. This corpus comprises about a quarter of a million words.

- A computational corpus of AP newswire news connected with Turkey. This comprises about 50,000 words.

- The ACL/DCI CD-ROM [1] of Wall Street Journal text. This contains (as far as

I can ascertain) a large part of the text of the Wall Street Journal for the years 1987,88 and 89.

The on-line material has mainly been used along with text processing utilities for locating particular lines of interest.

## 6.2 A Taxonomy of the Common Categories of Proper Name

A common first step in any description of PN's has been the creation of a taxonomy reflecting real world distinctions in the nature of the referent of the PN. Some of the taxonomies created in linguistic accounts of PN's were described in the last chapter. This seems a sensible approach, as we can expect to find differences between each group, and commonalities within each group. However, in providing descriptions of PN's, we also find great consistency across groups in the methods used to describe them. For the purposes of computational modelling the taxonomy should also reflect commonalities in the processing of each category. As the present work is based on news text, it is interesting to consider any taxonomy in use by journalists. Just such a system is presented in the Chicago Manual of Style [166], a reference book for American journalistic and editing style. This describes the following major groups of PN (with considerable sub-divisions within each group):

— Personal Names (incl. titles and offices)
— Nationalities and Groups of People (e. g. Oriental, Italian, Aborigine)
— Place Names
— Words derived from Proper Names (e. g. philistine, vulcanize, french fries)
— Names of Organisations
— Historical and Cultural Terms (such as events, treaties, legal cases and cultural movements)
— Calendar and Time Designations
— Religious Names and Terms (incl. deities, religious groups and writings)
— Military Terms (incl. military awards, battles, and forces)
— Ships, Aircraft and Spacecraft
— Scientific Terminology (incl. names of plants and animals, medical, chemical, and geological terms)
— Trademarks
— Titles of works (incl. books, periodicals, television and film)

The nature of a taxonomy obviously depends on its purpose, and the above is aimed at producing consistent typography in newspapers. The groupings seem to be made on both the grounds of logical consistency, and typographical consistency. As we are considering news text it may seem logical to follow strictly the taxonomy given here, in that the ways of describing and representing PN's should vary in accordance with these guidelines. However this does not appear to be so, for a variety of reasons. Firstly the Chicago Manual of Style is not used by all journalists. Secondly it provides guidelines rather than strict rules. Thirdly much of the text that would be analysed from a news wire is yet to be edited into its final and correct form. Aside from this, much of the taxonomy is more specific than is needed for our purposes, or else is not suitable for our purposes. Religious groups, for instance, can be described in much the same way as normal organisations, but here they appear in a different group. Treaties, which are above included in Historical and Cultural Terms, are in fact very similar in form to some organisation names. Most

| Category | Description |
|---|---|
| Personal Names | Names for animate beings |
| Place Names | Countries, buildings, geographical features, towns and regions |
| Corporation Names | All types of group, including businesses, governments, pressure groups, charities etc |
| Legislation Names | Acts, treaties, bills, charters etc |
| Information Source Names | Printed and electronic media, including newspapers, films, and tv programs |
| Event Names | Wars, Revolutions, Disasters etc |
| Object Names | Vehicles, military hardware, computers etc |
| Origin Names | Proper Adjectives ('Spanish') and Proper Names ('Italian') |

Table 6.1: Classes of Proper Names

calendar and time designations can be considered to occur as entries in the lexicon, and need no additional analysis beyond this. Finally some of the categories, and their sub-divisions, occur either not at all or very infrequently in standard news text (e. g. scientific terminology).

Given these considerations, and the fact that in this thesis we are focusing on PN's in news text, we have produced an alternative taxonomy, which is shown in Table 6.1.

All of these categories exhibit particular properties of their own, which justify their separation (in addition to the logical separation implied by the differences in their real world referents). Although the same syntactic constructions can be used to describe any of the above groups, there is enough variation between them to make a category by category description a necessity.

We desire the description of PN's presented to be as widely useful as possible. Two key criteria are felt necessary for this — the description should not be tied to a particular implementation, and it should be as un-ambiguous as possible. It is felt that a formal approach will fulfill these criteria, and hence permit the reconstruction of this work by other researchers. Therefore the linguistic description is accompanied by a model of the syntax and semantics of PN's and their surrounding context. In figure 6.1 we define the syntactic and semantic domains. Our model maps entities from the syntactic domain to the semantic domain.

The Genus of a PN [1] is the category it is assigned to, based on the syntactic context in which it occurs. The Genus categories outlined in figure 6.1 are the only ones that exist in our model. The Differentia terms are slot labels, which, together with their fillers, describe more fine-grained features of each PN within a Genus category. The differentia slots are filled by items from the lexical domain, or by further differentia slots and fillers, e. g. 'role(president,property(former))'. The sort of 'definition' we aim to produce with this model (and its implementation in FUNES) would be called an 'intensional' definition in the field of terminology.

The operator **sem** serves to map syntactic categories to semantic cases. Cases consist of a functor from the set of differentia terms and an argument (or arguments) from the set of lexical entries, e. g. 'works_for(government)', or 'field(trade,industry)'. It takes a

[1] Some may question the suitability of assigning a PN to a Genus category through use of a supertype/isa link. This is because PN's are individual entities in the world, and cannot be 'defined' in the same way as 'tables' or 'presidents' (see [61]). They may thus be construed as assertional knowledge rather than definitional type knowledge. An alternative approach would be to have entries for stereotypical PN's (towns, roles, corps etc) in the KB, and then assert all PN entries as instances of these entries, with some or all of the possible slots filled.

SYNTACTIC DOMAIN

LEXICAL ENTRIES, identified by <>

SYNTACTIC CATEGORIES, such as ADJ, NP, and PP. Normal brackets () indicates their contents are optional. An asterisk indicates arbitrary repetition of the starred constituent.

SYNTACTIC/SEMANTIC CATEGORES FOR KEYWORDS. These are defined as the set:
{ROLE_KW, LOC_KW, CORP_KW, LEGIS_KW, EVENT_KW, ISOURCE_KW, OBJECT_KW}
(In FUNES, each of these is defined as a noun with semantic category role, loc, corp etc).

SEMANTIC DOMAIN

GENUS TERMS: these are defined as the set:
{[human,name], [loc,name], [corp,name], [legis,name], [event,name], [isource,name], [object,name]}

DIFFERENTIA TERMS: these are defined as the set:
{supertype, role, origin, works_for, location, distance, direction, superpart, abbrev, fullname, namehead, composed_of, run_by, field, made_by, property, related_to, staff, product, assoc, boss_of}

GENDER CATEGORIES: these are defined as the set:
{n, m, f, b, unknown} where n=neuter, m=male, f=female, and e=either (i.e. male or female)

CONNECTIVES, defined as the set:
{^, / } where ^ = and, / = or

Figure 6.1: A Formal Model of the Syntax and Semantics of Proper Names

syntactic category as its input and supplies a semantic case (or set of cases) as its output. If the input is empty so is the output.

Each of the syntactic constructions which accompany a PN can be pictured as implying a particular semantics for that PN. We think of the elements in the syntactic domain as revealing 'Name Describing Formulas' (NDF's). Each of these formulas has its own semantics, which are used to build a semantic representation of the PN. The term NDF is chosen to show the similarity of this approach to work in the MRD field, where particular syntactic patterns in dictionary definitions are identified as revealing particular semantic qualities about the word being defined. In particular, Liddy [109] has coined the term 'Relation Revealing Formula' for mappings from particular syntactic constructions to specific semantic relations such as EXPERIENCER or PERFORMER. Each NDF serves to 'partially define' the PN in question, i. e. it places it in a taxonomy of genus terms, and provides differentia information in the form of further descriptive information. [2]

To illustrate the use of the model we will present a brief example. The following sections will use more extensive examples to illustrate the syntax and semantics of each Genus category. Consider the NP 'Congolese interim Prime Minister Andre Milongo'. This is described by the syntactic pattern 'Adj* Role_Keyword Capitalised_noun* '. This pattern reveals a ROLE NDF, which dictates that any capitalised nouns following the Role_Keyword be defined as a human name, with their role slot filled by the Role_Keyword, origin slot by the origin term, and property slot by any other adjective. Thus we map this syntactic pattern into the semantic element:

```
[noun([andre, milongo],
        Genus:[human,name],
        gender:Unknown,
        role:prime minister
```

[2] It might be claimed that we have only outlined the structure (or syntax) of the semantic domain, and that we have not given the terms within it any clear meaning. This is a criticism that can be (and has been) levelled at all semantic representations of natural language terms. The essence of the complaint is that semantic network or frame/slot formalisms merely link meaningless symbols with other meaningless symbols, and that this is a circular representation in which there is no semantic grounding, i. e. there is no final level which is not defined in more basic terms. This criticism however can be levelled at all language, and at the dictionaries that describe language. The problem is, how do we describe words unless it be in terms of other words. Some (e. g. Schank [148] and Jackendoff [91]) have attempted to produce descriptions in terms of 'primitives', but this merely moves the whole problem one stage lower (although Jackendoff has attempted to ground his primitives in some sort of conceptual system). In addition these approaches only really apply to verbs, they do not solve the problem of defining objects or types. The famous game example of Wittgenstein showed that it does not seem possible to produce a set of defining criteria for such entities. Some NLP workers have attempted to solve the problem by 'grounding' (see Harnad [75]) their systems in an external reality, using a perceptual (visual) representation of objects as the final level in a semantic network ([66, 157]. Although this might be feasible for such things as tables and chairs (and even here, as prototype theorists [143, 91] have argued, there are problems), it is not so for people and places.

To a large extent however such wrangling is unnecessary. A computer does not need a human-like knowledge of language, it needs a sufficient semantics to enable it to achieve a specified task. For an NLP system this will be syntactic and semantic analysis of a sentence. For this, the grouping of words into widely-recognised syntactic and semantic categories is all that is needed. This will permit a system to complete its parse and produce a semantic representation for a sentence. Grouping into Genus categories like those outlined above will permit case filling operations (e. g. if something is classed as 'loc name' it is a valid filler for an AT_LOC case) and, at a higher level, template filling and data extraction (e. g. if a lexical item is classed as 'corp name' it can be a candidate for filling the PERPETRATOR slot in a terrorist event). The semantic representation above is also of a suitable form to aid in the automatic construction of printed PN lexicons.

Thus, instead of pursuing an 'absolute' semantics we can look for an 'ends-driven' semantics. It is considered that the above description is sufficient for purposes of NL analysis and data extraction.

| Term | Abbreviation | Meaning |
|---|---|---|
| Proper Name | PN | The name of a real or fictional entity. Can be composed of all or any of the following components. |
| Proper Noun | Pnoun | A capitalised single noun that has no meaning beyond its function as a name, e. g. Michael, Paris. |
| Capitalised Constituent | CC | A capitalised word of any syntactic category (but usually a noun) that does have a normal meaning, and retains this meaning when it occurs as part of a PN e. g. **Society** for the **Abolition** of **Vivisection** |
| Ambiguous Noun | AmbN | A capitalised word of any syntactic category (but usually a noun), that does have a normal meaning but loses this meaning when capitalised and used as part of a PN, becoming in effect a Pnoun e. g. Bush, Baker, Thatcher. |
| Proper Name Constituent | PNcon | Any constituent of a PN, i. e. a Pnoun, CC or AmbN |

Table 6.2: Terms used in the description of Proper Names

origin:congo
property:interim ]

The process by which this mapping is achieved in the FUNES system is described in the following chapters. In the following sections of this chapter we describe each category of PN, both linguistically and formally. To facilitate understanding, in table 6.2 we describe the terms used throughout this and subsequent chapters, and the abbreviations used for them.

## 6.3 Personal Names

The category of Personal Names consists of names for particular humans or animals, living or dead, real or fictional. Personal names can, in theory, be composed of virtually any name or non-name constituent or group of constituents. However in news text certain forms are more common than others. We begin the description of personal names by looking at the structure of the name itself, before moving on to consider the nature of the words that surround it.

### 6.3.1 The Nature of Personal Names

Personal Names, like all PN's, are identified in English by their initial capital(s). Common formats for personal names include:

1) $\geq 1$ Pnoun, e. g. John Prescott, Kinnock, John Gordon Sinclair
2) $\geq 0$ Pnoun + 1 AmbN, e. g. George Bush, Bush
3) $\geq 0$ Pnoun + $\geq 1$ Initial + $\geq 1$ PNcon, e. g. G. K. Chesterton, P. Phillips, Franklin D. Roosevelt.
4) $\geq 0$ Pnoun + link constituent + $\geq 1$ PNcon, e. g. Daphne du Maurier, Max Von Castle, de Gaulle, Zia ul-Haq, Ahmad Hussein al-Khodair.
5) $\geq 1$ Title/Keyword + $\geq 1$ PNcon, e. g. Dr. Livingstone, Chancellor Helmut Kohl, Mrs Thatcher.

As shown in some of the above examples, foreign names (non-English) can include various linking constituents like 'van', 'du' 'ibn', and 'y'. Sometimes these are capitalised, and sometimes not. Their inclusion in the lexicon enables them to act as indicators of personal names. Personal names can also include apostrophes, as in 'O'Connor' or 'd'Abuission'.

For the most part personal names stand out from normal lexical items both by being capitalised and containing constituents that have no meaning beyond their name. Thus even if they are unknown we can hope to distinguish them in the text. However, while plainly distinguishing them from non-name words this does not always distinguish them from other categories of PN. Companies can frequently be named after their founders, and objects of all sorts can be given personal names. Therefore often this distinction depends on either the name being so well known no ambiguity is possible or else the occurrence of dis-ambiguating material (dubbed the Within Text Description, or WTD) occurring alongside the PN. Such material also serves to provide a precise description of the name it accompanies. This material is just as important in the analysis of PN's as the nature of the PN itself, and we now turn away from the structure of the personal name to examine the nature of the surrounding context which so often describes it.

### 6.3.2 The Nature of the Personal Name Within Text Description

Dis-ambiguating information on the nature of the name is provided in the form of a Keyword (KW). A KW is a common noun that serves to place a PN within one of the categories described in the FUNES taxonomy. Each category of PN has a corresponding KW of the same category. For personal names this can occur in two places, either preceding the PN or following it. The KW can be in an appositive NP [3] or it can occur in the same NP as the PN itself. The KW used to describe a personal name is called a ROLE KW. This is a noun that describes the role that a person performs in the world, e. g. plumber, president, nurse or scientist. Such words frequently describe an occupation, as that is usually how people are characterised in news text. But they can also describe temporary states (e. g. hostage), or more permanent attributes such as wife or brother.

Titles such as 'Ms' and 'Sir' can also be characterised as key words in that they place the following PN into a particular category, even though they are not strictly nouns (and are handled somewhat differently in FUNES). Personal Names can thus be described in any of the following ways, where X indicates a personal PN :

1. **X ⟨comma⟩ KW_NP (PP) ⟨comma⟩** e. g. Boris Yeltsin, president of the Russian Federation,

2. **KW_NP (PP) ⟨comma⟩ X ⟨comma⟩** e. g. The President of the Russian Federation, Boris Yeltsin,

3. **KW_NP X (PP)** e. g. Russian Federation President Boris Yeltsin

4. **X ⟨comma⟩ Age_P ⟨comma⟩** e. g. Peter Wilson, 21,

[3] Throughout this thesis we use the terms 'apposition' and 'appositive' in a fairly general way, following Gershman [68]: ' appositives are used in English to further specify the meaning of the noun they follow'. We would extend this description as a descriptive appositive NP can precede the PN it describes. Quirk et al [136] present a considerably more detailed description of apposition, including full and partial, strict and weak, and restrictive and non-restrictive apposition. In their terminology, the sort of apposition we shall describe is strict, non-restrictive apposition. The semantic relationship between the apposed NP's is one of equivalence and attribution. The construction which we refer to as 'KW PN' (e. g. President Bush) is described by them as strict restrictive apposition, a term which also includes constructions such as 'the port of Dubrovnik'.

The KW_NP, whether it occurs in apposition or not, has the following form:

KW_NP → (Det) (Adj*) (Noun_comp*) Role_KW

The fourth pattern which serves to give a person's age has the form:

Age_P → Digit*/ age(d) Digit* / Digit* years old

These syntactic patterns can be read as saying that any noun(s) that occur in position X, and are capitalised, comprise a personal PN whose semantics are given by a Role Name Describing Formula. Thus they can be used both to identify personal PN's in text, and to produce a description of that PN, composed of genus and differentia information. The first three syntactic patterns shown above reveal the same Role NDF, which we call ROLE NDF1. Thus each pattern is simply a variant way of conveying the same meaning.

The semantics of ROLE NDF1 are:

PN Genus = [human,name]
PN role = KW
PN gender = KW gender
PN differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem(PP)

The fourth syntactic patterns reveals ROLE NDF2, the much simpler semantics for which are shown below:

PN Genus = [human,name]
PN age = Digit*

In the FUNES system, the **sem** operator is embodied in the semantic analysis stage of processing. As explained above, **sem** takes as input a syntactic category, and supplies as output a semantic case. The case label can be used to provide a corresponding slot in the 'name frame' being built for the PN under analysis. So, if we consider the NP 'British Prime Minister John Major' one of the inputs to **sem** would be the Adjective 'British'. **Sem** would output 'origin(British)', leading to the creation of an origin slot in the frame describing 'John Major'. The differentia slots are filled from the semantic analysis of the adjectives, noun complements and PP's. The 'name-frame' produced for the above NP would be:

PN Genus = [human,name]
PN role = [prime minister]
PN gender = Unknown
PN origin = british

The most common cases output by **sem** are shown in table 6.3. Each of these contributes a differentia slot in the name-frame representing the PN they describe.

When the KW_NP is used appositively, PP's may be appended to it. This is not permitted when the PN follows the KW_NP directly. However the PN itself may be followed by a PP. In this case the two most common cases are Works_for and Origin, e. g. 'President Carlos Menem of Argentine' or 'Mark James of Kleinwort Benson'.

These are the most common ways in which personal names are described within the news text that has been examined. Of course personal names are not always described, and may just occur alone. [4] In addition there are various other constructions that may serve to provide information. These are:

[4] The issue of how many PN's occur with and without a description of some kind is an important one, and will be discussed in the presentation of results in chapter 11.

1) Origin. This can be given by a PP or by an Adjective:
e. g. **British** Prime Minister John Major
e. g. Prince Victor Emmanuel **of Savoy**
2) Property. This can be given by an adjective (or possibly a noun comp):
e. g. Mircea Snegur, the **former** Communist ...
3) Field. This can be given by a PP or a noun comp:
e. g. the current **Industry** Minister, Org Marais,
e. g. Louis Terrenoire, minister **for information** under President de Gaulle
4) Works_for. This can be given by a PP or a noun comp:
e. g. **UN** Secretary General Boutros Boutros Ghali
e. g. Derek Kys, chairman of the giant **General Mining Union Corporation,**
5) Related_to. This is given by a PP:
e. g. Rita Marley, the mother **of Bob Marley,**
6) Associate. This is given by a PP:
e. g. Mr Nicholas Biwott, a colleague **of Daniel arap Moi,**
7) Product. This is given by a PP:
e. g. The author **of Les Filles du Calvaire,** Pierre Combescot

Table 6.3: Differentia Information for Personal Proper Names

- Direct Definition, e. g. 'Carlos Fuentes is the most influential author in Mexico', 'John Smith has become leader of the Labour Party'.

- A Relative Clause following the PN, e. g. 'John Smith, who has become leader of the Labour Party'.

- A Relative Clause following the Appositive NP, e. g. 'Aung San Suu Kyi, the opposition leader who is married to a British university lecturer,'.

- A PP following the PN, e. g. John Bennet, of 53 Wiltshire Gardens, Bridlington,'.

The first three cases are examples of PN's being described through a VP. Except where the verb is 'be/become', this topic has not been intensely investigated. We discuss the subject further in chapter 11, when we consider extensions to the FUNES system. Some cases are already handled, as described in chapter 8. The first two cases can be viewed as expanded appositives, or more correctly appositives can be viewed as contracted copular sentences. We rarely see constructions of the form 'Daryl Gates is the Los Angeles Chief of Police. He is under pressure to resign' , it is far more common to see this descriptive information presented via an appositive, i. e. 'Daryl Gates, Los Angeles Chief of Police, is under pressure to resign'. In the third case, the relative clause has been mostly found to give non-definitional information, referring to actions on the part of the PN. The last case was briefly mentioned above. Post-PN PP's are usually used as in the above example, to give people's addresses, or to give their employer, or for political figures, their nationality.

Another phenomenon that is true of personal names, as with many others, is that they can occur in more than one form. By this we simply mean that people can be referred to by their full name or their surname or some other combination of name and initials.

This presents problems for the reader of the text, whether human or computer, who must realise that the same individual is being referred to in each case. Thus we can have 'F. D. Roosevelt', 'Franklin D. Roosevelt', 'President Roosevelt', 'Franklin Delano Roosevelt', and so on. In chapter 8 we show how FUNES copes with ambiguous references of this sort.

This completes the account of simple cases of personal name description. We now turn to discuss more complex descriptions involving conjunction.

### 6.3.3 Personal Name Within Text Descriptions involving Conjunction

The most complex area in the construction of PN's and descriptive material is that of conjunction. This involves the analysis of plural KW's and ellipsis. All of the patterns described above can conjoin in various ways. Firstly we can have a simple conjunction of any of the patterns, e. g. 'President Mitterand and President Bush'. These are simple cases of NP conjunction. More complex is the possibility of a conjunction of descriptions applying to the same personal name, or a conjunction of personal names being described by the same description. More complex still is a conjunction of PN's and descriptions where one description applies to one PN, the other description to the other PN. Here we will not go into the cases of simple NP conjunction, as this is a straightforward repetition of each pattern. We will look at conjunction within each of the three patterns described on page 77.

The appositive patterns can ascribe a single description to two individuals, e. g. 'Marcel Proust and James Joyce, arguably the most significant authors of the 20th century'. Conversely, we can have a single individual and a conjoined description, e. g. 'Chairman and chief executive, Steve Scott,'. Description provided by a preceding_KW does not exhibit this duality, the only possibility is a plural KW describing two individuals, e. g. 'Presidents Mitterand and Bush'. Where we have a single individual and two descriptions then the NDF applies both descriptions to the one person; where we have a single description and two individuals, the NDF applies the description to both individuals. We have chosen not to show the syntactic patterns and NDF's here so as to prevent the reader becoming lost in a morass of detail. They are all contained in appendix N.

The most complex construction involves conjunction of both the PN and the descriptive NP. This can involve ellipsis of part of the second KW_NP. Some examples are shown below:

1) Peter Marks and Andrew Winstanley, the president and treasurer,
2) Peter Marks and Andrew Winstanley, the president and treasurer of Amoco corp,
3) Peter Marks and Andrew Winstanley, the Amoco president and treasurer,
4) Paddy Ashdown and John Smith, the leaders of the Liberal Democrat and (of the) Labour parties (respectively),

The formation of an NDF for this type of construction is complicated by the possible ellipsis of crucial sections of the conjoined descriptive NP, and by ambiguity in the application of adjectives and noun complements to their corresponding PN's. Where only one of the KW_NP's has accompanying noun complements or PP's then we can reliably assume they apply to both the personal names, as in examples 2 and 3. Where there is an accompanying adjective on the first KW it may apply to both, as in 'former president and treasurer', or just the KW it directly precedes. A construction like the last is too complex to describe at a superficial level. Such constructions have been found very rarely, presumably because of the need for clarity in most news text. The conjoined NP or PP must be correctly attached to the initial NP or PP, and the appropriate descriptive elements applied to the appropriate personal names. The small number of examples observed of this type prevent a deeper analysis. We have stressed many times the data-driven nature of the work, and this is reflected in the depth of description given to various WTD's. Those which have been observed most frequently are described in most detail, while the rare cases receive only superficial coverage.

This completes the description of personal name PN's. Despite the complexity of some of the syntactic patterns which we have shown, the vast majority of cases are described by either a single KW_NP, or a simple appositive NP.

In the next section we move on to look at the nature of place PN's.

## 6.4 Place Names

Place names refer to all kinds of geographic entity: countries, towns, rivers, streets, seas, buildings, regions etc. Within this section we shall not dwell much on the use of apposition. Although this can be used to describe any place PN, it is not used to the same degree as for personal PN's. This is because place names frequently contain descriptive elements as part of the name, i. e. the name is self-descriptive, making the use of an appositive NP less necessary. This makes division of discussion on the name and its description less useful than for people. However we will still commence by looking briefly at the nature of the place name.

### 6.4.1 The Nature of Place Names

This resembles to some extent the structure of personal names, the main difference being the fact that initials are not used with place names. In addition, the KW can here both precede and follow the PN while being within the same Noun Group. In many cases it is actually incorporated into the PN. Place names commonly take the following forms:

1) $\geq$ 1 Pnoun, e. g. London, Wogga Wogga, Hong Kong.
2) $\geq$ 1 CC, e. g. Three Bridges, The Cape of Good Hope
3) A combination of Pnouns and CC's, e. g. River Thames, Abbey Wood, Hyde Park.

Obviously many place names that occur in news text are foreign (as with personal names), and in this case it may be that the name does actually mean something in its mother language. While this may make it in some ways an ambiguous noun, we still treat it as a Pnoun, as in English it really has no meaning beyond its function as a name.

Keywords incorporated into the name can frequently provide a description of the place name they help form. However due to historical events this may be misleading, e. g. Abbey Wood is a suburb of London and not a wood as such at all. We may occasionally have a contradiction between a descriptive element in the name, and one used outside the name, e. g. 'the small town of Red Bridge'. This is not something that has been investigated. [5]

[5] Such examples are (or should be) of great interest in the long running debate on the meaning of PN's. The fact that 'Abbey Wood', while appearing to have meaning in the same way as a common name or definite description, is actually meaningless offers support for the conjecture that PN's only refer by arbitrary association.

### 6.4.2 The Nature of the Place Name Within Text Description

As we said above, the place name and its description are much more interwoven than is the case for personal names. This partly accounts for the lack of appositive description. Another reason for this is that the large place names (such as countries and major cities) seem to be considered so well known that further description is not necessary. Thus we never see examples like 'Germany, a large central European Democracy'. It is strange that however well known people are they invariably receive some kind of description upon their first occurrence in a news story. Thus we frequently see cases of 'President Bush', or 'John Major, the Prime Minister'. This is perhaps an effect of the greater transience of individuals on the world scene. A third reason for the relative lack of apposition is the preference for the construction 'the town of X', or 'the central Soviet Republic of X', as opposed to 'X, a central Soviet Republic'.

Below we show the general syntactic patterns for place name WTD's:

1. **X ⟨comma⟩ KW_NP (Dir_NP) (PP*) ⟨comma⟩**, e. g. 'Listica, a predominantly Croatian region in Bosnia',

2. **(Det) KW X,** e. g. Lake Michigan, the River Thames

3. **(Det) X KW,** e. g. Trafalgar Square, the Rocky Mountains

4. **KW_NP ⟨of⟩ X** e. g. the city of Quebec.

5. Directional Types.

The KW_NP has the form 'Det (Adj*) Loc_KW'. A Loc_KW (location KW) is a common noun of semantic category 'loc', e. g. town, river, square, region. Directional words (such as 'north' and 'south') can also be regarded as Loc KW's in some ways, in that they permit classification of the following word (if it is capitalised) as a place name, but they lack the differentia information of the normal KW's. If we hear about 'the River Nile' then we know it is a place name, and a river, but if we hear about 'North Korea', although we know it is a place name we do not infer it is a type of north. (Directional words can also occur followed by 'of', as in 'north of Florence', again indicating a place name, but here they are not strictly functioning as KW's as they are not in the same NP as the PN).

As with personal names described by KW's, place names described by patterns 2–4 can receive additional description in an appositive.

The first four syntactic patterns reveal PLACE NDF1, the semantics for which are shown below:

PN Genus = [loc,name]
PN supertype = KW
PN gender = n
PN differentia = sem(Adj*) $\wedge$ sem(PP) $\wedge$ sem(Dir_NP)

Adjectives and PP's most frequently serve to give superpart information, so we could have 'Horsham, a small town in Sussex' or 'Copenhagen, the Danish capital'. Adjectives can also give general property information, e. g. 'the Lebanese **border** town of Qabrikha'. The following PP's and/or Directional_NP can also give the specific location of the place name. Thus we could see 'in Basra, 20 miles from the Iranian boarder,'. (We describe the structure of the Dir_NP below when we discuss directional types).

Perhaps the most interesting phenomenon concerning place names is the question of Keyword Attachment. This is what Carroll [30] investigated as the question of namehead formation. This issue only arises when we have a place name formed from a Pnoun and a KW. The question of attachment depends on various factors. Some of these were mentioned by Carroll, as described in chapter 5. He observed that the attachment appears to vary depending on the category of the KW, and he presented a few examples of this. In addition he observed that the location of the KW in relation to the PN also has some influence. It appears that where the KW precedes the Pnoun the attachment is always optional. The class of names in which this is the case is quite small, comprising planets, mountains (single ones), rivers and lakes. For each of these the name can occur with or without its KW.

When the KW follows the Pnoun, then the category of the KW must be taken into account when deciding if it should attach or not. Whilst in general language there appear to be fairly relaxed rules governing attachment, in the sublanguage of news text the rules are tighter. This again can be put down to the need for clarity in describing news events. It is also connected with the two concepts of familiarity and uniqueness. As many authorities have pointed out one of the main functions of a PN is to provide a listener/reader with a unique referent. Many following KW's will allow themselves to be dropped from the name, providing the remaining element provides a unique referent. Thus if we look at the KW 'ocean', this permits foreshortening with 'Atlantic' and 'Pacific' as they are Pnouns which provide a unique referent. However the words 'Indian' and 'Arctic' are not. While they are Pnouns, they do not provide unique referents. We find a similar situation with regard to seas, where the first element is a CC. With 'Yellow Sea' and 'Black Sea' we cannot foreshorten, whereas with the Pnoun 'Mediterranean', which provides a unique referent, we can. We can expect Pnouns to be more likely to provide unique referents than CC's. The other major factor to consider is context/familiarity. In certain contexts, where one's listener is known to be very familiar with the concepts under discussion, foreshortening may occur where in other contexts it could not. We can see this with museum names. Normally these cannot foreshorten, but one can imagine situations in which they could, e. g. 'Of all the museums in London, the Natural History is my favourite'. It might be thought that the foreshortening phenomenon is entirely controlled by the necessity of providing an unique referent, but this does not seem to be the case. For instance there is only one Trafalgar Square in London, but it is never referred to as 'Trafalgar'.

What we appear to have is a situation in which certain classes of KW will accept foreshortening, provided a unique referent is provided, and other classes which will not accept foreshortening at all. As news text aims for clarity, and cannot assume all readers to be familiar with the same thing, foreshortening is not as wide-spread as it may be in conversation (for example).

As we will see in subsequent chapters, information can be included in the lexical entry of a KW to aid in this attachment decision.

Buildings are included in the class of Place PN's. While certain classes of building PN's are justifiable lexical entries (such as museums, palaces, etc), in that they are unique entities, there are other classes of building whose lexical entry is more questionable. A notable case is where we have a place PN followed by a building KW e. g. 'Liverpool Cathedral', 'Doncaster Station'. It is debatable to what degree such constructs can be considered PN's (this dimension of nameness could be associated with the philosophical debate on names vs definite descriptions). Even if they are, it is also debatable whether they should be listed in a lexicon. As most towns in the Western world have some sort of station (and town hall, police station and so on) such inclusion would soon lead to a lexical explosion. It is awkward to say at what point we start disallowing entry to the lexicon. This point will be discussed further in subsequent chapters. There also exists a class where the PN names the owner of the following KW, as in 'an Olivetti factory.' The presence of the indefinite article clarifies the situation. The article can also help distinguish a unique entity (i. e. a PN) from a member of a group, as in 'a Manhattan hotel' and 'The Manhattan Hotel'.

Lastly in this section we consider the directional patterns alluded to above. -These do not directly place the PN within a supertype through use of a KW, rather the fact that a place is being talked about can be inferred from the fact a location for it is given. The syntactic patterns for directional constructions are shown below:

1. **X ⟨comma⟩ Dir_NP ⟨comma⟩**, e. g. In Crawley, 30 miles to the south of London,

2. **X ⟨comma⟩ (PP*) ⟨comma⟩** e. g. 'by Majdel Silm, at the western tip of Israel's self-declared security zone'

3. **X ⟨comma⟩ X ⟨comma⟩** e. g. Paris,Texas,

The syntax of the Dir_NP is given in appendix N. In pattern 3 directional information is given by a superpart PN. This pattern is clearly signalled by 2 PN's in apposition, neither having a determiner. [6]

Syntactic patterns 1 and 2 reveal PLACE NDF 2, pattern 3 reveals PLACE NDF 3, the semantics for which are shown below:

PLACE NDF2 :
PN Genus = [loc,name]
PN location = sem(Dir_NP)/sem(PP*)
PN gender = n

PLACE NDF3:
PN Genus = [loc,name]
PN superpart = PN2
PN Gen = n

This concludes the account of simple place name descriptions. We now turn briefly to look at the use of conjunction in place name patterns.

### 6.4.3 Place Name Within Text Descriptions involving Conjunction

As with personal names we can have a simple conjunction of any of the above patterns. It is also theoretically possible to have a conjunction of KW_NP's, used so as to describe a single place name, although this has not been observed. Any examples we might attempt to construct sound rather strange, e. g. 'London, a city of 7 million people and the capital of England,'. The commonest cases that we have actually observed are shown below:

1. **X ⟨and⟩ X ⟨comma⟩ plural_KW_NP (PP) ⟨comma⟩** e. g. Hanover and Munich, two beautiful old towns in Germany,

2. **Plural_KW X ⟨and⟩ X** e. g. Lakes Michigan and Ontario

3. **(Det) X ⟨and⟩ X plural_KW** e. g. the Hudson and Mississippi rivers

4. **Plural_KW_NP ⟨of⟩ X ⟨and⟩ X** e. g. the cities of Cork and Dublin

The nature of the NDF's for these do not differ in substance from the NDF's for the singular cases, and so will not be shown here (they are included in appendix N).

[6] Although some place names do indeed have determiners (e. g. the US, the UK) they would not be used in such a construction. We do not talk about 'London, the UK,' but instead say 'London, England'.

| | |
|---|---|
| 1 | ≥ 1 Pnoun, e. g. Amstrad,Rolls Royce. |
| 2 | ≥ 1 Pnoun plus a corp KW, e. g. Polly Peck International. |
| 3 | ≥ 1 Pnoun and its field or product (a CC), e. g. Allens Catering, Lipton Tea. |
| 4 | ≥ 2 Initials, e. g. NATO, EEC. |
| 5 | Initials or PNcons linked by an ampersand, e. g. M&S, White & Mackay. |
| 6 | A name comprised entirely of CC's in a noungroup, with a corp KW as headnoun e. g. the National Welfare Council |
| 7 | A name comprised entirely of CC's involving PP's e. g. the National Society for the Protection of Cruelty to Children |
| 8 | Exceptions. This is for various exemplars which do not clearly fit into any of the above categories, e. g. names made up of CC's with no KW (Help the Aged, American Watch). |

Table 6.4: Corporation Proper Name Forms

## 6.5 Corporation Names

The category of Corporation names (Corp PN's) includes all groups of people, such as political, charitable, and pressure groups, and all companies/corporations. As with place names, or perhaps to an even greater degree, with corp PN's we find an intermingling of the name and its description. This intermingling presumably stems in part from the advantages for a company in having its field of operation be immediately apparent from its name. Corp names also show interesting properties in their ability to foreshorten. They also present problems in the nature of their conjunction patterns, as it is possible for single names to contain a conjunction. In fact corp names are probably the most complex of all the categories of name we shall describe.

### 6.5.1 The Nature of Corp Names

Corp PN's have the most varied forms of all the categories we describe. The different possibilities are shown in Table 6.4. As with place names we are less likely to see an appositive description when the name itself contains a description (although this is not impossible). In UK news it is quite common to have company names like 'Amstrad' and 'Olivetti', sometimes on their own but usually with a following apposition NP, whereas in US news it is very rare to see a name without a following KW such as 'corp', 'ind' or 'inc'. When we have a corp PN composed of two Pnouns (such as 'Price Waterhouse', 'Rolls Royce'), the basic form of the name is indistinguishable from a personal name. Thus, unless the name is extremely well known, we will usually find an appositive description to dis-ambiguate the name. Sometimes verb number can do this, but it is common to see corps used with both a singular and a plural verb (e. g. 'Price Waterhouse has announced', 'Price Waterhouse have announced').

It should be noted from the above examples that the corp KW is a very important constituent in the corp PN.

### 6.5.2 The Nature of the Corp Name Within Text Description

The syntactic patterns for corp WTD's can be grouped into two categories, depending on whether the corp KW forms part of the corp PN. In all of the patterns the KW_NP has the form 'Det (Adj*) (Noun_comp*) Corp_KW'. The first group of patterns, revealing CORP NDF1, is shown below:

1. **X ⟨comma⟩ KW_NP (PP) ⟨comma⟩**, e. g. Banca della Swizzera Italiana, the sixth biggest bank in Switzerland,
2. **KW_NP (PP) ⟨comma⟩ X ⟨comma⟩**, e. g. the Spanish opposition parliamentary coalition, United Left,
3. **KW_NP X**, e. g. the building society Abbey National

The semantics of CORP NDF1 are shown below:

PN Genus = [corp,name]
PN supertype = KW
PN gender = n
PN differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem(PP*)

In all of these cases the corp PN does not contain a corp KW.

The second set of patterns, in which the PN does contain a KW, is :

4. **KW_NP** e. g. The Black Socialist Party
5. **Pnoun (Noun_comp*) KW** e. g. Barclays Bank, Vogelstruitbult Metal Holdings
6. **KW_NP PP*** e. g. Economic Community of West African States

These patterns reveal CORP NDF2, whose semantics are:

PN Genus = [corp,name]
PN gender = n
PN differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem(PP*)

As can be seen, the structure of the PN varies in the above patterns. In pattern 4 the corp PN and the KW_NP are the same thing, the PN forms its own WTD. For pattern 5 the corp PN can exist in two forms, both with and without the KW (this is similar to some place PN's, e. g. 'Sahara' and 'Sahara Desert'). So we can talk about 'Barclays' and 'Barclays Bank' or 'Vogelstruitbult' and 'Vogelstruitbult Metal Holdings'. Pattern 6 is the same as pattern 4 in that the corp PN is composed entirely from its descriptive components.

The difference between CORP NDF's 1 and 2 stems from the way the corp PN is formed, and from the absence of supertype information in NDF 2. Where the corp PN is basically composed of Pnouns, which have no real meaning, we need to preserve the supertype information carried in the appositive or the preceding KW. However where the PN itself carries the KW, the supertype slot is superfluous, as its information is carried by the PN.

The nature of the differentia information revealed by **sem** is more complex than that found with place PN's, resembling more the relatively varied information found with personal PN's. The **sem** operator is used in exactly the same way, to derive semantic cases which are used as slots in the PN name-frame. The most common information given is shown in table 6.5

As shown in the above examples when a corp PN contains following PP's, these may indicate a variety of information. When the PP gives origin information it is sometimes not possible to tell if it is strictly part of the name or not. There would be ambiguity in example 6 if the 's form were not used, i. e. if we had 'the Pakistan People's Party of Benazir Bhutto' it would be hard to tell if this were the full name, or if the 'of Benazir Bhutto' were just added to give additional information. This would explain why the 's

1) Origin. This can be given by an adjective, a noun comp or a PP:
e. g. Alleghany, a **New York-based** insurance concern,
e. g. **Pakistan** Muslim League Party
e. g. Evangelical Church **of Germany**
2) Property. This can be given by an adjective (or noun comp):
e. g. the **right-wing** Freedom Party
3) Superpart. This can be given by a noun comp or PP:
e. g. the **UN** Food and Agriculture Organisation
e. g. Hill Samuel Bank, **TSB's** merchant banking arm,
4) Field. This can be given by a noun comp or PP:
e. g. National **Rivers** Authority
e. g. Joint Council **for the Welfare of Immigrants**
5) Composed_of. This can be given by a noun comp or PP:
e. g. the **Boilermakers** Union
e. g. National Union **of Mineworkers,**
6) Run_by. This can be given by a PP:
e. g. **Ms Bhutto's** opposition Pakistan People's Party

Table 6.5: Differentia Information for Corporation Proper Names

form is used in such cases. The most common prepositions that form part of a corp name are 'for', 'of', and 'on'. As we will show in chapter 9, not all PP's that follow a corp name will form part of the name. In fact to be considered part of the name the initial PP must be of case 'Field', 'Origin' or 'Composed_of'. We have observed one or two examples of names with a 'to PP' where the PP indicates purpose, e. g. 'Committee to Preserve American Color Televisions'. Were a significant number of such cases to be observed they could be included in this group, but at present the number of cases is too few for this.

The Exception class mentioned at the start of the section contains corp PN's that are composed of CC's, but with no corp or product KW. We make this an exception class as so few exemplars have been observed, in comparison to the large number of cases which follow the 'standard' rules we have produced for corp name formation. Examples are 'Age Concern', 'Help the Aged' and 'American Watch'. There is the possibility of a new group, formed from a 'people group' noun plus a PP, e. g. 'Young Americans for Freedom', 'Women against Violence against Women'.

A Corp PN may also be indicated through its preceding a role, i_source or object KW. In this situation, instead of having a direct WTD, the context in which the corp PN occurs implies that it is indeed a corp. Such a context is of use in inferring that an unknown Pnoun is a corp PN. So if we read about 'the Ferranti chairman' it is very likely that 'Ferranti' is a corp PN. Similarly if the Pnoun is followed by an isource or object KW then it is likely that it is a corp Pnoun. So we might have 'a Harris poll' or 'an Amstrad computer'. In this case there is some ambiguity in that sometimes the Pnoun may actually be naming the isource or object, e. g. 'the Herald newspaper' or 'a Chieftain tank'. The determiner is of some help in differentiating isources from corps — we would never see a newspaper called the Herald described as 'a Herald newspaper'. However this does not help in all cases, as we might see a poll carried out by Harris described as 'the

Harris poll'. It seems that we can differentiate isource KW's, which indicate an isource PN, from isource nouns, which indicate a corp PN. We talk more about these problems in the sections on isources and objects as they present many difficulties. The syntactic pattern for this context is:

- (Det) PNcon* role_KW/isource_noun/object_KW (PN) e. g. a Reuters journalist, Olivetti chairman Carlo de Benedetti, a Universal film, the new Cray supercomputer.

Of course, any sort of corp PN can occur in this context, but it is of most interest for identifying those corps which lack KW's, as in these cases it is the only source of information we have.

Appositives can be used to give an initial form of the full corp name, or, if the corp is most widely known by its acronym, the appositive may give the full name. More usually though such information is given in brackets. As examples we could have 'The National Funeral Directors Union, the NFDU,' or 'NATO, the North Atlantic Treaty Organisation'.

One problematic issue in the description of corp names is whether nationality adjectives should always be considered part of the name. This is a similar problem to the question of building names that are preceded by place PN's. Some corps names that involve a nationality adjective are truly unique entries, deserving of lexical entry, e. g. 'British Nuclear Fuels'. However others are simply entities that are possessed by many, if not all, countries, e. g. 'the Australian Labour Party', 'the French Army'. As with place KW's we have found that corp KW's can be grouped depending on the nature of the preceding Pnoun they are likely to take. The word 'army' for instance is most likely to be preceded by a nationality Pnoun, whereas the word 'corporation' is most likely to be preceded by a name Pnoun. The case of the KW can also help decide on the 'nameness' of the compound, if it is not capitalised then the chance of the compound being a true corp PN are very small. Nationality can also be given in a following PP, in which case this constituent can sometimes be present and sometimes absent. In this case we really have to consider both possibilities as PN's, e. g. 'Ford of Britain' and 'Ford'.

Finally we must consider the problem of corp name variant forms. These differ somewhat from place PN variants, in that with a place name either variant is likely to occur whenever we encounter it in a story. Thus we are just as likely to read about 'Everest' as 'Mount Everest'. Corp variants resemble personal name variants more, in that the typical pattern is to first give the full name, and on subsequent occurrences use a foreshortened form. So, as we might first read about 'Neil Kinnock, the former Labour leader', and subsequently just see 'Mr. Kinnock', we also first see 'Capcom Investment Services', and subsequently just 'Capcom'.

In the nature of their variation there emerges a clear difference between those PN's composed entirely of CC's, such as 'Central Intelligence Agency' or 'Industrial Chemicals Incorporated', and those which have a Pnoun component plus CC's, such as 'Barclays Investment Group' or 'Polly Peck International.' The latter have shortened forms consisting of the Pnoun on its own, whilst the former foreshorten into acronyms, usually formed from their initials. The heuristics observed for foreshortening of corp names are shown in table 6.6. As we show in chapter 9, FUNES can use knowledge of these heuristics to match the shortened name to the longer.

This concludes the analysis of simple corp names/WTD's. Next we consider cases involving conjunction, a class in which corp names present many interesting cases.

Pnoun* of Loc → Pnoun, e. g. Ford of Britain → Ford
Pnoun1 + Pnoun* → Pnoun1, e. g. Asahi Juken → Asahi
Pnoun1-Pnoun → Pnoun1, e. g. Thomson-CSF → Thomson
Pnoun1-Pnoun → Initial_Pnoun1 Initial_Pnoun, e. g. Hewlett-Packard → HP
Pnoun1 & Pnoun → Pnoun1, e. g. Saatchi & Saatchi → Saatchi
Pnoun1 & Pnoun → Initial_Pnoun1 & Initial_Pnoun, e. g. Marks & Spenser → M & S
Pnoun + (Field etc words) + KW → Pnoun, e. g. Invergordon Distillers Group → Invergordon
CC* → Initials, e. g. Accounting Standards Board → ASB

Table 6.6: Heuristics for Corporation Proper Name Variant Forms

### 6.5.3 Corp Name Within Text Descriptions involving Conjunction

As well as exhibiting all the problems with conjunction that we have encountered with personal and place names, corp names present additional complication in that the name itself can involve a conjunction (e. g. 'The New Jersey Sports and Exposition Authority'). Therefore the number of the KW becomes very important in determining the nature of the corp PN or PN's.

The NDF's for conjoined corp names are the same as those for singular corp names except that all the rules apply to 2 PN's, or that two groups of information are applied to the same PN. They are shown in appendix N. Here we show some of the syntactic patterns to demonstrate the complications that arise when a corp PN actually contains a conjunction, as opposed to being conjoined by one.

With description via an appositive, we can have a plural KW_NP describing 2 corp PN's. As with place names it is less common to see a conjunction of KW_NP's describing the same corp PN. The syntactic patterns are shown below:

1. **X ⟨and⟩ X ⟨comma⟩ plural_KW_NP (PP) ⟨comma⟩** e. g. ICI and Hanson Trust, the top two UK companies,
2. **Plural_KW_NP (PP) ⟨comma⟩ X ⟨and⟩ X ⟨comma⟩** e. g. the two largest banks in Switzerland, BKI and SBC,

As corps can contain conjunctions we can also have a single corp PN which contains a conjunction and is described by an appositive, e. g. 'Lane and Lane, the building group'.

Pattern 3 (page 86) shows a similar variation. We can have a conjunction of two corps, or a single corp involving a conjunction:

a) **plural KW_NP X ⟨and⟩ X**
e. g. the building societies Abbey National and Halifax
b) **KW_NP Pnoun* ⟨and⟩ Pnoun***
e. g. the building society Bradford and Bingley

As pattern 5 is a simple Pnoun/KW reversal of pattern 3 the same conjunction variation applies, so we can have 'the Abbey National and Halifax building societies', and 'the Bradford and Bingley building society'.

Pattern 4 can involve conjunction in 2 ways. Firstly (a rare case) we can have 2 PN's conjoined with ellipsis, giving the pattern:

**(Det) (Adj1*) Noun_comp1* ⟨and⟩ (Adj2*) Noun_comp2* plural_KW**
e. g. The Black Socialist and Tory Asian parties.

But we can also have a single name of this kind which has 'and' within it, e. g. 'The Licensed Brewers and Victuallers Association'. This has the pattern:

**Det (Adj*) Noun_comp* ⟨and⟩ (Adj*) Noun_comp* single_KW**

Finally pattern 6 shows a similar picture to pattern 4. It is possible (although uncommon) to have a conjunction of two different corps, e. g. 'The Societies for Abolition of Vivisection and Legislation on Medical Experimentation'. More frequent are single corps of this type which contain a conjunction in their name, e. g. 'The Association of Futures Brokers and Dealers', or 'The National Association for the Care and Resettlement of Offenders'. The patterns for these are in appendix N.

For the most part complex interminglings of corp PN's are avoided, as they become very hard to follow. However corp PN's which involve a conjunction are fairly common.

The three categories of PN that have been described are by far the most common found in news text. It is precisely their high frequency of occurrence, and thus the many examples we have been able to examine, that have permitted the detailed analysis performed. The next four classes are less common, and thus we have not been able to analyse them to the same level of detail. Nevertheless they all possess interesting properties.

## 6.6 Information Source Names

The class of Information Source or Isource PN's includes all forms of printed and electronic information, such as newspapers, magazines, radio and television programmes, films and books. It is a class which shows, perhaps more than any other, the data-driven nature of this thesis. To fully analyse the whole class would almost be a Phd in itself, including all the esoteric film and book titles, and the even more bizarre publications output by unheard of government offices. Fortunately we have established the boundaries of this work as including only those PN's which are common in news text. This excludes a large part of the Isource group. So we do not describe such titles as 'Aeronautic Regional Sub-committee triennial finance and accounts report', or even 'Journal of Pharmacology'.

The most common information source PN found in news text is that of newspapers themselves. It is these which we concentrate on. English newspapers and magazines are commonly composed of 1 or 2 CC's (e. g. 'The Independent', 'The Sunday People', 'Private Eye'). They can also involve a following PP (e. g. 'The Mail on Sunday'). The definite determiner is very common but by no means strictly necessary. The nature of foreign journals is more difficult to describe, as they are in a foreign language, and so will differ from case to case. As the main English papers appear to be assumed to be known, they are not often described. This is not the case for foreign papers which are invariably given a WTD. Given the lack of description accompanying English papers the surrounding context must be used in a more indirect way to identify the PN. We have identified a variety of common contexts for English newspaper names, which are shown in appendix N. These are useful in handling unknown names, but given the general lack of description, we decided to enter the most common daily paper names into the lexicon in FUNES.

The picture for foreign newspapers and the majority of magazines is quite different. Firstly, the huge number of these prevents any attempt at full lexical coverage (as does the appearance and disappearance of papers in more unstable countries). Secondly, given their novelty to the British reader, they are invariably accompanied by a WTD. The syntactic patterns established are:

1. **X ⟨comma⟩ Det (Adj*) (Noun_comp*) i_sourceKW ⟨comma⟩**, e. g. Hurriyet, another Istanbul newspaper,

2. **Det (Adj*) (Noun_comp*) i_sourceKW X** e. g. the Saudi daily Ashaar al-Awsat, the Democratic Union newspaper Duma, the building journal Tradetalk.

3. **X i_sourceKW**, e. g. the Herald newspaper, The People's Daily paper, Cosmopolitan magazine.

Isource KW's include paper, newspaper, daily, magazine, journal, and weekly. The first three are by far the most common. These patterns give rise to Isource NDF2 (Isource NDF1 for English papers is shown in appendix N):

PN Genus = [isource,name]
PN supertype = KW
PN Gen = n
PN differentia = sem(Adj*) $\wedge$ sem(Noun_comp*)

The semantic analysis of the adjective usually gives rise to origin, or political orientation information. The analysis of the Noun_comp gives origin, publisher or other associated corp information.

The WTD revealed by pattern 3 (i. e. where the KW follows the PNcon) may give rise to the problem of isource/corp differentiation, as was mentioned in the section on corps. This refers to the fact that the preceding PNcon may name the owning or publishing corp or it may name the paper. When the KW follows the PNcon and it is capitalised, this problem does not arise as it clearly forms part of the PN e. g. 'Morning Herald', 'Barking Chronicle'. However, when the KW is not capitalised we must consider its precise type. The problem can then be overcome by realising that the class of isource nouns can be divided into KW's and non-KW's. It is only the KW's which indicate isource PN's, as it is only these type of isources (e. g. newspapers and magazines) that are commonly named. The non-KW's, such as 'poll', 'article' and 'survey', are not commonly named, and so do not indicate, or form part of, isource PN's. It is only the non-KW's which indicate a preceding PNcon is a corp (as in 'a MORI opinion poll', 'the CBI survey...'). However even with an isource KW there is a situation when a preceding PNcon may name a corp. This occurs when there is an appositive following the KW which names the isource in question. If this is the case it implies that the initial PNcon is the publisher or owning corporation. If we were to imagine the above example 'The Herald newspaper' as being 'The Herald newspaper, Kulgung,' then we would infer 'Kulgung' to be the name of a paper owned or published by the Herald group.

An additional problem here is the category of PN in which we should place papers. In different contexts they can be CORPS (as in 'he works for the Times' ), ISOURCES ('an article in the Times') or OBJECTS ('my watch is under the Times'). If the PN in question is unknown it will be derived as a different category in each context. This is as it should be. We include them under i_source, as that is the most common context in which they are used. (The use of i_source as CORP is really metonymical.)

The WTD patterns can be conjoined, in similar ways to those we have met already. Thus we can have 'the popular French dailies Le Monde and Figaro', or 'The Sydney Herald and National Enquirer newspapers'. The syntactic patterns and corresponding semantics are shown in appendix N.

Other sorts of i_source PN do occur in news text, the most common being those describing television and radio programmes and films. The few examples of these that have been observed mostly use appositives. However in US news it is common to give a television programme title following its network company, e. g. 'ABC's American Bandstand', 'NBC-TV's Today program'. UK news does also use a KW pattern, as in 'The BBC television programme Panorama', or 'the Kevin Costner movie The Bodyguard'.

## 6.7 Legislation Names

Like the last category, the dearth of examples in news text has prevented a detailed analysis of this group. There again appears to be a difference between UK usage and US usage, with UK news always capitalising these PN's, and US news not always doing so. According to the Chicago Manual of Style, legislation that is pending should not be capitalised. The commonest KW's observed are Act, Bill, Treaty, Charter, Constitution, Declaration and Concord.

In structure these correspond very closely to certain corp name structures. The three types of Legis PN observed are:

1. **Det (Adj*) (Noun_Comp*) LegisKW**, e. g. The Emergency Powers Act, the Education Reform Bill.

2. **Det (Adj*) Pnoun (Noun_comp*) LegisKW**, e. g. The Sherman Act, The Hawley-Smoot Tariff Act

3. **Det (Adj*) LegisKW PP***, e. g. The Treaty of Rome

Any of these may be described by an appositive although this is not very likely (and has not been observed), as the sort of description one might expect to give about a piece of legislation would be more likely to occur as a VP or entire sentence. Type 1 carries its own WTD, and is therefore less likely to be so described. The NDF revealed by the above patterns is very simple:

PN Genus = [legis,name]
PN supertype = KW
PN Gen = n
PN differentia = sem(Noun_comp*) $\wedge$ sem(PP)

The most common information provided by the PP or Noun comp is origin and field (as in Treaty **of Utrecht**, Bill **of Rights**, **Community Care Act**, and **British North America** Act).

The conjunction patterns for legis PN's show even more clearly their similarity to corp PN's. They exhibit the same duality of two conjoined PN's vs. a single PN containing a conjunction. The only pattern in which this has been commonly observed is the first, so we can have the examples 'The Education Reform and Local Government Bills', and 'The Competition and Service Bill'.

Patterns 2 and 3 could exhibit similar variation, although I have not observed any examples. We could imagine 'The Sherman and Roosevelt Acts' or 'The Sherman and Roosevelt Act'. For pattern 3 although one could imagine a single act name containing a conjunct none has been observed. It is certainly possible to link two PN's of this pattern though, e. g. 'The Treaties of Rome and (of) Versailles'. The patterns for all these types are contained in appendix N.

## 6.8 Event Names

These also are relatively rare. Human history being what it is, they also invariably refer to, or describe, happenings of some unpleasantness, such as wars and revolutions. The two types observed are:

1. **Det (Adj*) (Noun_comp*) EventKW**, e. g. The Cultural Revolution, the Second World War.

2. **Det (Adj*) (Noun_comp*) EventKW PP***, e. g. The Retreat from Moscow, The Siege of Leningrad.

As ever, it is feasible to provide further info through an appositive but unlikely. For the most part these PN's are signalled by event KW's, but there are a few exceptions such as 'the South Sea Bubble'. There are also exceptional cases like 'WW2' or 'World War II'.

The sketchy EVENT NDF formula revealed by the above patterns has the following semantics:

PN Genus = [event,name]
PN supertype = KW
PN Gen = n

Given the sparsity of examples we have not attempted a description of any differentia information. From the few examples we have seen all we can say is that it is common for the location connected with the event to be given.

As regards conjunction, no cases of a single event PN involving a conjunction have been observed, but it is quite possible to have conjoined events, as in 'the Yom Kippur and Vietnam wars', 'the French and Russian Revolutions'. The syntax and semantics for this are shown in appendix N. These cases cause a problem in that the conjoined element violates some of the conditions for normal event PN's. Firstly the determiner is absent as it is ellided, and secondly the KW can be lower case, whereas with a single PN it is always capitalised. Although this does not matter in a descriptive sense, as we just have a different pattern for the conjoined case, it causes trouble in the computational analysis, as will be described in chapter 10.

## 6.9 Object Names

The last class we shall look at, that of Object PN's, is a varied group presenting many problems for analysis. These problems are increased by the lack of examples observed in the news text studied. This means that to present a complete analysis has involved considering additional examples, both fictional and historical. The main groups described in this class are vehicles (cars, planes, boats), military hardware (missiles, tanks, guns etc) and electronic machinery. These are the objects most commonly described in news text. They can take the following forms:

1. (Det) + $\geq$ 1 Pnoun, e. g. Mary Rose, the Bismark. Apart from the determiner this type is identical to a certain class of personal PN's presumably because objects are frequently named after people.

2. (Det) + $\geq$ 1 CC, e. g. the Ark Royal, a Sea King (helicopter), the Spirit of St Louis.

3. Combination of Pnoun and Object KW, e. g. the Battleship Missouri, the tanker Caledonia

4. Pnoun or CC plus digit, e. g. The Queen Elizabeth II, Saturn V,

5. Serial number plus object KW **or** Maker + Serial Number, e. g. AK-11 assault rifle, the new Honda MZ-750.

Like Isource PN's, Object PN's can be preceded by a determiner. When the PN is preceded by a KW the determiner may be retained, or it may be dropped. The decision to retain or remove the determiner depends on the effect it has on the sentence's readability. For example, the phrase 'the cruiser The Achille Lauro' sounds awkward, so the determiner would be dropped to give 'the cruiser Achille Lauro'. As we shall see in chapter 10, if the determiner is not dropped it causes difficulties in parsing the Noun Group.

One of the main problems in the analysis of object KW's is the fact that an object KW can also be a component of a corp PN. In addition the PNcon that accompanies the KW can be awkward to classify, as it may refer to a corp, a person, or the object itself.

The common syntactic patterns used for describing object PN's are:

1. **(Det) X ⟨comma⟩ KW_NP (PP) ⟨comma⟩**, e. g. The Mary Rose, the flagship of Henry VIII,

2. **KW_NP (PP) ⟨comma⟩ (Det) X ⟨comma⟩**, e. g. The largest aircraft carrier in the US navy, the Michigan,

3. **KW_NP X**, e. g. the Greek freighter Skoplos

4. **Det (Adj*) X KW**, e. g. the 856 foot Ozman Azmi tanker

5. **Det Pnoun* Serial no. (KW)** , e. g. the Amstrad AD-50, the Pershing II rocket,

6. **Corp_PN X**, e. g. the Ford Sierra, a Rolls Royce Silver Shadow

The KW_NP has form '(Det) (Adj*) (Noun_comp*) Object_KW'. The first three patterns have a clear semantics, revealing Object NDF1:

PN MSC = [object,name]
PN supertype= KW
PN Gen = n
PN differentia = sem(Adj*) $\wedge$ sem(Noun_comp*) $\wedge$ sem (PP)

(The fourth pattern can reveal various NDF's, as we shall discuss below).

As ever, a name containing a KW can still be described by an appositive. The combination of an apposition NP and a PN composed purely of Pnouns provides the clearest pattern for describing an object, as there are no problems in determining the relationship of the Pnouns to the object KW in the appositive. Definition by KW in the same Noun Group is more ambiguous as the PN can occur in various relationships with the KW. This partly depends on the relative positions of the KW and the PN. When the KW precedes the PN (as in pattern 3) the relationship is clearly a naming one, e. g. 'the Lebanese ship Rubinion 18', 'the Battleship Potemkin'. When the KW follows the other name constituents, as in patterns 4 and 5, the relationship between the KW and the PNcons is ambiguous. The following relationships are possible:

- Name, i. e. the preceding constituents directly name the whole compound, as in 'The Black Bird spy plane', 'a Sea Wolf missile'.
- Manufacturer/Owner Corp, i. e. the preceding constituents name the company which owns or makes that object, so we might have 'an Apple computer' or 'A Royal Navy helicopter'.
- Creator, i. e. the preceding constituents name the human creator of the object, as in 'a Van Gogh painting' or 'a Bunuel film'.

The Name relationship reveals OBJECT NDF1. However when the KW is followed by an appositive that clearly gives it a name this will exclude the Name relationship between the PNcons and the KW (this is the the same situation we encountered with Isource names). If we were to see the phrase 'the Black Bird spy plane, Swoop VI, ' we would infer that 'Swoop VI' is the name of the plane, and so 'Black Bird' must be the name of the manufacturer or owner. The main problem here is in differentiating the naming of an object PN from the naming of a corp PN and the object it makes (i. e. types 1 and 2 above). We can use various clues to aid in this decision. For instance where we have a serial number type name (as in 'the A-10 aircraft', 'F-104 fighter bomber') this clearly names the object. It might be thought that the type of determiner would aid in this decision. If we have an indefinite determiner then this often indicates a particular object made by a company, e. g. 'a Panasonic tape recorder', 'a Zanussi washing machine'. A definite determiner should indicate a unique object name, e. g. 'The Gemini space probe'. However this is not always so. It is possible to name classes of objects, rather than unique objects, and in this case we can use an indefinite determiner to refer to one of that class, e. g. 'a Chieftain tank'. We can also use the definite determiner to refer to the whole class of objects manufactured by a particular company, e. g. 'the Amstrad computer'.

A further problem in the classification of post-PNcon KW's is that the KW may actually be part of a corp PN, rather than just having a corp constituent that is its maker or owner. So we can talk about 'an Apple computer', i. e. a computer made by the corp Apple. But we can also talk about 'Apple Computers', i. e. the actual corp itself. One indicator is the lack of determiner (or the plural number). But it is possible to have such a usage without it referring to the corp as a whole, e. g. 'The firm has been using Apple computers for years.' So the case of the KW is also crucial for the decision, if we are talking about a corp, the KW will be capitalised.

The third type of relationship mentioned above, which we have not talked about so far, is that of creator. This is clearly identified by the KW being in the subclass of Artwork nouns (paintings, sculptures, films, books). This category tends to merge into Isource. As many such PN's may be viewed as both objects and isources it may be more advantageous to see it as a separate category on its own. At present, though, we simply put it as a subclass of objects. The class is interesting in that depending on the position of the KW, the PN can describe an artwork, or its creator. In the situation where the KW follows the PNcon (as in 'a Van Gogh painting', 'a Jean Negalusco film') then the PN invariably names the human creator. However, if the KW precedes the PNcon, then the PN is naming the thing created. In the former case the NDF revealed is ROLE NDF5:

PN Genus = [human,name]
PN product = KW

Rules could be drawn up to convert the Product slot into a Role slot, e. g. if KW = painting then Role = artist. When the KW precedes the PNcon then OBJECT NDF1 is revealed. It is possible (and quite common) to have both situations at once, so we could have 'the Joe Orton play Loot', or 'the Monet painting Morning on the Seine').

Objects can also be defined by following a known corp, where the object PN is a product of the known corp. Such a description is used for brand names, and tells little about the exact nature of the object. It simply places it in the object genus. However it is possible to have a KW appended to this pattern, e. g. 'the Mcvities HobNob biscuit'. This case reveals OBJECT NDF2:

PN Genus = [object,name]
PN supertype= KW
PN Gen = n
PN made_by = Corp PN

As with all other categories, object PN's can be conjoined. It is possible to have a conjoined appositive description applying to a single PN, but this has not been observed. Basically objects occur infrequently in news text, and when they do they are most frequently described with a KW. So the most common conjunction pattern is conjunction of KW's, e. g. 'The nuclear submarines Nautilus and Seaview'. However the patterns for all types are contained in appendix N.

## 6.10 Summary

This chapter has presented a description of the common syntactic contexts in which the major classes of PN appear. For each class, we have described the nature of the PN itself, together with the nature of the accompanying material which serves as a WTD. Frequently the WTD is contained within the PN. The syntactic patterns serve both as a descriptive tool and as a tool for the detection of unknown PN's, in that any capitalised words occurring in the right position in a pattern will be classified as a PN of that particular class. Each syntactic pattern has a corresponding semantics which define the PN in terms of genus and differentia information. Table 6.7 below serves as a summary of the major classes of PN we have dealt with, and their distinctive features. The reader may wish to refer back to this while reading the following chapters.

We have tried to keep the description at a high-level throughout, so that it is not tied to any particular implementation. In the next chapter we show how FUNES implements the model presented here, to enable it to analyse and acquire a variety of PN's as it processes the text in which they occur.

| PN CATEGORY | PRECEDING KW | FOLLOWING KW | APPOS | INCLUDE PP'S | DIFFERENTIA | EXAMPLES |
|---|---|---|---|---|---|---|
| PERSONAL | YES | NO | YES | RARE (in present day) | origin, works_for, age, role, related_to assoc, field,boss_of property,product | President Bush / Paddy Ashdown, the leader of the SDP / Steven Potts, 56, a carpenter from Maidstone, / Steven Douglas, brother of Robert Douglas, / Ross Perot, billionaire businessman and predisential candidate, |
| CORP | YES | YES | YES | YES | origin, superpart, field, composed_of, staff, product, property, isa,run_by | Amstrad, the electrical corp, / Hill Samuel Bank, a division of TSB, / National Consumers Council / Ford of Britain/ Campaign for Nuclear Disarmament / Bank of Credit and Commerce International / Allens Catering / |
| LEGIS | NO | YES | RARE | YES | field, origin | Treaty of Rome / Bill of Rights/ Community Care Act,/ British North America Act / Competition and Service Bill |
| PLACE | YES | YES | YES | RARE | superpart, location, property, isa | River Thames / Hyde Park / London / the Polish boarder town of Szczecin / the Uzbekistan capital, Tashkent, |
| OBJECT | YES | YES | YES | RARE | isa, made_by, property, origin, | the Cray C-90 supercomputer/ the Trident missile / Cyclone, Tandem's new high-speed computer / the Lebanese ship Rubinion 18 |
| ISOURCE | YES | YES | YES | YES | isa, origin, property, | NBC-TV's Today program / GQ magazine / the newspaper Al Fayer / Italy's largest circulation daily, Corriere della Sera, |
| EVENT | NO | YES | RARE | YES | isa, origin, | The French Revolution / The Battle of Marston Moor/ The Black Death / |

Table 6.7: Major Descriptive Features of Proper Name Categories

# Chapter 7

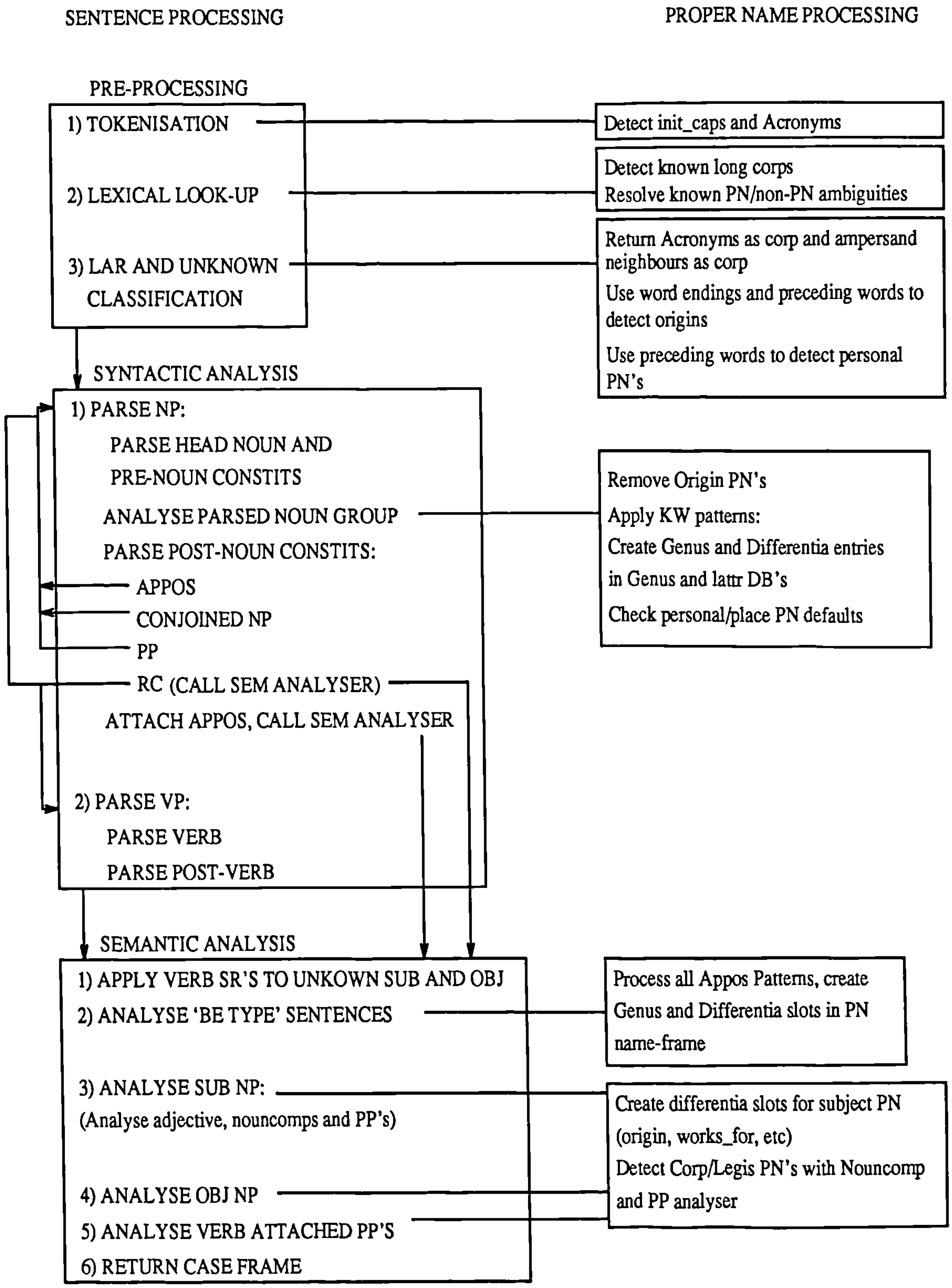
# Computational Analysis of Proper Names in FUNES

## 7.1 Introduction

In this chapter we consider how an NLP system can utilise the detailed model of PN's presented in the last chapter to process the sorts of PN's commonly encountered in news text. We describe how the syntactic patterns can be used to produce detailed grammar rules for the parsing and analysis of PN's, and how the accompanying semantic description can be used to produce lexical and knowledge base entries from the analysis of these patterns. This chapter is meant to serve as an introduction to the topic, and to give a general outline of how our approach works. In the following chapters we turn to an in-depth account of the problems produced by each category of PN introduced in the last chapter, and present solutions for these category-specific problems.

We begin by describing how the KW's of chapter six can be used to produce a genus category and a supertype/role slot for an unknown PN. We then show how any accompanying adjectives, noun complements and PP's can be used to produce additional differentia information, in the form of slots and fillers taken from the set of differentia slots described in chapter six. We then look at the contribution that morphological information can make in the discovery of unknown origin PN's. Next we consider the compilation of the acquired knowledge into frame-like entities which are used to update the system's lexicon and Knowledge Base. Finally we end by discussing problems in the handling of known PN's.

It must be stressed that all of the processing of PN's takes place in the context of general syntactic/semantic analysis, and is not a separate process entirely for the handling of PN's. One of our most important points in this thesis is that as PN's are described linguistically within the sentence in which they occur, their analysis can take place in exactly the same manner as the analysis of a 'normal' sentence. Thus the same processes that derive a semantic form for a sentence like 'Stock prices have collapsed in London overnight', will work in exactly the same way to derive a semantic form for a sentence like 'Judith Hart, former chairman of the Labour Party and member of the Wilson Cabinet, died at Queen Mary University Hospital, Roehampton, south-west London.' However from the semantic form of this second sentence we can derive definitional/descriptive information on all the PN's contained within it. Figure 7.1 shows how the PN analysis described in this chapter fits into the whole Text Understanding process outlined in chapter 3.



SENTENCE PROCESSING
PROPER NAME PROCESSING
PRE-PROCESSING
1) TOKENISATION
2) LEXICAL LOOK-UP
3) LAR AND UNKNOWN CLASSIFICATION
Detect init_caps and Acronyms
Detect known long corps
Resolve known PN/non-PN ambiguities
Return Acronyms as corp and ampersand neighbours as corp
Use word endings and preceding words to detect origins
Use preceding words to detect personal PN's
SYNTACTIC ANALYSIS
1) PARSE NP:
PARSE HEAD NOUN AND PRE-NOUN CONSTITS
ANALYSE PARSED NOUN GROUP
PARSE POST-NOUN CONSTITS:
APPOS
CONJOINED NP
PP
RC (CALL SEM ANALYSER)
ATTACH APPOS, CALL SEM ANALYSER
2) PARSE VP:
PARSE VERB
PARSE POST-VERB
Remove Origin PN's
Apply KW patterns:
Create Genus and Differentia entries in Genus and lattr DB's
Check personal/place PN defaults
SEMANTIC ANALYSIS
1) APPLY VERB SR'S TO UNKOWN SUB AND OBJ
2) ANALYSE 'BE TYPE' SENTENCES
3) ANALYSE SUB NP:
(Analyse adjective, nouncomps and PP's)
4) ANALYSE OBJ NP
5) ANALYSE VERB ATTACHED PP'S
6) RETURN CASE FRAME
Process all Appos Patterns, create Genus and Differentia slots in PN name-frame
Create differentia slots for subject PN (origin, works_for, etc)
Detect Corp/Legis PN's with Nouncomp and PP analyser

Figure 7.1: The Relationship between Proper Name Analysis and Sentence Analysis

## 7.2 The Formation of a Genus Category

The Genus of a PN is formed from its accompanying KW. As described in chapter 6, this can occur both within the same Noun Group as the PN, or within a preceding or following appositive NP. Although exactly the same information is contributed in either case, the two situations can be considered different from a grammatical viewpoint, in that in the former, the KW occurs within the same NP, while in the latter it occurs within a different NP. Within an NLP context this amounts to the more elegant treatment of the former in the syntactic stage of analysis, and the latter in the semantic stage. We begin by describing the processing of a PN and KW which occurs in the syntactic stage.

### 7.2.1 Analysis of Key Word plus PN

The KW-PN/PN-KW pattern can be analysed via the construction of rules which are applied to the Noun Group returned by the system parser. For such a process we need:

a) A list of the constituent nouns (the Nlist), containing information on case, semantic category and an indication of whether they are known or unknown.

b) The Head noun itself (Head), together with its accompanying information.

The process is illustrated in figure 7.2.

Here we focus on the two boxes labelled 'Apply Post-KW Patterns' and 'Apply Preceding KW Patterns'. Two sets of rules are utilised. The first examines the semantic category of the Head, to determine if it is indeed a KW. In the result of these rules failing the second set of rules scan through the Nlist, looking for a preceding KW. Of the PN categories that take a following KW only the following are easily detected in the syntactic stage: place names, object names, event names and isource names. Personal names do not take a following KW, and corp and legis names are best examined in the semantic stage, due to their commonly being composed of several syntactic constituents.

If the Head should be a noun of semantic category place, object, event or isource, it indicates the potential presence of a PN of the corresponding category. As will be described in the following chapters there are peculiarities within each of these categories, as regards the case of the KW and the relationship and attachment of the PN and the KW. The basic principle is the same however — those nouns within the Nlist which are capitalised are considered to comprise the PN.

The genus category of the PN is formed from that of the Head, by simply appending 'name' to the semantic category of the Head. The supertype slot is formed by the creation of an 'isa' link between the PN and the Head. For example, the NP 'an F-104 fighter' has a Head of semantic category 'object'. There is only one noun in the Nlist, and this is capitalised. Therefore the following information can be returned about it:

genus: [object, name]
supertype: fighter

The same procedure applies for those rules which detect preceding KW's. If a noun in the Nlist is of semantic category role, object, corp, isource or place then (providing it is the rightmost KW) any nouns to the right are taken as comprising a PN of the corresponding semantic category. The genus category is formed in the same manner as for following KW's, as is the supertype slot. However where the KW is of semantic category 'role', a role slot is formed instead, and the genus category 'human name' is returned . As an example, consider the NP 'the Liberian rebel leader Charles Taylor'. This contains the KW 'rebel', however this should not be used as there is another KW to the right —

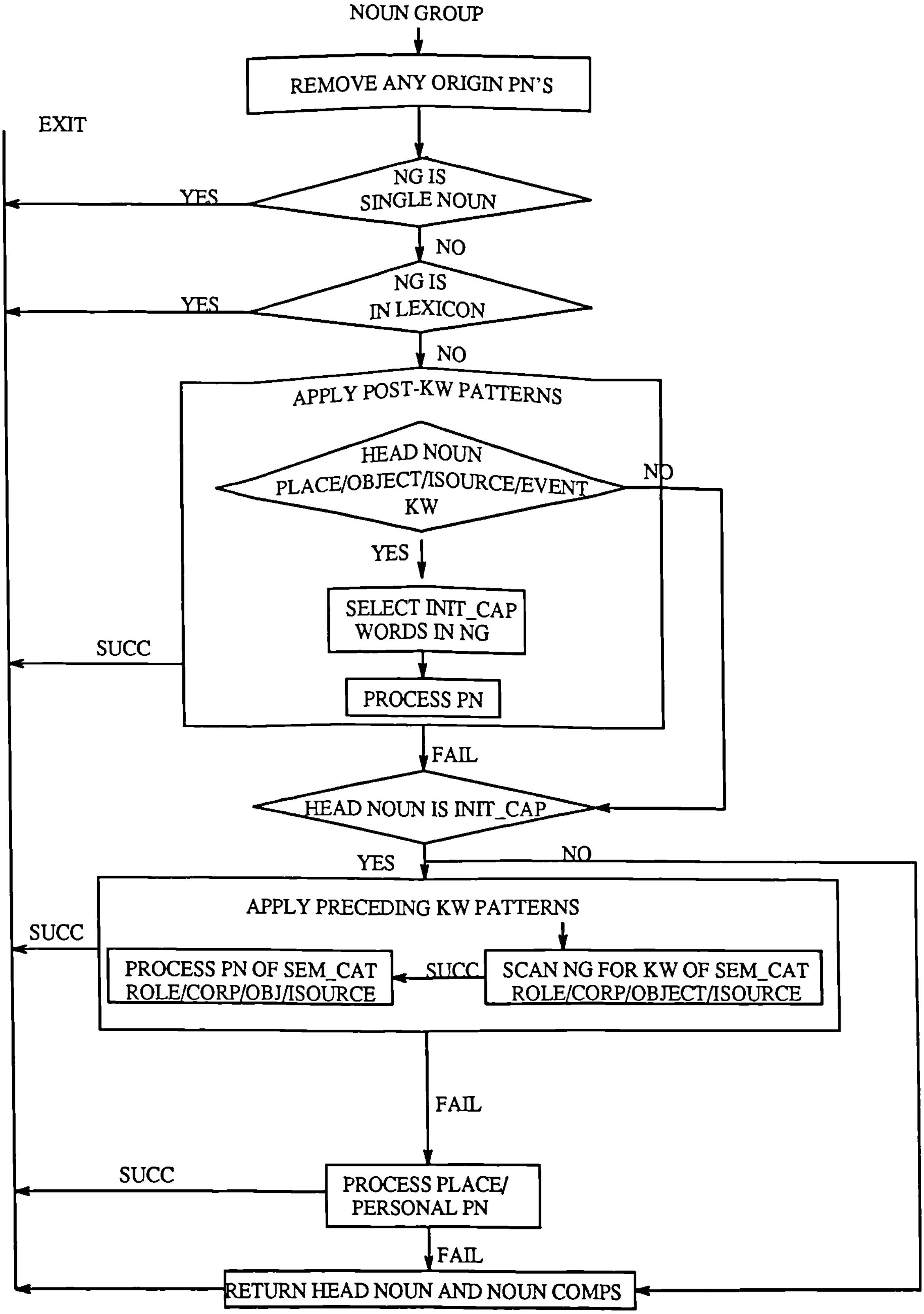

NOUN GROUP
REMOVE ANY ORIGIN PN'S
EXIT
NG IS SINGLE NOUN
YES
NO
NG IS IN LEXICON
YES
NO
APPLY POST-KW PATTERNS
HEAD NOUN PLACE/OBJECT/ISOURCE/EVENT KW
NO
YES
SELECT INIT_CAP WORDS IN NG
SUCC
PROCESS PN
FAIL
HEAD NOUN IS INIT_CAP
YES
NO
APPLY PRECEDING KW PATTERNS
SUCC
PROCESS PN OF SEM_CAT ROLE/CORP/OBJ/ISOURCE
SUCC
SCAN NG FOR KW OF SEM_CAT ROLE/CORP/OBJECT/ISOURCE
FAIL
SUCC
PROCESS PLACE/ PERSONAL PN
FAIL
RETURN HEAD NOUN AND NOUN COMPS

Figure 7.2: The Processing of Proper Name—Key Word Patterns

'leader'. This is utilised to create the following information about the words to its right, in this case 'Charles Taylor':

genus: [human, name]  
role: leader

Additional noun complements contain potential differentia information, which can be derived in a subsequent semantic stage of processing.

A few comments are in order. Before applying the KW detection rules the whole Noun Group can be looked up in the lexicon, to save pointless work should it be known in its entirety. The rules are much facilitated by the prior removal of origin PN's (such as 'Liberian' above), otherwise these can be picked up as part of the PN. Should none of the KW rules match, it is possible that the Noun Group is comprised entirely of Proper Nouns (e. g. 'Michael Williams', 'Hong Kong'). This is checked for if the KW patterns have failed. Finally it may be that the Noun Group is an unknown compound common noun, in which case none of the rules will match, and a default rule will simply return the head noun and accompanying noun complements.

The PN that the KW accompanies may be known. This can be discovered after the specific rules have isolated it within the Noun Group. A comparison between the semantic categories of the PN and the KW will reveal cases of ambiguous common nouns being used as PN's (such as 'Bush', 'Baker', or 'Butler').

### 7.2.2 Analysis of a Key Word within an Appositive Noun Phrase

The definition of KW must be stretched a little here to include such things as digits and directional words, which frequently occur in appositives and can be used to produce genus and other definitional information in the same way as KW's.

In the case of an appositive description, the KW is in a different syntactic constituent from the PN, so treatment by the use of grammar type rules is not as viable. We would argue that the best way to view a PN-appositive pair is as a reduced copular sentence, i. e. a 'be' sentence without the 'be'. This permits the analysis of the three examples below in exactly the same fashion:

'Alberto Fujimori is the President of Peru'.  
'Alberto Fujimori, who is the President of Peru',  
'Alberto Fujimori, the President of Peru',

In all of them the meaning that is being conveyed is the same, (as it is if the PN and descriptive NP's are reversed). Therefore the logical place to analyse such patterns is in the semantic stage of analysis.

The detection and attachment of appositives is a complex process, and will be described in detail in the next chapter. The form of a descriptive appositive NP is the same for all categories of PN. Its detection takes place during the parse of the sentence. Upon completion of the parse of the PN and the appositive, the appositive must be attached to the correct constituent. Problems of attachment ambiguity arise when an appositive follows a PP. Again we postpone detailed description of the resolution of this ambiguity until the next chapter.

The situation we consider here is the application of the KW from the descriptive NP to the PN to create a genus category and a differentia slot. The process comprises several stages:

1) Locate the PN and the descriptive head noun
2) Inspect the semantic category of the head noun
3) Create a link between the PN and the head noun, the nature of this link being dependent on the semantic category of the head noun
4) Create the PN genus category from the head noun semantic category

The method of location of the PN and the descriptive head noun depends on the nature of the system that is processing them. In FUNES they are held in subject noun and object noun registers. The nature of the link between the PN and the descriptive noun can be one of several:

- Location: This is revealed by the descriptive NP head noun being a directional word (e. g. north, south, north-east etc) or a measure noun (e. g. mile or kilometre). In this case the head noun alone cannot be used to fill the Location slot, so a compound term is used, produced from the analysis of the other information in the descriptive NP. As an example the phrase 'San Nicloas, 200 miles north-west of the capital,' would lead to a location term like the following:

```
location :
            direction :(north,west)
            distance : (measure(mile), number(200))
            of :       capital.
```

- Superpart: This is revealed by the presence of two PN's with no determiners, e. g. 'Edgecombe Park Road, Plymouth', or 'London, England'.

- Age: e. g. 'Paul Brian, a 21-year-old,' or 'Paul Brian, 21,'.

- Role: This is revealed by the NP head noun being of semantic category role or relative. An interesting variation on this form is the use of the word 'counterpart', e. g. 'Michio Watanabe, the Japanese foreign minister, met his Indian counterpart, Madhav Sinh Solanki, and urged India to sign the Nuclear Non-Proliferation Treaty'. In this case it makes no sense to give the PN's role as 'counterpart', the previous role must be used to create the present role slot.

- Origin: If the NP head noun is an origin word it is better to recover the country of origin and create an origin slot, rather than to assert an 'isa' slot . For example, from the phrase 'Ginio Ganev, a Bulgarian, ' rather than deriving the slot 'isa : bulgarian', it is more desirable to produce 'origin : bulgaria'. This is the action taken if the head noun is an origin word.

- Isa (also called Supertype): This is the default, covering all cases which have not been matched so far.

The creation of the genus category differs somewhat from the situation where the KW occurs within the same Noun Group. For LOCATION and SUPERPART relationships, the genus returned is [loc, name], (i. e. a place name). For AGE, ORIGIN and ROLE relationships it is [human, name] (i. e. a personal name). Otherwise it is formed as before, by appending 'name' to the semantic category of the descriptive NP head noun.

In FUNES, when this process of linking the PN and its KW has been completed, the semantic analysis of each constituent continues separately, just as it would for a non-descriptive sentence. The analysis of a NP consists of the analysis of accompanying adjectives, noun complements, and attached PP's. These may contribute differentia information on the PN they qualify. In the next section we look briefly at this process.

## 7.3 The Derivation of Additional Differentia Information

Differentia information is carried in adjectives, noun complements and PP's. As described in the last chapter, these constituents can lead to the creation of a variety of slots in the PN description — works_for, origin, field, composed_of, and several others. The differentia slot used in the building of the PN name-frame, and the case label used in the building of the sentence case-frame are exactly the same.

The derivation of a case label for pre-nominal complements (adjectives and noun complements) is achieved by examining each word at a time, to see if its semantic category is one that indicates a specific case. For instance nouns of semantic category 'abstract' tend to contribute 'field' information (e. g. 'transport minister'), while origin PN's contribute origin information (e. g. 'the British Foreign Secretary'). The default case for any complements that have not been analysed is simply 'property'.

When a case has been returned for an adjective or noun complement an entry is also made into the learned attribute DB (or lattr DB, the temporary store used to hold differentia information). This entry takes the form of a triple, e. g. (X, origin, britain). The first argument of the triple depends on the way in which the present NP occurred. If it was a descriptive NP in apposition to a PN, then the PN is used. So 'Douglas Hurd, the British Foreign Secretary' would lead to a triple of the form '(hurd, origin, britain)'. However, if the present NP is not in apposition then the head noun of the NP is used, in the above example the compound noun 'foreign secretary'.

In the case of corp PN's that are composed of several Capitalised Constituents, then the whole corp PN is used as the lattr DB entry. For example, the corp PN 'British Nuclear Fuels' would lead to the creation of two entries, both indexed on the PN itself, and conveying the differentia information 'origin(britain)' and 'field(nuclear)'.

The derivation of case labels for post-nominal complements (PP's) is more complicated, and was described in chapter 4. Appendix H contains a full specification of each of the cases which contribute differentia information on PN's. In the following chapters we will examine each of these in detail, as different cases tend to be associated with different categories of PN. The basic procedure for entering the differentia information into the lattr DB is the same however. When a PP case is returned, it is inspected, and if it is a case conveying differentia information a triple is entered into the lattr DB.

This brief description completes the overview of the handling of PN's from the point of view of creation of genus and differentia slots. It can be seen that the process is a multi-stage one, with information from each level of processing contributing to the final definition. The contribution of the various stages of analysis is summarised below:

1. Syntactic Analysis : Identify PN's and apply KW's to neighbouring PN's to produce genus and role/supertype slot.
2. Semantic Analysis:
   (a) Apply KW's from appositive NP's to produce genus and major differentia information.
   (b) Analyse accompanying adjectives and noun complements to produce further differentia information and detect corp and legis PN's.
   (c) Analyse PP's for same purpose.
3. Compilation Stage (not described above): gather together all information to produce lexical and knowledge base entries for PN's.

Next we consider the role that morphological information can play in the analysis of PN's.

## 7.4 Morphological Information

Although the processing of morphological information is typically the first step in any NLP system we have postponed consideration of it to enable a reader to get an overall feel for our approach to the computational processing of PN's, before exploring specific aspects in more depth. We begin by looking at problems connected with standard morphological rules in the handling of PN's.

### 7.4.1 Morphological Problems

A great problem in the handling of PN's is the question of the application of normal morphological rules to words that may be PN's, i. e. words with initial capitals. Words that fall into the Proper Noun category of chapter 6 will not be analysable by these rules, e. g. stripping the 's' off 'Frances' will not be an appropriate action. However words that fall into the Capitalised Constituent (CC) category will be analysable, as they are merely common words spelled with initial capitals, e. g. 'General Union of Voluntary **Societies**'. The class of ambiguous nouns that are in fact PNouns even though they appear to be CC's can also cause problems. For instance, stripping the 's' off 'Childs' or 'Miles' will be wrong as it will reveal a known common noun root that it has no connection with.

This problem also concerns efficiency. For instance capitalised comparative and superlative adjectives are seen very rarely, if at all, in the news text we have examined. [1] Therefore, it is inefficient to apply the rules to derive root adjective forms to all words not found in the lexicon, as the chances of them applying to a capitalised word are minute.

Our solution is an heuristic one. Rules to strip 's' are applied but with the proviso that if a known name is revealed the rule is reversed. This would prevent production of the name 'Carlo' from 'Carlos', or 'France' from 'Frances'. To overcome the problem of incorrectly revealing known roots some common cases of human names that end in 's' have been entered into the initial lexicon (e. g. 'Childs', 'Miles', 'Parks'). If it should still happen that a known root is incorrectly derived from a PN we leave it up to the noun group analyser to override the incorrect meaning. [2] This decision was taken on the grounds that the number of CC's that end in 's', and which thus should be stripped, is greater than the number of ambiguous Proper Nouns that end in 's' and have a common word root, and so should not be stripped.

Rules to derive verb or adjective roots are not applied to potential PN's, on efficiency grounds.

A second morphological-based problem that has plagued simple PN extractors is the problem of sentence initial words being classified as PN's. Within an NLP system this causes few problems, as each word is looked up in the system lexicon. So known sentence initial words will be in no danger of being classified as PN's. The only problem comes from sentence initials that are unknown. These may be PN's or they may be normal words. The potential problems this may cause are good reason for having as complete a base lexicon as possible. If a sentence initial is unknown, and it is not followed by another adjective

[1] We exclude sentence initial words here. Imperatives have been found rarely in news, and as they are not covered by the FUNES grammar we do not consider them further

[2] As a system builds up its lexicon of personal first names, ambiguous surnames such as these will become less of a problem, as the preceding known name can be used to indicate that they are also names. It appears that first names are less prone to ambiguity than second names.

nor ends in 'ly', then it will be classified as an unknown noun. However there is very little chance of it erroneously being classified as a PN if it is not, as the syntactic pattern in which it occurs is unlikely to be one of those described in chapter 6. The only pattern which it might match would be a 'PN KW' pattern, and due to the fact that the KW would not be capitalised the only categories which would make a match would be places or objects. For instance if the word 'rough' were unknown and occurred at the start of a sentence such as 'Rough seas caused ...', or the word 'enemy' occurred as 'Enemy missiles ... ', then these unknowns would be processed as place or object PN's. [3] We have not had such a case occur in FUNES, which only has a base lexicon of 2,000 roots. With a more realistic lexicon (say 10,000 roots) the chances of occurrence are minute.

Having discussed the complications which PN's present for a system as regards the application of conventional morphological rules, we now turn to discuss any advantages that morphology can offer in the identification of PN's.

### 7.4.2 PN-oriented Morphological Heuristics

Although the derivation of morphological rules for most classes of PN does not seem possible, certain word endings are indicative of the origin class of PN. However, just as the application of standard morphological heuristics to PN's can produce errors, so can the application of origin PN-oriented rules to normal words and to other categories of PN.

We can prevent many problems in the former area by not applying origin PN heuristics to non-capitalised words. However we still have to contend with possible errors that may be caused by the application of such rules to other classes of PN or to CC's. The solution is to be very judicious in the heuristics that are used.

The ending '-ian', for example, was originally used to classify a word as an origin PN. However as more example texts were processed, and more news text scanned, the number of counter-examples increased to such a point that the heuristic could no longer be used effectively. Words such as Brian, Qian, Jian, [4] Caucasian, Sebastian, and many more, mean that it cannot reliably be used on its own. So the heuristic has been adapted (along with the entry of most countries into the initial lexicon) to look up the word after removal of the n, and if the root is a known place PN, to assign the word as an origin PN.

The use of the endings '-ia', '-ica' and '-land' to classify a word as a place PN are similarly unreliable. In fact, it was eventually decided in FUNES that a simpler solution to the problem of place and origin PN's was to include those referring to the world's countries in the initial lexicon. This decision was based on the unreliability of morphological heuristics, the importance of reliable identification of origin PN's for facilitating the application of other PN patterns, and the relatively small number of entries needed. However there still remain a large number of regionally based origin PN's, and so the morphological rules (which have been found to be more reliable) shown in Table 7.1 have been retained. These heuristics are only applied to capitalised words. Moreover preceding words that provide more definite information as to a semantic category (such as name determiners like 'dr' and 'mr', known human names, and directional words like 'central' or 'northern') are inspected first.

We now turn to look at the problem of common noun/ Proper Name ambiguity.

[3] It might be thought that because the KW is plural this should prevent the PN rule firing. This is not so, due to conjunction cases, such as 'the Adriactic and Aegean seas'. However, the position of the unknown could be monitored, and if it was sentence initial, and the KW was plural, then the PN rule could be blocked.

[4] A simple length limit check on the stem could be used to prevent short words like this being misclassified, but would not handle the longer counter-examples.

```
if words ends in '-ish'
  then return as Proper Adjective
else if ends in 'ese'
        then return as origin PN
else if ends in 'cans' or 'ians'
        then return as origin PN minus s
else if ends in 'can' or 'ian' and root is known loc
        then return as origin PN
else if ends in 'shire'
         then return as place name
```

Table 7.1: Heuristics for Classifying Place and Origin Proper Names

## 7.5 PN Ambiguity

The problem of common noun/PN ambiguity refers to the case where what appears to be a Capitalised Constituent is really a Proper Noun. This is mainly a problem in the processing of personal PN's. Examples we have mentioned already include the names Bush, Thatcher, and Major. For a correct analysis, the system processing such words must realise that they are names, and thus the definition obtained from the lexicon should be abandoned.

The fact that these words appear as CC's is the real source of the problem. Due to the fact that many CC's appear in PN's and retain their lexical definition we cannot blindly throw away the definition simply because the word is capitalised. The solution to this problem rests on the fact that in the majority of cases, surnames will not appear on their own (excluding headlines). The words that surround the ambiguous constituent can be used to signal the fact that the lexical definition does not apply in this case.

In a news item we can expect to see a surname accompanied by one of the following — a name determiner, a firstname, or a role KW. Each of these contributes the information that the following word is in fact a human name. It is in the syntactic stage of analysis that this problem is most effectively addressed, as it is here that the KW's are used to create a definition for the accompanying PN. As explained in the first section of this chapter, when the PN has been located it is looked up in the lexicon. The recovered definition (assuming one is recovered) is compared to that of the KW. If they should not be the same, then it is assumed that the PN is an ambiguous common noun. A new semantic category is created for it using the category of the KW, and from then on it is treated as a PN.

If the ambiguous surname is accompanied by a name determiner then it is simply returned as a personal name. If not, the whole PN will be looked up in the lexicon, and if known the definition returned. If it is not known the capitalised nouns are scanned for their semantic categories, and if any one should be a known human name then the whole group is returned as a personal PN.

Where an ambiguous word is very common, either because it is the name of a prominent figure, or because it is simply a popular surname (e. g. 'Brown', 'White', or 'Black'), the two forms are both held in the lexicon. This decision is motivated by efficiency concerns — a word such as 'Bush' is at present going to occur as a PN very many times. To save the system going through the process of redefining it each time, it is entered into the lexicon in both its common noun and PN meanings. When a word with a dual entry is retrieved from the lexicon, its initial letter can be checked, and, if this is capitalised, then the PN meaning is used. Such an approach should not be seen as a failing on the system's behalf, as in the majority of cases it could determine the PN usage anyway, and would enter the new meaning itself.

In the rare case that an ambiguous word does occur on its own, then higher level functions such as verb or prepositional selectional restrictions could be used to signal the ambiguity. For example, the verb 'shoot down' takes an object or human noun as its object, so the sentence 'Six Tornadoes were shot down' would produce a violation.

It was originally feared that the problem of name/non-name ambiguity would be a large one to overcome. This was due to a consideration of the problem in the worse case (i. e. a single ambiguous word with no dis-ambiguating context). However, as often happens, an examination of real data shows that this worse case simply does not occur with any great regularity, and so the ambiguity problem remains troublesome in theory but not in practice.

We have now almost completed our introductory account of PN handling. Two things remain. First we consider the issue of combining all the acquired knowledge, and forming new entries in the lexicon and Knowledge Base. Secondly, we look at the issue of handling known PN's, and the differences and similarities between this and the handling of unknowns.

## 7.6 PN Knowledge Compilation

We stated previously that the derivation of definitions/descriptions on PN's was a multi-stage process, in which different facts were derived and recorded at each level of processing. Obviously at some stage all this diverse knowledge must be drawn together into a single unit. We refer to this as 'name-frame compilation' (so the output of our system can be seen as composed of two types of 'frame' — case-frames and name-frames).

This process is carried out at the end of each sentence. A final process is initiated at the end of the entire story, in which the 'name-frame' is entered into the system lexicon and knowledge base. The name-frame compilation is illustrated in the flow-chart in figure 7.3 below.

To clarify this process we must explain the various storage mechanisms which hold unknown nouns and their descriptive information. Firstly there is the 'Unknown DB' — all nouns which are not found in the lexicon are entered into this in the syntactic stage. This permits the name-frame compilation stage to immediately find all the unknown words in a text. Secondly there is the 'Genus DB' — this holds each unknown together with the genus category (or categories) that have been established for it. This DB was described in chapter 4. Each time an unknown is encountered in a story, the genus category derived from that particular encounter is entered into this DB. If the unknown should already have an entry here the new category is compared to the existing categories. This permits the hypothesized genus categories to be refined as new descriptive contexts are analysed. The context which led to the particular semantic category being created is noted in the comparison procedure, so that weaker contexts will not over-ride stronger contexts. For instance, semantic categories created from KW's cannot be over-ridden by those created from verb or prepositional selectional restrictions. Finally there is the learned attribute DB, or lattr DB — this holds all the differentia information derived on PN's (common nouns will not have differentia information derived). Each entry in this DB takes the form

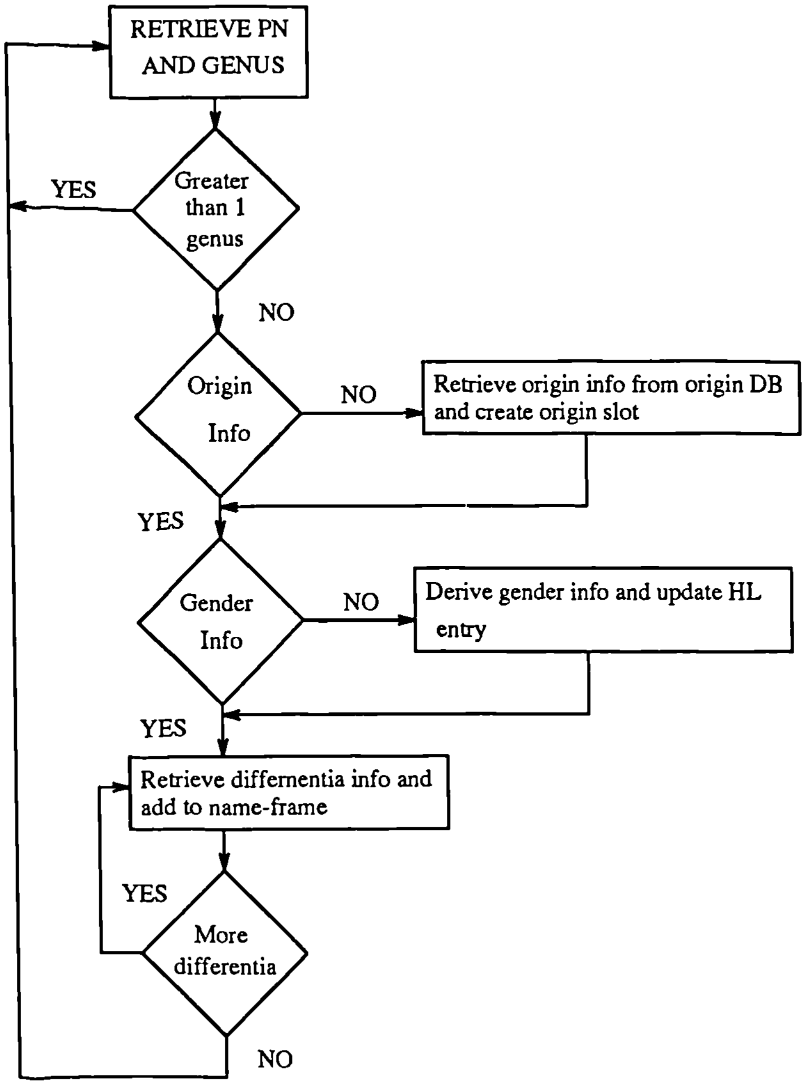

RETRIEVE PN AND GENUS
YES
Greater than 1 genus
NO
Origin Info
NO
Retrieve origin info from origin DB and create origin slot
YES
Gender Info
NO
Derive gender info and update HL entry
YES
Retrieve differnentia info and add to name-frame
YES
More differentia
NO

Figure 7.3: Name Frame Compilation

of a triple :

(Frame, Differentia_slot, Slot_fill)
e. g. (bush, role, president)

The name-frame compilation stage utilises the contents of these DB's to build lexical and knowledge base entries for the unknown words. At the end of each sentence the Unknown and lattr DB's are erased.

As shown in the diagram, at the end of each sentence, each unknown is withdrawn from the unknown DB, and looked up in the Genus DB. If it has a single genus category it is retrieved for name-frame compilation. (If an unknown should have more than one genus category then these are retained in the Genus DB for future refinement). This process consists of locating origin, gender and other differentia information acquired throughout the analysis of the sentence, and entering it into a name-frame for the unknown in question. Gender information can be picked up from role words or pronoun reference. If from the former, the history list entry must be updated to increase the chance of a future successful pronoun reference. Differentia information on unknown PN's is retrieved from the lattr DB. The name-frames produced have the form shown below:

```
[noun(Noun, Sem_cat, Gender), Slot1:Fill1,... Slotn:Filln]
```

Slot labels are taken from the set of differentia slots described in chapter 6. Slot fills are lexical items or compound terms. Compound terms are produced where the slot fill itself has an entry in the lattr DB. For instance, when processing the phrase

'Rudolph Agnew, former chairman of Consolidated Gold Fields,

the word 'former' will be entered into the lattr DB in a triple with the word 'chairman', which will be the slot fill in a triple with the word 'Agnew'. So that this information can be entered into the name-frame, all entries [5] found in the lattr DB as slot fills are checked to see if they also have entries as frames. The final role slot produced for 'Rudolph Agnew' in the above example would be '[role: [chairman, property(former)]]'.

The final processing of the name-frames takes place at the end of the processing of the whole news item. Here additional frames are created for the firstnames and surnames if the PN is a personal PN. This enables the system to utilise the firstname and surname should they occur in another person's name (e. g. after processing 'Michael Smith' subsequent exposure to 'Michael Peters' would be facilitated as the first name would be known). These are entered into the lexicon in the standard form, i. e. noun(Noun, Sem_cat,Gender). The name-frame for the whole name is then split up, and the Head entered into the lexicon in the above format, the tail (consisting of the differentia slots) being entered into the knowledge base.

Within the KB personal names are indexed to the surname, in the form shown below:

```
kbase(Surname):
[Slot1: Fill1 ... Slotn: Filln], e. g.
kbase(bush):
[firstname: george, fullname: [george, bush], role: [president],
 origin: [united, state]]
```

Thus far in this chapter we have been mainly considering the handling of unknown PN's. It might be thought that unknown PN's are the main source of difficulty for an NLP system. However this is far from so, as we show in the concluding discussion on handling known PN's.

## 7.7 The Processing of Known PN's

The initial motivation for this thesis was the problems that unknown words present to NLP systems working on real, un-edited text. We initially focused on PN's as they were continually cited as the main source of unknown items in evaluative studies. This is indeed a formidable problem for NLP systems seeking to process data such as news text, which abounds in obscure PN's. However, we soon came to realise that it is not just their novelty that makes PN's a problem for Computational Linguistics. A problem just as large is the complexity of their syntactic structure, and of the syntactic contexts in which they occur. Thus, even if our system had entries for all the PN's within a text in its lexicon, this would not solve all the problems of PN analysis. For example, even if we should know that 'Jose' and 'Rameros' are human names, this still leaves a great deal to analyse when

[5] except for fullname and firstname entries

encountering the phrase 'Arce Battalion commander Jose Rameros, a long-time member of the FMLN and extreme right-wing fanatic, ...'.

When dealing with known PN's, the NP's that contain them must still be analysed in the same way as if they were unknown, so that the PN itself can be revealed. When the PN is a complex company name construction it may avoid detection altogether in a simple lexical look-up stage, and thus an analyser that is aware of the construction of corp PN's is needed to detect the name in a later stage. To a large extent, therefore, the processing of known and unknown PN's is identical. The main differences are that we do not have to construct a semantic category for known PN's, and that for corp PN's we do not have to work out the relationship of the component words to the whole PN. The extent to which we analyse differentia information is more debatable, as we discuss below.

There are two extremes to answering this question, and each depends very much on the application that the NLP system is pursuing. At one extreme is the approach that says if a PN is known then leave it alone, all that we need is knowledge that it is a noun and its semantic category. Such an approach might be taken by a system undertaking a rapid and shallow analysis of large amounts of text, for some sort of routing or summarisation application. Here the focus is on throughput. At the other extreme is an approach that says even if a PN is known, treat it as if it were not and learn all the information that is available, comparing this to existing information and retaining that which is new. Such an approach might be taken by a system undertaking a deep analysis of texts, with a view to answering questions on the participants' actions and reasons for these. In such a scenario detailed information on people, places and companies would be very helpful. Another approach that would benefit from this approach would be one which was assisting in building up detailed profiles of people and companies for DB creation or publication of 'Who's Who' type gazetteers.

What should be stressed is that in either case, a system that is undertaking a full syntactic/semantic analysis will produce descriptive information on the PN's appearing within the text, as this information is conveyed by the text, in the same way as the event-oriented information within the text. The correct analysis of the NP 'George Bush, the US President, ', for example, demands a semantic representation that defines George Bush as the US President.

However, it is the use that is made of this information that is at issue. Our approach to handling known PN's involves gaining as much information about them as possible — almost treating them as if they were unknown. At the end of processing an input text this new information is then compared to that held in the knowledge base, and that which is held to be new retained and added to the existing entry. The issue of which information clashes with the existing information, and which does not is an interesting one, and will be taken up again in the next chapter. It is possible that, if the new information is completely different from the old, then we are in fact dealing with a new entity that just happens to share a name with the existing one.

## 7.8 Summary

In this chapter we have introduced our approach to the computational handling of PN's, based on the syntactic and semantic description of PN's and their surrounding context outlined in chapter 6. We have described:

- The application of morphological heuristics. We have shown how the application of standard morphological rules to PN's can lead to erroneous classification, and how some origin PN's can be identified via morphology.

- The application of KW patterns in the syntactic stage of processing. These can be used to created genus and role/supertype slots for unknown PN's, and permit the detection of known PN's.

- The application of KW/appositive patterns in the semantic stage of processing.

- The derivation of differentia information, also in the semantic stage.

- The compilation of the knowledge gained from the above processing into lexical and KB entries for the unknown PN's.

- The issues arising from encounters with known PN's. We have shown that it is not simply the novelty of many PN's that can cause problems for a system attempting to process news text, but the complexity of the PN structure and the structure of the surrounding context.

The chapter was intended to give the reader a simple outline of the approach and its implementation in FUNES. This, it is hoped, will help in understanding the discussion of category-specific issues that appears in the following chapters. In the next chapter we describe personal PN's. In chapter 9 we describe corp and legis PN's, and in chapter 10 we describe the remaining categories — places, isources, events, and objects.

# Chapter 8

# The Structure and Analysis of Personal Names

## 8.1 Introduction

In this chapter we describe the problems which personal PN's present for computational analysis. We illustrate this discussion with solutions we have implemented in the FUNES system. However, the problems and solutions are described in as system-independent a way as possible, to aid their widest application. We structure the discussion around the different levels of processing a PN will receive in a typical NLP system. These are outlined below:

1. Pre-Processing. At this level, the only constituents we have to operate on are individual words. Even so, personal PN's can often be identified by an examination of the words which precede them.

2. Syntactic Analysis. At this level the parser is building individual words into syntactic units, permitting the analysis of a group of words as a single NP (for example). This enables more global contexts to be used to provide descriptions for PN's, such as the use of KW's to classify several following words as a personal PN. This grouping also enables apposition and conjunction, and the attachment problems they present, to be handled.

3. Semantic Analysis. The semantic level deals with the relationship between different syntactic units. At this level, therefore, we can utilise yet wider contexts for the description of PN's, such as the relationship between two NP's in apposition, and the case relationship between a PP and the NP to which it attaches.

This chapter looks at all of these stages, and the problems which arise in each. We also discuss the handling of known PN's, which is in many ways, the same as the handling of unknowns.

We begin with an examination of pre-processing and lexical representation issues.

## 8.2 Personal PN Pre-Processing and Lexical Representation

Known personal PN's can be held in the lexicon, in the same way as normal lexical items. For a system that is automatically updating its lexicon, it is advisable to hold the PN's in a separate lexicon, which it is always possible to edit and revise. In the FUNES system, all personal PN's are held as nouns, distinguished by the term 'name' appended to their semantic category, e. g.

8.1 noun(john, [human,name], m)
noun([margaret,thatcher], [human,name], f)

No personal PN's were supplied by hand, except for some common nouns which often occur as personal PN's (e. g. 'White', 'Green', 'Miles', 'Major'). As described in chapter 7, these ambiguous words are held with two definitions, and the letter case of the word as it occurs is used to choose the correct definition.

In the opinion of some (e. g. [108]) a lexicon should only hold those items whose meaning is not in any way derivable from their constituent parts. Although this might be very nice in theory, in NLP it greatly facilitates processing to hold as many compounds as possible in the lexicon. This concern also applies to PN's, which can be viewed as lexical compounds. Thus, as well as holding 'john' and 'smith', it will also facilitate processing to hold 'john smith' as a separate item. This approach only makes sense for famous individuals who are likely to occur very often in news. Such an approach is also advocated in [130]. However, we do not want to hold every single fullname we process, or the lexicon will soon grow to an unmanageable size. There are various ways of handling this problem. Lexical entries could be sent to a user for approval before entry, or a user could manually scan the lexicon regularly and remove those enties she felt were unnecessary. A system could also handle the problem automatically, by keeping count of the number of times PN's were accessed, and if this number fell below a certain threshold removing the entry.

The detection of fullnames is another issue which needs consideration. Normal compounds are indexed on their first word, and if this word is noticed in a text the compound is searched for. However, for personal PN's this would soon become a slow process if there were many entries for the same firstname. Our solution in FUNES is to hold the name as a compound, but not index it on the first word. Instead, discovery of the fullname is left until the syntactic stage.

In the pre-processing stage the only items which are available for examination are individual words, which have been separated out by a tokeniser. As no syntactic structure is available, the only readily accessible items which can be used to classify unknown words are the preceding and following words. For a parser to perform its task, the parts of speech of every word it encounters must be known, hence the necessity for a pre-processor to determine this information for all unknowns. With unknown capitalised words the examination of the preceding word can also reveal the unknown's particular semantic category. For personal PN's, the following heuristics have been developed which will classify an unknown capitalised word as a personal PN if:

1) it is preceded by a name determiner (such as Mrs, Dr, Sir)
2) it is preceded by a role KW (such as President, Prince, nurse)
3) it is preceded by a known personal name or name component (such as van or du).

We have seen a similar approach in most of the existing work that has considered the problem of PN's. However, the majority of this work has had an ad hoc feel, where each case is handled as it appears. Our approach utilises a detailed lexicon of KW's, and general rules which are applicable to all cases of PN's that have been encountered, and it is felt, to most cases that ever will be encountered in news text.

The morphology of personal PN's has not been investigated as it appears to offer no obvious clues to their semantic class. In [50] some systems developed heuristics for the detection of Spanish personal names, e. g. unknowns ending in 'z' or 'o' were classified as Spanish names by the PRC Praktus system. As these were likely to be the only unknowns encountered, this approach worked. But where numerous other types of unknown words may be encountered, including any sort of name, from any language group, such an approach is not at all feasible.

## 8.3 The Syntactic Processing of Personal PN's

As we move through the various stages of analysis and combine the constituent words of a sentence into larger and larger groups, more distant words become available for use in describing PN's. Personal PN's are composed of nouns, and so they will be analysed by a NP parser. The separation of a group of words into a single NP enables us to utilise the KW's introduced in chapter 6 for classifying several words as a single personal PN.

In this section we discuss the use of KW's in the description of personal PN's, and an heuristic approach to handling personal PN's which occur with no KW. We also discuss the problems of apposition, and outline methods for detecting and attaching appositives correctly. Finally we look at NP-NP conjunction, and show how this interacts with apposition to greatly complicate the analysis of personal PN's.

### 8.3.1 The Use of Role Keyword's in Classification of Personal PN's

Personal PN's commonly occur with a preceding KW. This KW can be used to produce a semantic category and role information for an unknown PN. Even if the PN which follows is known, the KW can be utilised in its detection, and in checking for consistency between the lexical definition of the PN and the definition given by the KW.

If the PN has no KW, and is known, then it is easily detected after the parser has separated out the Noun Group, by looking up the whole Group in the lexicon. For example, in the sentence

**8.2** 'Paddy Ashdown has hit out at rumours that he had an affair with his secretary'

the PN 'Paddy Ashdown' forms an entire NP. If this is looked up after it has been parsed, and 'Paddy Ashdown' is known, then an entry will be found in the lexicon, and the words discovered to form a single personal PN, which can henceforth be processed as a single entity.

Detection of a role KW is easily carried out by scanning the Noun Group returned by the parser. When such a KW is detected, if there are nouns to the right and they are capitalised, then these can classified as a single personal PN. The role KW thus provides a way of locating a personal PN which may be embedded within a complex Noun Group, e. g.

**8.3** 'former British Nuclear Fuels executive vice-president Derek Peterson'

When the PN has been located it can be looked up in the lexicon to discover if it is known or not. If known, the semantic category contained in its lexical entry must be compared to that of the KW. This comparison will detect ambiguous common nouns that can also occur as PN's, such as 'Bush', 'Baker' or 'Thatcher'. If the two semantic categories do not match this means the semantic category returned from the lexicon should be abandoned for the semantic category supplied by the KW. For example, after parsing the NP 'Secretary Baker', detection of the KW would lead to 'Baker' being looked up in the lexicon. If only the common noun definition were known, this would be rejected after comparison with

| Info Type | Example |
|---|---|
| Plural KW in Preceding NP | Directors Roger Billings and James Miller |
| Name-type Determiner | Mr Keith Lynch, Sir David Attenborough |
| 1 noun in NG is a known name | Peter Smith (if Peter is known as a human name implies 'Peter Smith' is also a human name). |
| All nouns in NG are unknown | Marion Francesca Boulty |
| All nouns except 1 unknown | Steven Hill, Mark Butler, Dominic Rush |

Table 8.1: Methods for Detection of Lone Personal Proper Names

'Secretary', and a new semantic category of 'human name' created. If the PN is not found in the lexicon, then no comparison is necessary, and the KW semantic category of 'role' leads immediately to a semantic category of 'human name' for the unknown.

In FUNES, the semantic category is returned with the PN, and entered into the appropriate NP register. The lattr DB is used to hold the role information provided by the KW. Only the surname is retained for semantic analysis, so 'firstname' and 'fullname' triples are also entered into the lattr DB. From example 8.3 above, the following triples would be entered into the lattr DB:

8.4 (peterson, firstname, derek)
(peterson, fullname, [derek, peterson])
(peterson, role, [executive,vice,president])

The occurrence of a role KW is a great aid in the analysis and processing of personal PN's. However, personal PN's do not always occur with a KW, they can also occur alone. This is discussed in the next section.

### 8.3.2 Analysis of Lone Personal PN's

When a personal PN occurs with no preceding KW, various methods can be used to produce a semantic category. Firstly, if it is followed by an appositive this will provide the missing category. However, this will not be available during the Noun Group analysis of the PN, and so cannot be used immediately. Even with no appositive, more global constraints could be used (such as selectional restrictions) to produce a semantic category. Again these are not available for use at this point, and moreover they only assign broad categories (as described in chapter 4).

It is possible, though, to consider a method based around default categories. If the Noun Group is unknown, and no KW's have been detected, and it is comprised of capitalised nouns, then it is must be a PN, and most likely a personal PN. This is a reasonable hypothesis, but we cannot be exactly sure — due to the similarities between personal, place and corp PN's it could actually be any of these. Although a personal PN is the most likely (since this is the commonest type of PN, and most other types of PN will occur with a KW), it is sensible to look for any other dis-ambiguating information.

The information which can be used is shown in Table 8.1. Firstly, it is possible that this PN is conjoined to a preceding PN which contained a plural KW. If a record is kept of the preceding KW and its number, then this can be utilised to build a definition for the present PN.

Secondly, if a name-type determiner was present in the Noun Group then this clearly indicates a personal PN. A personal PN is also definitely indicated if one or more of the capitalised nouns is known as a human name.

If none of these is the case then our confidence is reduced, although the personal PN hypothesis is still most likely. Place PN's most often occur in 'at type' PP's, so, if this is the case a place PN hypothesis should be preferred. If not we utilise a heuristic approach. This examines the semantic category of each of the nouns, and if all are unknown then a personal PN is returned. Even with a lexicon of only 2,000 roots it is unlikely that all of the words in an unknown compound noun would be unknown, whereas it is quite likely that all of the words in a personal PN would be (bearing in mind that if one of them was a known human name then the whole has already been returned as a human name). However, there is one final problem, which concerns the occurrence of ambiguous common nouns in personal PN's. Names like 'Peter Butler' or 'Steven Hill' will evade this heuristic, if their name constituent is not known. To overcome this problem, one of the words in the group is allowed to be known, but not of semantic category corp/legis/isource/object. These categories are disallowed because of their rarity in personal PN's, and because of their frequency in corp PN's (for in FUNES corp PN's with head noun KW's are not analysed until the semantic stage).

These heuristics have performed well. The main problem is confusion between personal and corp names. These can take similar forms (as corp PN's can be named after their founders). As the default category is personal name, a corp name such as 'Price Waterhouse' would be returned as a personal name. If it is later given a definition by an appositive the earlier incorrect decision can be over-ridden. In the unlikely case that it was not given a further description, then it would be entered with an incorrect semantic category. As such a case is so rare, this is not considered a great problem. An additional consideration is the type of news one is intending to process. In the domain of general news, setting the default to personal name is correct. However if one was specifically targeting business news, where corp PN's are as common as personal PN's, the personal default might be reconsidered. This also depends on how concerned one is about incorrect entries. Even in business news where the number of corp and personal names is about equal, the large majority of corp PN's will contain a KW. So the number of personal PN's without KW's will still exceed that of corp PN's without KW's. If it is very important to avoid errors though, then these non-KW PN's could be returned with two hypothesized categories, which future exposure could refine.

This completes the discussion of issues arising from the occurrence of single personal PN's within an NP. Next we consider the problem of apposition. From a syntactic viewpoint the main issues are detection and attachment.

### 8.3.3 The Appositive Noun Phrase

We define an appositive NP as an NP that is enclosed in commas and follows a previous NP, e. g.

8.5 Ahn Byong Hwa, the trade minister,
His wife, Fermina Diza,

The function of the appositive is to give descriptive information about a preceding PN; or it may supply a PN for the description provided in the previous NP. Although there appears to be no rule saying that appositives should only be used in a pairing with a PN, this is invariably the case. Appositives are ubiquitous in news text, and any system seeking to parse this sort of text must be able to detect, parse, and attach them.

As appositives are always enclosed in commas, the presence of the comma is helpful in signalling the potential presence of an appositive. However, there are many other constructions which can follow a comma, so their presence is not conclusive. In the FUNES system, upon detection of a comma, the look-ahead buffer of unparsed words is inspected to decide if an appositive is present.

The complete analysis of an appositive consists of detection, attachment and semantic analysis of the appositive and the NP to which it is attached. We begin by looking at the process of detection.

### The Detection of an Appositive

When a post-noun comma occurs, a variety of constituents can follow. Below we describe these, and methods for processing them.

- The comma is followed by the word 'called' or an adverb and then the word 'called'. This type of construction can effectively be treated as a normal appositive, by removing the word 'called'.

- the next word (NW) is a form of 'write'. This is a common pattern in UK news, the word 'write' introducing the author of the article or a phrase such as 'writes our foreign correspondent'. As such information is redundant for an understanding of the article we have found the simplest way to handle it is to end the parse at this point, i. e. ignore the material following the comma. A similar type of construction places the author or source of the story first, and the verb second, e. g. 'officials reported', 'a source said', 'the Japanese foreign office reported yesterday'. This is a less clearly flagged construction, as the constituent that actually follows the comma is an NP. We have implemented a heuristic solution which tests for the word 'say' or 'report' within six words of the comma and within two words of the end of the sentence. If this pattern is found the parse is again ended at this point.

- The NW is 'however'. When this occurs in such a position it can simply be removed and the look-ahead buffer re-examined. In most cases it will be followed by a VP.

- The comma is followed by ', respectively, '. Although this word is important for a high-level understanding of a sentence, as it indicates specific attachment criteria, it does not indicate an appositive. As the FUNES system cannot make use of the information it conveys we simply remove it.

- The NW is a verb of tense 'ing' or 'past'. If this comma is not the closing comma of an appositive, this construction invariably flags a reduced relative clause, e. g. '... filed past the coffin of Major Roberto d'Aubuisson, **regarded** as the architect of El Salvador's notorious right-wing death squads'. For a system which cannot handle reduced relative clauses (such as FUNES) this construction is very helpful in that it explicitly flags them. In FUNES the appropriate relative pronoun (and 'was' in the case of a past tense verb) are appended to the list of unparsed words. This will enable the relative clause to be analysed at the appropriate time.

- The NW is a relative pronoun, a verb group member or a preposition. In all cases this signals that an appositive will not follow.

- As was shown above, it is possible to have a NP follow the comma and yet not be an appositive. This can also occur if the sentence started with a link word, e. g. 'Although the BJP is the largest single party, Gujarat is governed by ... '. Again our solution is heuristic. If the sentence did begin with a link word then a verb must not occur within three words of the comma, unless another comma occurs first.

In the FUNES system, upon detection of a comma all of the above conditions are checked for, and if none holds then the NP parser is called recursively to parse the appositive NP. When this has been parsed it must be correctly attached. This issue is considered next.

### The Attachment of an Appositive

If an appositive NP immediately follows a subject or object NP then its attachment poses no problem as it clearly attaches to that NP. However if the appositive should follow a noun-attached PP then there is attachment ambiguity — the appositive NP could attach to the PP or to the preceding NP. This is most common when an 'of PP' precedes the appositive. For example in the sentence

8.6 'Soviet soldiers patrolled the streets of Tiraspol, centre of the self-proclaimed Dnestr Republic' ,

the referent of the appositive is the PP 'of Tiraspol'. However in

8.7 'The President of the Ivory Coast, Felix Hophouet-Boigny' ,

the referent is the NP 'the President'.

Various heuristics can be used to aid in the attachment decision. A global syntactic constraint exists which says an appositive cannot attach to a PP that does not immediately precede it. This helps resolve problems when an appositive follows a string of PP's. The only possible attachment points are the main NP and the directly preceding PP. To choose between these a single heuristic is commonly all that is needed as the great majority of ambiguous Cases involve personal names appositives. The heuristic is:

```
if the appositive is a human name
then attach to the target phrase whose head noun is a role word
else if the appositive is a number
then attach to the target phrase which is a personal PN
```

Should this not apply then the heuristics shown in Table 8.2 are used. In all of these heuristics the main NP is checked first, and, if it fulfills the criteria, it is selected. This sidesteps the issue of choosing between two similar candidates. The heuristics reflect the fact that the most common patterns for appositives are to have a PN and a KW-NP, or failing this a PN and a non-PN NP. Compatibility can be assessed by inspection of the Knowledge Base to see which pairs share a superordinate semantic category. The heuristic for handling 'in PP's' attaches to the main NP as an 'in PP' invariably introduces a place PN, and these are rarely described by a following appositive.

When the main NP is a PN and the appositive carries the descriptive NP, the attachment is usually simpler, as PN's are less often followed by PP's. However, both personal PN's and corp PN's can be followed by place names. In this case the attachment is to the main NP. When a personal PN is followed by a corp PN the attachment is genuinely ambiguous. Compatibility can be used if an accompanying KW is present, as this will have established semantic categories for the PN's. For example in the phrase

8.8 'Peter Wilson of SG Securities, the global insurance company, '

'Peter Wilson' will have been classified as a 'human name' (by the default personal PN heuristic describe above). 'SG Securities' will have the semantic category of its head noun, i. e. corp. When compatibility is assessed 'human' will not be compatible with 'corp' (the

```
if the appositive is some other category of PN
   then attach to the target phrase whose head noun is a KW
   else attach to the target phrase which is compatible
   else if the preposition is 'in'
             then attach to main NP
             else attach to PP
else if appositive not a PN
          then if one phrase is a PN and the other is not attach to the PN
          else if PP is a place PN
                    then attach to main NP
          else attach to target phrase which is compatible
          else select main NP
```

Table 8.2: Attachment Heuristics for Appositives

category of 'company'), but 'corp' (the category of 'Securities') will be. Therefore the appositive will be correctly attached to 'SG Securities'. Had the phrase been (the much less likely)

**8.9** 'Peter Wilson of SG Securities, a former Kleinwort executive, '

then 'human' would be compatible with 'role' (the category of 'executive'), and so the appositive would be attached to 'Peter Wilson'.

The attachment of appositives is an area that has not received anything like the consideration given to PP attachment in the Computational Linguistic literature. In fact, the author has not been able to find a single paper dealing with the subject in depth. Yet, it is an important task, just as important as the handling of PP attachment in the analysis of news text. The importance of correct handling of apposition in the MUC-3 [50] task was pointed out by Chinchor [36]. It is obvious that all the systems described in [50] must be handling apposition in some way, but unfortunately no description is given as to how this is achieved. Chinchor's survey simply shows that systems performed much better on the easier appositives than on the more complex ones. Recall for easy appositive phrases averaged around 50%, while for the more complex phrases it dropped to around 15%.

The heuristics described above have performed extremely well in correctly attaching appositives. All of the cases in the FUNES Development Corpus are correctly handled. The majority of simple cases encountered in the test corpus were also handled correctly. However, as with the MUC-3 experiment, performance on more complex examples, especially those involving conjunction, decreased. This is described further in chapter 11. Appendix J contains many examples of the processing of personal PN's, and various types of apposition, taken from the FUNES Development and Test corpora.

So far, we have only discussed cases of single appositives. There are various other patterns in which appositives can occur which are more complex. These all involve a sequence of appositives, but differ as to how the NP's relate to one another. They can all be parsed in the same manner, the recursive calling of the NP parser to parse an appositive means that should the first appositive contain a second appositive the NP parser will be called again and so on.

- Address Type Appositives, e. g. 'Malton Drive, Fenham, Newcastle.' It is common to see these address type appositives occur in an 'of PP' to give a person's address (usually in court news items), e. g. 'Kevin Jerrett, of Edgecombe Park Road, Plymouth'. In the FUNES system, the NP parser would be called three times to parse the first example, on each occasion the syntactic Level would be incremented (see chapter 3 and appendix F). The appositives would be sent to the semantic analyser in reverse order, e. g. first 'Fenham/Newcastle', and then 'Malton Drive/Fenham'. In both cases a 'superpart' link would be returned.

- Multi-appositives. This refers to the case when more than one appositive describes a single person (not with conjunction which will be described below). There are two distinctive ways in which this can occur. The first is 'Descriptive NP, PN, Descriptive NP' e. g. 'The prosecutor, Nikos Katsaros, a New Democracy MP, '. These can be handled in the same way as address types, by the recursive application of the NP parser.

  The second, and more common pattern for a multi-appositive is in the form 'PN, Age, Descriptive NP', e. g. 'Mr Peter Crockett, 54, Staffordshire's deputy director of social services' . In this case passing the last two NP's to the semantic analyser would result in nonsense. If the main NP (54) is a number, the preceding NP (the PN) should be retrieved and sent to the semantic analyser along with the second descriptive NP. Upon completion it will again be sent to the semantic analyser with the age NP.

- Nested Appositives. This is the term used to describe a sequence of appositive NP's, where each describes the preceding PP, e. g. 'Alan Sugar, the chairman of Amstrad, the electrical retailer'. This pattern, like those above, is easily handled by the recursive structure of the NP parser.

This completes the account of the syntactic processing of appositives. The handling of their semantics was described in the last chapter. We move on here to discuss the problems presented to appositive analysis by NP conjunction.

### 8.3.4 Conjunction/Apposition Differentiation

Our focus on conjunction processing in this research has been NP-NP conjunction. We shall not consider simple cases involving only two NP's (e. g. 'the wind and the rain'), as they present few complications for processing, and little interest for PN analysis. Here we are concerned with the problems of apposition/conjunction interaction. This presents itself in several forms, some of which are outlined below:

1. Conjoined Appositive NP's describing a single PN, e. g. 'Abdel-Rahim Ahmed, leader of the Arab Liberation Front and a member of the PLO's Executive Committee, '

2. Conjoined PN's described by a single appositive NP, e. g. 'two other former First Jersey brokers, Arthur Basmajan and Dominic Padula,'

3. A conjunction of PN/Appositive pairs, e. g. 'The former vice-president, Jaramogi Oginga Odinga, and leading human rights lawyer, Paul Muite, '

4. Conjunction within a single appositive NP, e. g. 'Michael Heseltine, the minister for Trade and Industry'

5. A simple conjunct list, e. g. 'Peter Smith, John Andrews and Tony Anderson' or 'theft, deception and burglary'.

6. Conjunction of both PN and Appositive NP, e. g. 'The Prime Ministers of India and Pakistan, Narasimha Rao and Nawaz Sharif, '

This probably does not exhaust all the possibilities of interaction, but it gives a flavour of the sort of problems that can appear. It should be stressed that such constructions are not common, or at least not nearly as common as the simpler PN/appositive patterns. However the fact that they do occur means that some solutions must be proposed to handle them.

The problem is more easily discussed by being split into several subproblems. The first of these concerns the detection of the conjunction, and the presence or absence of more than one preceding NP. The second concerns the analysis of these preceding NP's — are they simple conjuncts or appositive/PN pairs. The third problem concerns the analysis of the conjoined constituent — is it separate from the preceding NP or PN, or linked to both this and the preceding appositive. Finally we must consider the problem of an appositive following a conjoined NP — does it just refer to the NP it follows, or does it describe several of the preceding NP's. We shall look at each of these in turn:

### Detection of Complex Conjunction

By complex conjunction we mean a conjunction following more than one preceding NP. In a system utilising registers to store NP's this is easily detected by checking the contents of the existing registers. In a system that handles apposition the 2nd NP in a conjoined string of NP's may initially be mistaken for an appositive. This is the case in FUNES. This error (which is rectified upon detection of the final conjunction) provides a simple way of detecting complex conjunction, by checking if any appositive NP's have been parsed.

If no appositives have been detected, it means we have a simple 'NP and NP' construction. The main parse can continue with the analysis by recursively calling the NP parser. If an appositive has been parsed, it means that we have a 'conjunction string'. In this case the previous NP's must be reviewed in an attempt to decide if they are unrelated conjuncts, or PN/appositive pairs. This is discussed next.

### Analysis of the Relationship of Preceding NP's

For the sake of simplicity we only consider here the situation of two previous NP's (although FUNES has handled up to five previous NP's). We have produced a case-based heuristic to decide if the preceding NP's are an apposition pair or not. It examines the letter case of the head noun of each NP. If all the previous NP's have capitalised head nouns, or lower case head nouns, then it is assumed that they form a simple unconnected list of NP's. In this case the parse can continue as in the simple case of 'NP and NP', with a recursive call to the NP parser. If, however, the two previous NP's have different case head nouns, then it is assumed that we have a 'PN/descriptive NP' pair.

In this case we must now decide if the conjoined NP is part of the preceding NP or totally unrelated.

### Analysis of the Relationship of the Conjoined NP to Preceding NP's

A conjoined NP that follows a string of NP's can either form part of a conjoined appositive description (e. g. 'X, the sales director and the marketing manager') or introduce a totally new entity (e. g. 'X, the sales director, and the marketing manager' ). In addition if the conjoined NP's are PN's, they could both be described by a previous descriptive NP (e. g. 'the sales directors, X and Y,) or the conjoined PN could be a totally new entity (e. g. 'the sales director, X, and Y...').

The heuristic used to determine if the conjoined NP refers to some totally separate entity is a simple one — the presence or absence of a comma before the word 'and'. If there is a comma it is assumed that the conjoined NP is beginning a new description (which is likely to also contain an apposition). For example in the phrase 'Peter Miles, the sales director, and the marketing manager... ', we will almost always see a comma after the noun 'sales director' , otherwise it would be unclear whether Peter Miles was both the sales director and marketing manager, or the marketing manager was someone else. In the FUNES system the semantic analyser is called at this point to analyse the first PN/descriptive NP pair, so that the appositive can be effectively removed. Upon completion the NP parser is called to parse the conjoined NP.

If there was no comma, the conjoined NP is assumed to be part of the preceding construction and it can be parsed immediately. It will be held at the same Level as the preceding NP, so in the semantic stage they will both be referred to the NP that preceded the comma. This would be the case if there were no comma in the above example, e. g. 'Peter Miles, the sales director and marketing manager... '. Here it is clear that Peter Miles is indeed both the sales director and marketing manager.

This comma-based heuristic is not totally reliable, but it has the advantage of being extremely easy to check for. There are other features that indicate how the conjoined NP should be attached but these are not 'local' and so are more difficult to utilise. For instance, if the conjoined NP is itself followed by an appositive then this clearly indicates that it cannot form part of any appositive NP referring to a previous NP. In the above example this would be the case if we had 'Peter Miles, the sales director, and the marketing manager, Mark Mitchelson...'. However the presence of a following appositive cannot be detected until the conjoined NP has itself been analysed. Moreover, if such a construction were present it would be very unusual not to have a comma before the conjoined NP. Another feature that could be used is the letter case of the conjoined NP. If this is different from that of the preceding NP then this also indicates that it is a separate entity, e. g. 'Peter Miles, the sales director and Mark Mitchelson ... '. Again it is unlikely to see such a construction without a separating comma (although we have shown it as such here). Finally, the presence of a determiner in the conjoined NP could also help in the decision. If the conjoined NP has no determiner then it is more likely that it also describes the preceding PN.

### Analysis of an Appositive following a Conjoined NP

If the conjoined NP is itself followed by an appositive this presents further difficulties, in that it may describe just the preceding NP or several preceding NP's. It will only describe several preceding NP's if they are all of the same letter case, and is only likely if they are in fact all PN's. For example it is possible to have 'the artist, sculptor, and architect, James Flavell...' , but it has not been seen. More common are cases such as 'Peter Miles and Mark Mitchelson, the company directors...'. The rarity of this type of construction following a string of NP's has meant that we have not attempted to analyse it. [1].

[1] This construction is more common in the Wall Street Journal corpus, which was used in the final evaluation of FUNES. Dealing with business and legal matters, this does utilise the construction, with strings of PN followed by a descriptive NP which refers to them all. FUNES lack of ability to accurately analyse this was one of the reasons for poorer performance on this type of text

When this type of construction follows a 'NP and NP' pair (e. g. 'Peter Miles and Mark Mitchelson, the company directors, ' or 'The author and doctor, Tony Harris' ) in FUNES we can only apply the appositive to the second conjunct. This is because at the end of each call to the NP parser, if there is a PN/appositive pair it will go straight to the semantic analyser. When the parse that analysed 'Mark Mitchelson' is completed, there is nothing to prevent the calling of the semantic analyser at this point, which will only analyse 'Mark Mitchelson' and 'company director'. Upon completion the appositive will then be removed and can never be applied to 'Peter Miles'. [2] The use of overly local heuristics to prevent the calling of the semantic analyser until the completion of the first NP parse is not sufficient in all cases. For instance if we had a phrase such as '... who criticised the police and the doctor, Tony Harris,' then we would want to call the semantic analyser before the completion of the whole NP parser, as the appositive PN is only described by the second NP 'the doctor'. Where the appositive NP contains the PN, as in the previous example, there is no way of telling whether both the previous NP's describe it, or only the directly preceding NP. Where the appositive contains a descriptive NP, the number of the head noun could be utilised. If it is plural, this indicates that it describes both the previous PN's.

So far the cases we have examined have covered most of the initial set of examples we introduced to illustrate the problems of conjunction/apposition interaction. The fourth example ('Michael Heseltine, the Minister for Trade and Industry'), in which the role description contains a conjunction, is handled by the passing of the preceding preposition to the recursively called NP parser. Thus, in this example, we would essentially be parsing 'Michael Heseltine, the minister for Trade and for Industry'. The two PP's would be attached to 'minister' and both would be analysed in the semantic stage to give a 'Field' case. (We discuss the topic of post-PP conjunctions more in the next chapter). Example such as the final one go beyond the analysis attempted here and can not be handled by FUNES at present.

There appears to be very little work describing these issues in any depth. Rau has referred to these problems in [137], but gives no detailed solutions. Agarwal and Boggess [3] mention the problem of long conjunction strings, and describe how this problem is worsened by the fact that some of the component NP's may be appositives. However, they propose no solutions. It will be interesting to observe how this problem is tackled in MUC-5, as the text that is intented to be used is the Wall Street Journal corpus, which abounds in this type of construction.

We have now completed our discussion of issues arising from the syntactic analysis of personal PN's. Next we describe issues in semantic analysis.

## 8.4 The Semantic Analysis of Personal PN's

In this section we will look in more detail at the topic of differentia information and its analysis. Differentia information occurs as adjectives, noun complements and noun attached PP's.

In chapter 7 we described the analysis of genus and other differentia information as it occurs in the appositive NP. This process is fairly straightforward, and raises few problems. All that is being carried out is the semantic interpretation of the 'Identity' sense of the verb 'be' (see [85] for a discussion of the three senses of 'be' in English). Where we have

[2] FUNES can, however, correctly analyse cases where the conjunction occurs within the appositive NP (e. g. 'Tony Harris, the author and doctor,' or 'The company directors, Peter Miles and Mark Mitchelson,').

conjoined descriptions, or conjoined PN's, each of the conjuncts is applied to the single NP it is apposed to. Thus from the phrase

**8.10** 'Tony Harris, the author and doctor, '

we must effectively analyse:

**8.11** 'Tony Harris, the author,' **and**
'Tony Harris, the doctor,'

Once the relationship between the PN and the appositive NP is determined, semantic interpretation of each separate NP can then be carried out. We look at this below.

### 8.4.1 Analysis of Pre-Nominal Complements

Pre-nominal complements are adjectives and noun complements. As concerns the analysis of personal PN's, these can give rise to the differentia information origin, property, age, field or works_for. We first look at adjectives.

#### Analysis of Adjectives

Adjectives basically give information on particular properties of the noun they qualify. In most grammatical texts it is customary to see examples with adjectives describing personal appearance or manner. In news text adjectives are rarely used in this way, the most common use is to qualify role descriptions, e. g. 'the **former** X'. The default case for adjective-noun relationships is 'property'.

In FUNES, origin PN's are processed as adjectives so we will discuss these here. They are revealed by their semantic category and serve to give country or region of origin of the head noun they qualify. If the headnoun is in an appositive relationship with a personal PN then this origin information equally applies to that personal PN. So from the phrase

**8.12** 'Chadli Benjedid, the Algerian President, '

the origin information 'Algerian' just as much applies to 'Chadli Benjedid' as it does to 'President'. This is not the case with 'property' adjectives such as 'former'. If we had a phrase 'Chadli Benjedid, the former Algerian President,' then 'former' will only apply to 'President' and not 'Chadli Benjedid'.

Noun complements of semantic category 'abstract' are also processed as adjectives in FUNES. [3] Nouns of this category tend to give field information when used with personal PN's, i. e. information which describes the area of their job, such as 'finance minister' and 'transport secretary'.

Information on age is more often given in an appositive. However it can be given in an adjectival construction qualifying a role noun, which is, in turn, in apposition to a personal PN, e. g.

**8.13** 'Michael Huston, a 41-year-old security guard'

Like origin information this can be applied directly to the PN which the head noun is apposed to.

[3] this is because noun complements that are not capitalised, and not of semantic category 'role' or 'corp', are added to the ADJ slot in the NP parser. This is so the analysis of the Ncomp slot can more reliably detect corp PN's.

### Analysis of Noun Complements

The majority of work in computational linguistics looking at noun complements has been concerned with the detection and interpretation of compound nouns (see [119, 118, 108, 59, 160]). When looking at PN's in news text, and personal PN's in particular, the types of noun complement one finds tend to be very different.

The most common form for noun complements in this domain is that of corp PN's, especially where the head noun is a personal PN or a role noun. In FUNES we have created a special corp PN analyser which detects corp PN's occurring in this fashion. This is described in the next chapter. If the head noun is a role noun or a personal PN then the case relationship between this PN and the corp PN contained in the noun complements is 'Works_for'. In the NP

8.14 'Heinz group chairman Tony O'Reilly'

we deduce that 'Tony O'Reilly' works for the Heinz group (and also that he is the chairman). This information is derived in FUNES, and entered into the lattr DB as a works_for triple — (oreilly, works_for, heinz).

Noun Complements can be used to convey a variety of other information, but for personal PN's 'Works_for' information is by far the most common. Subsequent chapters will discuss the other types of information that occur with different categories of head noun.

As with adjectives, any noun complements of semantic category 'abstract' will lead to 'field' triples, as these will be carrying field information. The default case returned for any noun complements that have not been analysed as 'works_for' or 'field' is 'property'.

In FUNES, at the end of the noun complement analysis, the Case frames derived for any adjectives and noun complements are joined together to form a pre-nominal complements Case Frame (as explained in chapter 3). This is in turn joined to the Case Frame for the head noun. So the NP 'Sri Lankan Tamil rebel leader Vellupilai Prabhakaran denied ...' would lead to a Case Frame:

8.15 '[agent(prabhakaran, role(leader), property(tamil,rebel), origin(sri lanka))]'

## 8.4.2 Analysis of Post-Nominal Complements

Post-nominal complements are noun-attached PP's. The PP cases that contribute knowledge on personal PN's are origin, field, works_for, related_to, associate and creator. In FUNES the case labels for PP's are derived by the PP modules (as explained in chapter 4). The precise circumstances under which each of these case labels is returned are fully specified in appendix H. Below we summarise the contexts which indicate each case:

- Works_for case: This is returned when the noun attached PP's form a corp PN, e. g. 'Lucy Fisher, senior vice president of Warner Bros'. The analyser used to detect this is described in the next chapter.

- Origin case: This is is returned by the preposition 'of' or 'from' plus a place name, e. g. 'the President of Japan', 'a tourist from Canada'.

- Field case: This is returned from PP's with 'on' and 'for', in which the head noun is of semantic category 'abstract' or 'event', or the preceding noun's semantic category is 'role' (e. g. 'an expert on ceteology'). There are many esoteric 'field-type' nouns which may be unknown to an NLP system (even one with a large lexicon). Utilising the pattern 'role-noun on X' to indicate the Field case means that this case can still be detected even if the person's field is unknown to the system.

- Relàted_to case: This is indicated when an 'of PP' is attached to a noun of semantic category 'relative', e. g. 'the brother of X'.
- Assoc case: This is indicated when an 'of PP' is attached to a noun of semantic category 'assoc' (e. g. 'a colleague of', 'a friend of'). The Assoc case implies that the relationship is two-way, so if we encounter a phrase such as 'X is a close friend of Y' we can deduce (X,assoc,Y) and (Y,assoc,X).
- Product case: this has been rarely encountered, and we have only identified the pattern 'creator-noun of X' to indicate it, e. g. 'the author of Our Mutual Friend', 'the painter of the Annunciation'.

In FUNES, after the PP modules have returned a case label, it is examined, and if it proves to be one of the cases which carry the differentia information described in chapter 6, a corresponding entry is made into the lattr DB. The PN is used in this entry, as opposed to the preceding noun. So, given the phrase

**8.16** 'Franco Malfatti, leading member of Italy's Christian Democrats',

instead of entering ('member, works_for, [christian, democrats]'), FUNES will enter ('malfatti, works_for, [christian, democrats]').

The preposition 'as' presents particular problems since it can occur in many different ways, e. g. as a preposition, as a conjunction, or in various collocations like 'as big as'. Some examples are shown below:

...Touvier, 75. He was charged with crimes against humanity for his role as intelligence chief
...will sign agreements on 23 other border disputes as part of a 1984 friendship
...boarder post. Diplomats saw the operation as a warning to fundamentalists
...may have killed as many as 60 people. The avalanche of debris swept through
...the sacking of Richard Humphreys as president of Saatchi & Saatchi Advertising Worldwide
...apparent power struggle to succeed Yitzhak Shamir as prime minister.
...Mr Wuliger will continue as chairman until Oct 11
...country's armed forces as rocket fire pounded the disputed, mainly Armenian,
...he was too involved in the Iran-Contra affair during an earlier stint as deputy director of the CIA.

In the ideal system all of these examples would be handled correctly. We have concentrated our analysis on those cases which convey information on PN's. In particular, 'as-type' PP's convey role information on personal PN's. When the NP to which the PP is attached is a personal PN, or when the PP head noun is a role noun, it indicates an 'acting_as' case, which in turn indicates role information. The main problem is which NP does this role information apply to. Where the NP to which the PP is attached is a personal PN this clearly indicates that the role information applies to that PN. However when the PP is attached to a verb, or when it attaches to some other type of NP, a decision cannot easily be made.

We have utilised a heuristic approach based on the simple criterion of success in applying the role information to the correct referent. This is shown in Table 8.3. Where the 'as PP' is part of a phrasal verb that is in fact conveying a different meaning (e. g. 'resign', 'replace'), then the relevant role information will be altered when the verb meaning is checked, as described below.

```
if PP is NP-attached,
  then if there is a Theme and it is capitalised
         apply info to Theme
         else apply to Preceding Noun
else if PP is VP-attached
       then if there is a Theme apply to Theme
              else apply to Agent
```

Table 8.3: Heuristics for Attaching As-type Prepositional Phrases

### 8.4.3 Analysis of Descriptive Verbs

The majority of information conveyed about personal PN's is conveyed by pre-nominal and post-nominal complements, usually those of a descriptive NP occurring in apposition, but also those surrounding the PN itself. However descriptive information can also be conveyed by verbs. These have not received as much analysis. The quantity of verbal definitions encountered in the Wall Street Journal (WSJ) corpus (see chapter 11) has lead us to conclude that research on verbal definitions is important. McDonald [121] has produced a system that is applied to WSJ text, purely to detect information on job changes, information which is invariably conveyed in VP's. The verbs which we have looked at are shown below:

- Verbs of Job Change
- Verbs of Death
- Verbs of Control/Ownership
- Verbs of Manufacture

The first of these categories is the most complex. It covers the verbs 'replace', 'take over from', 'succeed', 'resign' and 'retire'. The former are the more complex as they describe two job changes — someone who used to have a job and no longer has it, and vice-versa. The common pattern for the first three verbs are ' X will (take over from/replace/succeed) Y as Z'. It is debatable whether the interpretation of the meaning of these verbs, in terms of the job changes they imply, is part of semantic analysis or whether it is an inferred meaning, whose appropriate place is in a higher pragmatics stage.

We feel that interpretation of the job changes is an inference process, and so do not concern ourselves with them in the delivery of the semantic representation of the sentence. However, we do utilise a special verbal definition procedure which interacts with the lattr DB to re-arrange the information entered there on the PN's concerned.

These procedures will transfer the role information (Z) originally entered on Y (due to the 'as-type' PP) to X. In addition they append the term 'property(former)' to the role entry for Y. Any Works_for and Field entries for Y are also transferred to X.

An alternative way of giving this information is to say 'X will take over as Z when Y retires'. In this case, the role information from the 'as PP' will be applied correctly to X, and after the 'when clause' has been processed this role information will also be applied to Y, with the 'property(former)' added. Any Field and Works_for information will be treated in a similar fashion.

Finally the retire/resign verb can just occur on its own (as in 'X will retire as Z'). In this case the 'as PP' will be applied to X, and then the analysis of the retire verb will cause the term 'property(former)' to be appended to the role entry in the lattr DB.

The remaining categories are simpler to analyse. Verbs of semantic category 'end_life' inform the reader that the Theme of the sentence is now dead. Verbs of semantic category 'control' (such as 'run', 'control', and 'own') tell us that the Agent of the sentence is in charge of the Theme of the sentence. An example sentence, and the entry produced for it in the FUNES lattr DB, are shown below:

**8.17** 'Backe Group Inc is run by former CBS Inc President John Backe'
→ (backe, boss_of, [backe, group, inc]).

Verbs of manufacture (such as 'make' and 'produce') tell us that the Agent of the sentence makes the Theme, e. g.

**8.18** 'Gimson Tendercare, which makes stair lifts, ...'
→ ([gimson,tendercare], make,[lift,property(stair)]).

### 8.4.4 The Problem of Noun Phrase Reference

NP reference is a well-studied problem in computational linguistics [84, 173, 122, 7]. A simple approach considers indefinite NP's to be introducing new concepts, and definite NP's to be introducing known concepts that must be referred to previously mentioned concepts.

When dealing with personal PN's we encounter some interesting problems connected with reference. Firstly, the PN itself may have occurred before. The detection of this might be thought to be a simple matter of matching against previous PN's, but, as we explain below, this matching problem is complicated by the 'variant form' problem. The 'Mr Major' referred to in one sentence must be referred to the 'John Major' or 'Prime Minister Major' referred to in the last sentence.

However, it is also possible that the concept referred to by the name 'Mr Major' may already have been referred to in a completely different way, e. g. 'The British Prime Minister'. Some journalists [4] use this pattern of reference, in which they begin a story with a descriptive 'role NP', and then in the next sentence introduce the name of an individual which is co-referential with the previous description. Thus we may see:

**8.19** 'Journalists at Zambia's state broadcasting corporation have threatened a news blackout unless the authorities sack two senior editors alleged to have manipulated information before recent elections. They accuse Wellington Kalwisha and Peter Mwela of biased news presentation in favour of the former President, Kenneth Kaunda.'

Here the 'role NP' is 'journalists', and the two people it describes ('Wellington Kalwisha' and 'Peter Mwela') are not mentioned until the next sentence.

We have adopted as simple a solution to this reference problem as possible. It rests on the fact that the initial description provided is not paired with a PN when it originally occurs — so it is a 'free' description, and a candidate for application to subsequent PN's. The question is, to which PN's can it subsequently be applied ? The simplest solution is

[4] I have found this construction to be very common in the teletext Oracle and Cefax news services, but not so common in printed news text.

to consider only those PN's which occur in the immediately following sentence **with no accompanying description.** This simple solution will work in the majority of cases. It is, however, possible to have a PN occur with an accompanying description, and still be co-referential with the previous description, e. g.

**8.20** 'A leading Soviet economist has predicted that President Yeltsin's economic reforms have at best six months to begin working. Yuri Popov, senior economic advisor at the White House, has predicted ...'

A complete solution to such problems would involve the use of Focus [153, 73] in narrowing down candidate referents. In FUNES, the above solution is implemented by noting all Role keywords that have not been used in describing a PN. Subsequently, when a Personal PN occurs without any description, the previous unused keyword is applied to this undescribed PN. Once this process has been completed, all the unused descriptions are removed. In the first example above, 'journalists' and 'senior editors' are both saved as unused descriptions. When the unknown PNs 'Wellington Kalwisha' and 'Peter Mwela' are encountered, the most recent unused description is applied to them, leading to the creation of 'editor' role slots for both names.

This completes our account of the handling of unknown personal PN's and the problems these PN's can produce for an NLP system processing them. In the final section we consider the problem of dealing with known names, and also mention the process of matching variant form personal names to previous full forms.

## 8.5 The Analysis of Known Personal PN's

This section considers three topics. The first is the processing of descriptive information on known personal names. The last chapter discussed the different approaches to dealing with known names, and concluded that the way in which they should be handled very much depends on the application. Here we show exactly how FUNES handles known personal names. We then consider the detection of a known personal name, from the point of view of locating the correct KB entry. Finally we look at the issue of referral of personal names to their correct previous referent.

### 8.5.1 The Analysis of Differentia Information on Known Personal PN's

When a PN has been detected as known, a decision must be taken as to what to do with the descriptive information that accompanies it. As discussed in the last chapter this can be simply ignored if all one requires is a genus category, or, if one wishes a system to automatically update its KB entries on the basis of new information, it can be retained and compared to existing information. In FUNES we deal with known personal PN's in the same fashion as unknown PN's as concerns the processing of this information. In the name-frame compilation stage the differentia information in the lattr DB is gathered together in the same way as for unknowns, and then compared to the KB entry for that PN.

It is at this point that we must consider which slots indicate a clash if their slot-fills are different, and which slots can permissibly have several entries. For personal PN's the only slots which we do not allow to have more than one entry are Age and Origin. Age should obviously be over-written by the latest entry. Origin slots can be compared in an attempt to refine multiple possible origins to a single origin.

If there is an existing origin slot in the KB entry for a PN then we just retain it, and do nothing else. If however it has more than one filler then we attempt to derive an origin slot for the current occurrence of the PN, and compare the slot-fillers. (If there is no existing origin slot we simply update with the slot-fill from the current occurrence.) This comparison retains any common elements in each of the slots, and discards all the non-common ones. If there are no common elements it simply appends the two slots. Consider an existing entry for say 'Douglas Hurd'. This may have several origin fillers if his origin was not stated explicitly, as he will usually occur in foreign news stories. However across several stories the common element will most likely be 'Britain', and this will eventually be the one retained.

Role slots, Works_for slots, and Boss_of slots are allowed to have multiple fills at present. It is possible for a person to have more than one job, and work for more than company. This is especially true for corporate posts, where it appears to be common (judging from information given in business news and the WSJ corpus) for one individual to have several management posts, e. g.

8.21 William F Healy was named president, chief executive officer and a
director of this financial service partnership

8.22 E.J. Jackson, who previously was president of both Plain Petroleum
and its subsidiary Plains Petroleum Operating Co,

An approach could be taken in which these slots were treated like age slots, and only the most recent fills retained, or different fills could lead to an addition of the term 'property(former)' to the old fills. Assoc, Related_to and Producer can obviously have multiple fills.

All of these slots are compared with a simple list-list comparison. Any item of information in the new name-frame which is also in the old is dropped. At the end of this comparison, the new name-frame is appended to the old, and the new KB entry entered into the KB.

This approach to handling known PN's is at the more knowledge-intensive end of the continuum outlined in the last chapter. As we explained there, the fact descriptive information occurs in a sentence in the same manner as ordinary 'event-type' information, means that this information must be analysed if the sentence is to be correctly analysed, whether any PN's so described are known or not. Given this, the very minor additional work needed to record it and compare it to existing information seems to make such an effort worthwhile, as it enables a KB to adapt its entries to new information.

All of this endeavour regarding known PN's presupposes that the system is able to work out that a PN, once detected, is in fact known. It is to this problem that we now turn. We first consider the location of an initial mention of a name in the KB, whether it be a full form name or a variant form. We then consider the reference problem within a story, i. e. the referral of a subsequent occurrence of a name to a previous one.

### 8.5.2 Locating a known Personal PN in the Knowledge Base

When a personal PN occurs initially in a news story it will most likely be in its full form (i. e. firstname and surname, as in 'Saddam Hussein'). However it is sometimes the case that famous people may occur in an abbreviated form, with just their surname plus a KW, e. g. 'President Bush'. We need to be able to locate both of these forms in the KB in a simple fashion.

The choice is between indexing entries on the full form of the name or on just the surname. In the first case we will then have problems locating people by just their surnames, and in the second we will have problems when there are several entries for the same surname in the KB. In FUNES, due to the fact that only surnames are retained in processing after the detection of a name, we have chosen the second method.

This necessitates a solution to the problem of more than one entry for the same surname. This is relatively easy to achieve, using the firstname and role entries in the lattr DB. If a name has occurred in its full form then we will have a firstname entry, which can be compared to a firstname slot in the KB entry for that surname. If a name has occurred in just its surname form it will not have done so without a KW, in which case the role information conveyed by the KW can be compared to a role slot in the KB.

If a match should not be made we can backtrack and locate any other entries in the KB with the same surname, and repeat the matching process. If a match is still not made after all the relevant entries in the KB have been accessed, then the present entry can be assumed to be a new unknown name.

This matching process would locate the KB entry for 'George Bush' from an occurrence of both 'George Bush' and 'President Bush'. In the first case the triple '(bush, firstname, george)' (entered during the analysis of the parsed Noun Group) would be matched to the 'firstname:george' slot in the KB entry for 'bush'. In the second case the triple '(bush, role, president)' would match against the 'role:president' slot in the KB.

An occurrence of 'Peter Bush' or 'Chancellor Bush' would not lead to a match. In the first case the name would not be returned as known, as it would not be in the lexicon. In the second case the KB entry for 'bush' would be located, but the 'role:chancellor' entry in the lattr DB would not match the 'role:president' slot in the KB, so the name would be returned as unknown, and eventually a new KB entry produced.

In theory this might be a risky policy, as it is possible that someone may be referred to with a role KW that is just not known, but they are in fact the person held in the KB, e. g. if George Bush were the chancellor of a university it is imaginable that he may occasionally be referred to as 'Chancellor Bush'. However, in practice this will rarely, if at all, happen. People are only referred to with a 'role_KW PN' pattern if that role KW is virtually synonymous with their name. It is almost as if the role word has become a substitute firstname. If we did encounter someone referred to as 'Chancellor Bush' it is more likely that this is someone else, rather than a lesser known occupation of the president. If a news story wished to convey the fact that George Bush is in fact a chancellor it would be most likely to do it via an appositive, or sentential description.

We must also consider the problem of matching people who are introduced by a role KW plus their firstname, as can be the case for non-Western names, e. g. 'President Saddam'. To cope with these cases a link entry is made for the firstname to link it to the full KB entry. So when entering 'Saddam Hussein', as well as an entry of the form 'kbase(hussein): Slots', an entry of the form 'kbase(saddam):hussein' would also be made. When a known name is being located in the KB, this link entry is first checked, and if one is found then the crossreference term (the surname) is used to locate the full KB entry. If we encountered a reference to 'President Saddam', we would look up 'kbase(saddam)' and find the entry 'hussein'. Looking up 'kbase(hussein)' would then give us the desired information, which would then be checked as above.

As well as the problem of locating the correct entry in the KB for a full or abbreviated name form, we must also consider reference within a story when a name occurs more than once. This is described below.

### 8.5.3 Personal PN Reference

For the correct analysis of a news story we need to realise that the 'George Bush' referred to in one sentence is the same individual as the 'President Bush' referred to in another sentence. We must also match the same name when it occurs with all middle names or initials, or with only some or no middle names or initials.

The simplest way to handle this problem is to only retain the surname. Then, when the name occurs again and is in a slightly different form, the match is not hampered by this difference. If a name should initially occur in its full form, and then subsequently as a firstname (e. g. Saddam Hussein ... President Saddam), this approach will not work however. If the initial match fails, then the fullname information can be retrieved, and the new occurrence matched to the first word of this fullname entry.

This simple procedure has coped with all cases of personal PN reference encountered in the development and testing of the FUNES system. The only problem for such an approach is when two individuals with the same surname occur in the same story. This is uncommon, but is still handled in FUNES. When a personal PN occurs and an entry is found in the lattr DB with the same surname, then the new name is returned as surname(1). It is handled in this form throughout processing, until the name-frame compilation stage, where it is output in its correct form.

## 8.6 Summary

In this chapter we have discussed the problems which personal PN's present for computational analysis. This discussion has been structured around the FUNES system itself, and we have illustrated each problem with the solution produced for it in FUNES.

We have shown that the correct processing of a text containing many personal PN's, as news text does, demands a multi-stage approach. Although it may be tempting to try and deal with PN's in an initial 'PN pre-processing' stage so that they can be identified and labelled as early as possible, we feel that such an approach is doomed to failure, due to the great complexities of appositive attachment and conjunction/apposition interaction. Such problems can only be overcome by detailed syntactic and semantic analysis. A more realistic, and more successful, approach treats PN's at every stage of analysis, as each stage of analysis contributes something more to a definition.

In addition, the fact that personal PN's are described **within** the text means that the mechanisms which are used for the normal processing of a text, will also handle the PN's it contains. In this approach the new lexical and KB entries can be seen to emerge as a 'by-product' of the syntactic and semantic processing of a sentence. Thus, although the complete analysis of PN's does require additional work, this work is absolutely necessary for a correct analysis of the text in which the PN's occur. Moreover much of this processing can be carried out by procedures which also handle non-definitional constructions, e. g. apposition can be handled as a reduced copular sentence, and many differentia slots can be produced by PP modules which analyse all types of PP. The analysis of PN's involves an expansion of standard language processing techniques, not the creation of totally new techniques.

This concept characterises our approach to the handling of all classes of PN described in chapter 6. In the next chapter we move on to a discussion of corporation names, and show how the same mechanisms can be used in their processing.

# Chapter 9

# The Structure and Analysis of Corporation and Legislation Names

## 9.1 Introduction

In this chapter we discuss the categories of corporation and legislation PN's. For convenience we will refer to these simply as corp PN's, meaning both corporation and legislation PN's. When we wish to make it clear we are only talking about one class or the other this will be stated. These classes differ in many ways from the personal PN's described in the last chapter. In particular their own structure is much more complex and varied, as shown in chapter 6. This chapter takes the same form as the last, examining the analysis of corp PN's at each stage of processing — pre-processing, syntactic and semantic. The name-frame compilation stage is identical to that for personal names, and little will be said about it here. The handling of known forms and variant forms is a more interesting topic, and we will examine these in some depth at the end of the chapter. The diagram in figure 9.1 summarises the processing of corp PN's as implemented in the FUNES system.

## 9.2 Morphological Processing

Unlike origin PN's, corp PN's have no word endings that are particularly revealing of their category. As corp PN's are commonly composed entirely of Capitalised Constituents (CC's) they are very likely to reach the syntactic stage with all the constituents known. The problem then is to detect that these constituents form a single PN.

If a corp name contains an unknown Proper Noun (such as 'Boeing', 'Olivetti' etc), then this can only be returned as an unknown noun — no morphological heuristics have been derived for these nouns, and we consider it unlikely that any are derivable. The utilisation of neighbouring words to produce a semantic category for these words is possible, but as a corp KW may be several words away from the Proper Noun component, we consider it preferable to leave the utilisation of KW's until a later stage. The only exception to this is the presence of ampersands. Corp PN's that contain ampersands are most often composed entirely of Proper Nouns, which directly neighbour the ampersand. Therefore these are very helpful in producing a semantic category for unknown neighbouring words, and if an unknown word is followed or preceded by an ampersand it can be classified as

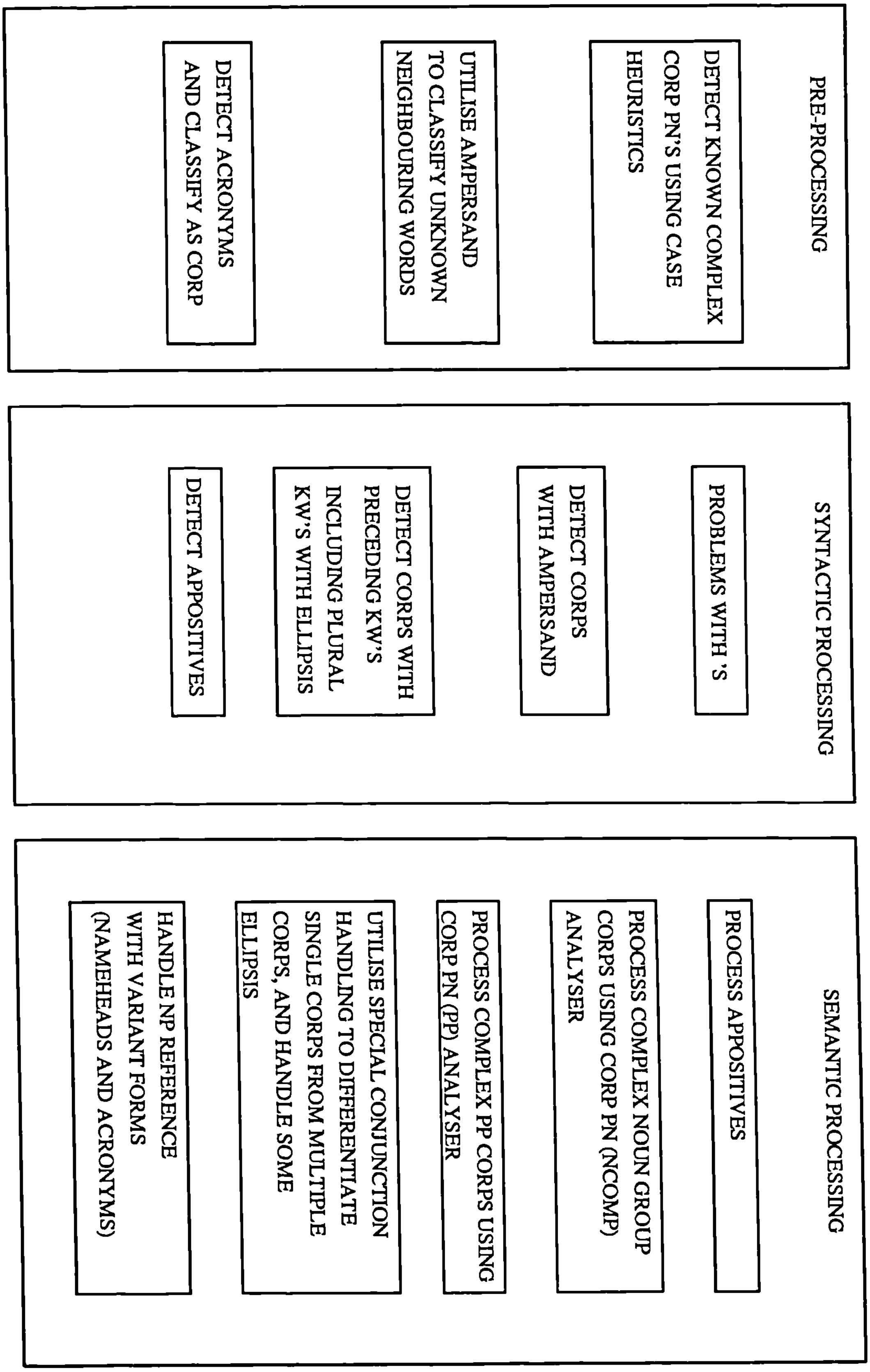

PRE-PROCESSING
DETECT KNOWN COMPLEX CORP PN'S USING CASE HEURISTICS
UTILISE AMPERSAND TO CLASSIFY UNKNOWN NEIGHBOURING WORDS
DETECT ACRONYMS AND CLASSIFY AS CORP
SYNTACTIC PROCESSING
PROBLEMS WITH 'S
DETECT CORPS WITH AMPERSAND
DETECT CORPS WITH PRECEDING KW'S INCLUDING PLURAL KW'S WITH ELLIPSIS
DETECT APPOSITIVES
SEMANTIC PROCESSING
PROCESS APPOSITIVES
PROCESS COMPLEX NOUN GROUP CORPS USING CORP PN (NCOMP) ANALYSER
PROCESS COMPLEX PP CORPS USING CORP PN (PP) ANALYSER
UTILISE SPECIAL CONJUNCTION HANDLING TO DIFFERENTIATE SINGLE CORPS FROM MULTIPLE CORPS, AND HANDLE SOME ELLIPSIS
HANDLE NP REFERENCE WITH VARIANT FORMS (NAMEHEADS AND ACRONYMS)

Figure 9.1: Corporation Proper Name Processing

corp, (e. g. 'AT & T', 'Coopers & Lybrand').

If an unknown word is all upper case, then it is highly likely to be a corp PN (e. g. BBC, EC, BR), and is returned as such in the FUNES system. The accuracy of this heuristic can be improved by the addition to the lexicon of common acronyms which are not corps, e. g. 'MP', 'VAT'.

Material in brackets can also provide information of relevance to corp PN's, in particular an acronym for a full name, or alternatively the full name of an acronym, e. g.

**9.1** 'The Economic Community of West African States (ECOWAS)'
'NASA (the National Aeronautic and Space Administration)'

If a bracket follows a word that is all in upper case, or if a set of brackets contain a word that is all upper case, then the word(s) within brackets can be saved, and utilised at a later stage in the formation of the KB entry for the corp PN they refer to.

It would add greatly to a system's efficiency to be able to detect known corp PN's that comprise many words, or span several syntactic constituents, in the pre-processing stage. We have derived a heuristic approach to this which we will describe in the section on known corp names. If this should fail, the known name will still be detected later in the processing, but it will have undergone extensive and unnecessary processing.

## 9.3 The Syntactic Processing of Corporation PN's

Unlike other categories of PN in the FUNES system, corp names do not undergo much analysis in the syntactic stage, even though they frequently contain KW's. This is for several reasons:

1. As corp PN's (but not legis PN's) can, and do, frequently occur as complements to other PN's, their presence in the syntactic stage cannot easily be detected. To attempt to analyse the corp in the NP 'Ferranti Electronic Corp chairman John Bloom' becomes overly complex at the syntactic stage, where the focus is on the detection of the head PN.

2. Corp PN's present problems as regards the analysis of conjunction. Different corp PN's can be conjoined (as in 'the high street banks Lloyds and Barclays'), but a single PN can also contain a conjunction within it (as in 'the floundering Structural Research and Analysis Corp'). The effective differentiation and analysis of the two different patterns must wait until the semantic stage when any conjoined NP's are available for analysis.

3. Corp PN's often contain PP's (e. g. 'Centre for Information and Language Studies', 'Bill of Rights'). At the stage of syntactic analysis of the initial NP, these will not be available for analysis, and so, like the above case, these must wait until the semantic stage when all constituents are available.

So, for the most part, the analysis of corp PN's takes place in the semantic stage. However there are some problems which must be considered and resolved in the syntactic stage if the PN is to be correctly analysed. The first area we shall look at concerns the problems that an apostrophe s presents in the parsing of a Noun Group.

### 9.3.1 The Problem of Apostrophe S in Parsing Corporation PN's

An apostrophe s (hence 's) normally signals the end of a NP, and so can be used as a signal to halt the Noun Group processor and make a further call to the NP parser. However it can also appear within a corp PN, [1] in which case the Noun Group processor must continue and parse the following nouns as part of the same Noun Group. The occurrence of an 's on a capitalised noun therefore gives rise to ambiguity, in that it can be a normal 's, e. g.

**9.2** 'Australia's Prime Minister',

but it can also be part of a corp PN, e. g.

**9.3** 'the Kurdistan Worker's Party'

In the former case the first NP ends at 'Australia's ' and a second NP parse must begin, the whole construction eventually being analysed as 'prime minister of Australia'. In the latter case we do not wish to analyse 'Kurdistan Worker's Party' as 'the Party of Kurdistan Worker', as, although this might be the meaning conveyed, the PN is a unit which should not be broken up in this way. Initially it might seem as if there is no way to differentiate such cases. However, extensive analysis of news text has been able to identify those words which commonly signal an indivisible corp PN. So the ambiguity can be resolved by inspection of the PN that carries the apostrophe. The following types signal an indivisible PN:

- People's
- Worker's
- Women's
- PN's following the determiner 'St' (for place PN's)
- PN's followed by Inc/Co/Ltd

In the news text we have examined writers appear to be very consistent as regards their use of the apostrophe and its placement. However, *Worker's* does also occur as *Workers'* (and also without the apostrophe). Therefore, no distinction is drawn between a preceding or a following apostrophe — both *Worker's* and *Workers'* signal an indivisible PN. This covers the vast majority of cases that have been observed, but there are some exceptions. Two common ones (that have been added to the lexicon) are 'Jehovah's Witnesses' and 'Reader's Digest'. There is also a potential group of building and place names, such as 'Pike's Peak' and 'Sportsman's Park', which will be discussed in the next chapter.

### 9.3.2 The Analysis of a Noun Group containing Corporation PN's

For reasons outlined above the analysis of most corp PN's is postponed until the semantic stage. Here we describe the different patterns of occurrence of corp names, and discuss what can be done at the syntactic stage:

- A Corp PN occurs as head noun. There are several different cases here, dependent on the particular form of the corp PN :

[1] It can also appear within a place PN, as described in the next chapter. This problem has not, however, been found to occur with any regularity in Legis PN's. An exception to this is the much vaunted 'Citizen's Charter.

- Corp PN is composed entirely of Proper Nouns [2] (e. g. 'he works for Olivetti'). In this case nothing can be done in the syntactic stage as there are no KW's present. In the semantic stage, a following appositive NP or prepositional selectional restriction could be used to create a semantic category.
- Corp PN has a corp KW as head noun (e. g. 'Food Stamp Bill', or 'Olivetti group'). Although here there is a KW that could be utilised to return a semantic classification of corp, we prefer to delay the analysis of such cases until the semantic stage, for reasons outlined above. This decision also permits the use of exactly the same analyser for both these cases and those where a corp PN of exactly the same type occurs as a noun complement to a role or object noun (see below).
- Corp PN has a preceding KW (e. g. 'the environmental group Greenpeace'). This is the only case where some analysis is considered appropriate at the syntactic stage. If the corp PN should be unknown, the KW can be used to create a genus category, in the same manner as role KW's do for personal PN's. If this is not carried out here, then the corp PN will not have a corp genus when it reaches the semantic stage, and so the corp PN analyser employed there will not be able to activate. The KW can also be used to return differentia information, in this case a supertype entry of the form '(PN, isa, KW)', for example '(greenpeace, isa, group)'.

- A Corp PN occurs as a noun complement to another head noun, e. g. 'Olivetti group chairman Carlo De Benedetti'. In this case the corp PN receives no analysis as the focus is on the head noun. Attempting to carry out the detailed analysis which corp PN's demand (due to their potentially complex structure which can provide a lot of differentia information) at the same time as analysing the head PN and its KW is too complex. In addition, as mentioned above, we wish to analyse these cases, and those cases where the corp PN occurs as the head noun, in the same fashion, using the same procedures.

The final construction that receives analysis in the syntactic stage is a corp PN that contains an ampersand (a construction not found with Legis PN's). We examine this below.

### 9.3.3 Corporation PN's containing Ampersands

As was mentioned in section 9.2, an ampersand occurring between two capitalised nouns clearly indicates the presence of a corp PN. It increases efficiency to detect ampersand groups as single corp PN's at the earliest possible stage, to save pointless analysis of their constituent words.

If the ampersand is treated as a capitalised noun, then the whole corp PN will be returned as a single Noun Group. Inspection of the Noun Group to see if it contains an ampersand will then permit the detection and classification of the corp PN. A problem with this overly simple approach is that sometimes an ampersand corp PN may be preceded or followed by descriptive nouns (including a corp KW) that are not part of the PN, e. g.

9.4 'the drinks group White & Mckay'
'the White & Mckay drinks group'

[2] Legis PN's always have a legis KW, so cannot occur in this manner.

These descriptive nouns can be differentiated by the fact that they are not capitalised. Therefore the capitalised nouns can be extracted from the Noun Group to form the PN, and the descriptive nouns used to create differentia information, [3] e. g.

9.5 '([white,&,mckay, isa, [group, property(drink)] )'

In FUNES this is achieved by appending the PN to the descriptive nouns, and letting the KW procedures return the supertype information. Finally the corp PN analyser, which is called in the semantic stage, will handle any remaining nouns, such as the 'drink' above. (Strictly this carries field information, but the analyser is not sufficiently sophisticated to tell this.)

This almost concludes our discussion of the syntactic side of corp PN analysis. We finish by mentioning two issues which were discussed at great length in the last chapter — apposition and conjunction.

Although appositives are not used as frequently with corp PN's as with people, they are still used, especially where the corp PN is composed only of Proper Nouns. The detection and attachment of these is carried out exactly as described in the last section. The semantic analysis is also the same, although different case labels predominate, as we will explain in the next section.

Corp PN's present particular conjunction problems as mentioned already. However the sorts of problems one faces with personal PN's, i. e. the differentiation of apposition and conjunction, and the use of a single appositive NP to describe more than one PN, do not occur with corp PN's. One rarely sees more than two corp names linked, so the syntactic side of conjunction is straightforward.

We now move on to discuss the topic of semantic analysis. Due to the variety of ways in which PN's can occur (e. g. as headnouns and noun complements), and to the variety of syntactic constituents which they can contain, the majority of analysis performed must be carried out in the semantic stage.

## 9.4 The Semantic Analysis of Corp PN's

The focus of this section is on the detection and analysis of corp PN's occurring both as single Noun Groups and as Noun Groups followed by PP's. However, they may also be described via an appositive NP, and we turn to this first.

### 9.4.1 Apposition

As explained above, those corp names composed only of Proper Nouns, or of Proper Nouns plus KW's, are more likely to be given an appositive description, since names composed entirely of Capitalised Constituents tend to be self-describing. Legis PN's have not been found to occur with appositives, although it is possible to imagine such an occurrence. When a corp KW is used to provide a description it conveys supertype information, as well as placing the PN described in the 'corp name' genus. In the FUNES system, the NP

9.6 'Hallwood Group, a Cleveland merchant bank, '

[3] In this example the word 'White' would be held in the lexicon as both a colour and a human name. This is because very common ambiguous personal names are held in the lexicon, as described in chapter 7. As it is capitalised it would be resolved to be a human name. However when it is formed into a corp PN the whole PN is given the semantic category 'corp name', and any component word definitions are simply abandoned

would give rise to an entry of 'corp,name' in the Genus DB and a triple

9.7 '([hallwood,group], isa, [merchant,bank])'

in the lattr DB.

In chapter 7 we explained how corp PN's composed solely of Proper Nouns can be misanalysed as personal PN's in the syntactic stage, (e. g. 'Touche Ross', 'Price Waterhouse'). Should such names be followed (or preceded) by a descriptive appositive, this can be used to over-ride the incorrect genus returned in the syntactic stage. The is achieved by making the process of NP reference and Genus comparison sensitive to this problem, so that any PN that is given a genus of 'corp name', when it already has a genus of 'human name', will only retain the 'corp name' genus.

The differentia slots derived from corp KW apposition tend to be of fairly limited kinds — origin, superpart, and field predominate. These can occur as pre-nominal (adjective or noun), or post-nominal (PP) complements. So, as an example of Origin information we might see:

9.8 'Alleghany, a **New York-based** insurance concern',
'ICI, the largest company **in the UK**',

Superpart information is more commonly conveyed via PP's, e. g.

9.9 'Boston Safe Deposit, **a unit of** Boston Co,'
'Peugeot-Talbot, **the UK subsidiary of** Peugeot of France'

Field information is most common in PP's, but can also be conveyed in noun complements or adjectives, e. g.

9.10 'Candela Laser Corp, **a manufacturer of** high-tech medical devices'
Capcom Financial Services, **a foreign-exchange** dealer,

Appendix H contains a full specification of the derivation of case labels.

The analysis of corp PN apposition is relatively straightforward, as conjunction is not used in the same way as for personal names. The complications in the analysis of corp PN's come from their complex internal structure, which can often contain conjunctions. It is to these problems that we now turn.

### 9.4.2 The Processing of Complex Noun Group Corporation PN's

Unlike personal names, corp PN's can have a complex internal structure. This structure is conveyed either in a single Noun Group or in a Noun Group plus PP(s). We have developed a corp PN analyser for each of these two situations, which detects the corp PN and analyses its structure to return the differentia information it contains. In this section we describe the nature of the single Noun Group corp PN. The presence of this type of PN is signalled in the following circumstances:

1. The head noun of the NP is a role word or a lower case isource word. In this case the corp PN is contained only in the noun complements. This pattern occurs when the corp PN is acting as a noun complement to a role/isource head noun, e. g. 'The Ferranti chairman', 'a Harris poll'. [4]

[4] An object KW can also flag a preceding corp PN. As we shall discuss in the next chapter, there is a problem here in determining whether the object KW is some type of product of the preceding corp (e. g. an Amstrad computer), or actually part of the corp PN (e. g. Mayfair Micros).

2. The head noun is a corp word, a capitalised isource word, a legis word, or can be **viewed** as a corp. [5] This will be the case when the corp PN occurs on its own in a Noun Group, e. g. 'University Patents Incorporated declared ...', '... vetoed the Chrysler bill' .

3. The head noun is a personal name. (This is possible with a Legis PN, as in 'Community Care Act stalwart Virginia Bottomley', and, due to the generality of corp/legis PN handling, will be handled. However, it is not a construction we have observed.) In many such cases the noun complements (if there are any) will contain a corp PN, e. g. 'Virgin Boss Richard Branson'. If this is the case then the final noun complement will be a role noun, as a personal PN cannot follow a corp PN directly (e. g. we do not talk about 'Virgin Richard Branson'). This role noun is not part of the corp PN, and so must be removed before the corp PN analyser is called. These cases are common, and often any optional corp KW is removed, so we see 'Virgin Boss Richard Branson' rather than 'Virgin Group Boss Richard Branson'. It is almost as if the 'role KW + personal PN' constituent is acting as the corp KW, so strong is this pattern in indicating a corp PN.

Of these cases only in 2) is a corp KW actually present. Cases 1) and 3) indicate the potential presence of a corp PN, but the list of words thought to comprise it must be further inspected. If the final noun complement is corp, legis, isource, or can be **viewed** as a corp, then we definitely conclude that a corp PN is present. If the head noun was a role word, a personal PN or an isource word, then, if the preceding noun complement has no semantic category, we also conclude we have a corp PN. This will be the case when we have a corp PN composed simply of one or more Proper Nouns, e. g. 'Sharedata chairman X', 'A Mcaro executive', 'a Granada documentary'.

The first task in the analysis of the remaining noun complements is to ensure that they are in the correct form. The second is to derive the differentia information they give on the PN as a whole. These issues are discussed below.

### The Final Form of a Noun Group Corp PN

One of the main problems in the analysis of corp PN's is the letter case of the last word in the PN (which we call the headnoun, even though it may not actually have been the NP headnoun). In the vast majority of cases where this is not capitalised, it indicates that the headnoun is not actually part of the name, but is merely acting as a descriptive KW, e. g. 'the Olivetti group', 'the Hanson conglomerate'. Alternatively the headnoun could be a subpart of a corp PN carried in the preceding words, e. g. 'United Nations forces', 'an Amtrak subsidiary'. When the headnoun is capitalised it indicates that it really is part of the PN, e. g. 'Azanian Youth Organisation', 'Maxwell Communications'. Therefore, only if the headnoun is capitalised should it be retained as part of the corp PN. [6]

[5] The **view** predicate allows certain words which are not actually of semantic category corp, to be interpreted as corp words when required. Such words are those of semantic category food, substance, object, and abs_product. These frequently occur as head nouns in corp PN's, e. g. 'Lipton Tea', 'British Nuclear Fuels', 'Sun Microsystems', 'Carlton Insurance'. In US news the suffixing of 'Ind', or 'Co' to any company name, regardless of whether it is really a part, considerably facilitates the detection of corp names, and the use of such a **view** predicate would not be necessary. The systems of [137, 104] rely heavily on suffixes, and would therefore perform poorly on names of this sort which do not contain suffixes.

[6] In FUNES this is achieved by examining the letter case of the final noun comp. If it is not capitalised then it is removed and the letter case of the next to last comp examined. This process is repeated until a capitalised word is found. If none of the noun complements are capitalised then we conclude there is no PN present, and no DB entries will occur.

Another problem, referred to in chapter 6, is the inclusion of origin PN's within corp PN's. The origin term could just be giving the origin of the corp, and not be a part of it, or it could be an intrinsic component of the corp PN. As was mentioned in chapter 6, corp KW's can be grouped according to the type of PN that commonly precedes them. Some select very strongly for an origin PN. Examples are 'army', 'navy', or 'city council'. If the headnoun is of this type then the corp PN is not retained for lexical update, due to the lexical explosion which would result. However this type of headnoun can be used to identify unknown origin PN's, which can be retained for lexical update. It would appear that this category of corp KW's could itself be split further, to indicate whether the PN was an origin PN (for army, navy, and air force), or a place PN (for court, city council, etc).

If the headnoun is not of this origin-revealing category, we consider any origin component to form a part of the potential corp PN, as long as the first noun complement is capitalised. If this is not so, then we would not be dealing with a corp PN at all, but some less specific corporate entity, e. g. 'the French contingent', 'the German group'. In FUNES, origin PN's will have been analysed during the adjective processing. Any origin detected there is passed into the corp PN analyser. The same is true of digits, which can also be part of corp PN's, e. g. 'the 8th Armoured Division'.

A final problem concerned with the form of the PN, and attachment of preceding words, is that of appositive description. If the NP which this potential corp PN is part of was described by an appositive, then it is possible that the appositive gave the name of the corp, and the present NP is giving descriptive information. If this is so we do not want this descriptive NP categorised as a corp PN. An example should clarify our meaning. In the phrase

**9.11** 'The Basque terrorist group, Eta, '

the first NP is describing the group named Eta. However in isolation the descriptive NP 'Basque group' will resemble a corp PN (cf. 'Olivetti group', 'Burton group'). So we also check if the headnoun was used as the descriptive noun in an appositive, and if so do not return the potential PN as an actual corp PN.

After the above problems have been resolved and the final form of the corp PN decided upon, it can be looked up in the lexicon to determine if it is known. It is only at this point that it has actually been revealed as a PN, and the correct form derived, and so it is only now that it has become available for lexical look-up.

Having considered problems concerning the form of the corp PN, we now discuss the analysis of the constituent words.

### Analysis of the Noun Group Corp PN's Constituent Words

The most common case labels found in Noun Group corp PN's are 'Superpart', 'Composed_of', 'Field' and 'Name'. For Legis PN's, the 'Composed_of' case does not apply. The 'Name' case is used to describe the Proper Noun component of names like 'Burton Group', which receive a case frame '[group, name(burton)]'. 'Name' is the case returned when all others have been checked for and the remaining component words cannot be found in the lexicon.

The Field case is used to describe the area or type of business of a corp. It it typically indicated by an abstract noun, e. g. 'the House **Agriculture** Committee', 'Metropolitan **Transportation** Authority', '**Coinage** Act'. In the FUNES system we have extended the classes which indicate a Field case, utilising the **view** predicate. This allows any word which can be **viewed** as 'corp' to be returned as a Field case.

The superpart case is the most complex case, as it essentially involves the discovery of a further corp PN among the components of the present corp PN. There are three possible situations in which this can occur:

1. The headnoun is not capitalised and a corp PN is contained in the noun complements, e. g. 'a PLO faction'.

2. The headnoun is capitalised and the whole Noun Group is a corp PN, and the noun complements contain an additional corp PN. This additional corp PN is the superpart of the PN contained within the whole Noun Group. For example in 'Senate Finance Committee' the complements contain the corp PN 'Senate', which is the superpart of the corp PN 'Senate Finance Committee'.

3. The situation is identical to that above, except that the corp PN contained in the noun complements is not the superpart of the whole Noun Group PN, but simply its namehead, e. g. 'Labour Party'. Here the complements contain the corp PN 'Labour', but this is not the superpart of the corp PN 'Labour Party', it is the same thing. This complication stems from the fact that PN's like 'Labour Party' can occur with and without their KW.

These cases are revealed by inspecting the semantic category of the noun complements. If a word of semantic category corp is detected, or a word which is all upper case letters (hence referred to as Corp1), then we have a potential Superpart case. If the headnoun is not capitalised then we have an example like 1) above. Corp1 and all the preceding noun complements are returned as the superpart corp. In the 'PLO' example above we would return the case frame '[faction,superpart(plo)]'. If the headnoun is capitalised we must differentiate types 2) and 3) above. There is no sure way of doing this. A compromise solution is to check the number of noun complements. If there are more than two words, we assume we have a type 2 example, otherwise a type 3. This heuristic is based on the finding that a type 2 case will most often give further information on the subpart corp (e. g. the word 'Finance' in 'Senate Finance Committee', or 'Health' in 'Commons Health Select Committee'), so there will be more than two noun complements. The example 'Senate Finance Committee' would lead to the case frame

**9.12** '[committee, field(finance), superpart(senate)]'.

The 'Composed_of' case is used to describe the components (members) of a corp PN. It is indicated by a noun complement of semantic category 'role', e. g.

**9.13** 'Bartenders Union'
'American White Nationalist Party'
'National Consumers Council'.

Any unknown noun complements can be assumed to be the name element of the corp, as described above, and any remaining words are covered by the default 'Property' case.

In the FUNES system all the case-frames are appended together to obtain a case frame for the whole Noun Group. If the head noun of the Noun Group was a role noun or a personal PN, then a works_for triple is entered into the lattr DB, linking the role/personal PN and the corp PN. In addition a 'staff' entry is also made between the corp PN and the role/personal PN. This is illustrated below:

**9.14** 'National Consumers Council President Brian Mitchelson'
→ (mitchelson, works_for, [national, consumer, council])
→ ([national, consumer, council], staff, president(mitchelson)).

### 9.4.3 The Analysis of Prepositional Phrase Corporation PN's

Corp PN's of this type are composed of an NP with a corp KW headnoun, and one or more attached PP's, e. g.

**9.15** 'Commission for Racial Equality'
'Treaty of Versailles'

Whereas Noun Group corp PN's tend to be business type organisations, PP corp PN's tend to be government or pressure group type organisations. As with Noun Group types, the component parts of the name give information on the nature of the group. Successful analysis must first detect the PN, and then extract the differentia information given in these component parts.

As we stated in the conclusion to the last chapter, it is our feeling that the analysis of PN's should utilise the same mechanisms as the analysis of the text they occur in. In accordance with this principle, the analysis of PP corp PN's in the FUNES system takes place during the analysis of NP-attached PP's, which is grammatically what these names are.

The NP in such a corp is revealed by being capitalised, and by having a corp KW as its headnoun. In the following section we consider the nature and analysis of the initial PP.

#### Analysis of the Initial PP

An initial NP-attached PP and the NP it is attached to, are considered to form a corp PN if:

1) the NP head noun is capitalised, AND
2) the NP head noun is of semantic category corp, legis or event, AND
3) the PP case label is Composed_of, Field, or Origin

(If the head noun is an 'event' noun then we are dealing with an event PN, e. g. 'the Great Fire of London'. These are handled in the same way as corp and legis PN's).

The three cases outlined in point 3 above have been determined to be the only cases encountered for the first PP of a corp PN composed of PP's. This heuristic is based on examination of several hundred PN's of this type, from both US and UK news. The only exception found was 'International Committee of the Red Cross' (which has been added to the lexicon.) It is not possible to derive such a heuristic for any subsequent PP's in the corp PN, nor is it necessary. Any subsequent PP's are automatically considered to form part of the name. (Appendix H contains a full account and specification of how each case is derived).

In the FUNES system if the NP and PP are revealed to form a corp PN, they are entered into a temporary storage buffer, together with the semantic category of the PN (which can be 'corp name', 'legis name', or 'event name', and is derived from the semantic category of the NP headnoun), and the PP case-frame.

Subsequent analysis of the PP itself is problematic in that the PP may itself have the form of a corp PN, e. g.

**9.16** 'National Society of Voluntary Groups'

(This problem also applies to the main NP.) We do not wish to return either of these as corp PN's, and so within the single Noun Group corp PN analyser a check is made that the Noun Group which it is analysing is not contained within the corp PN (PP) buffer. If so the Noun Group is not considered a candidate corp PN.

PP Corp PN's may of course comprise more than one PP. Below we consider the nature of any additional PP's, and the final form of the PN.

### Attachment of Additional PP's and Formation of the Full PN

The sole criterion for inclusion of additional noun-attached PP's in a corp PN is the presence of a partly-formed PN in the corp PN buffer. If there is one, then this PP is automatically added. It is more difficult to assess the contribution of these following PP's in terms of the differentia information they convey. If the case label returned for the PP was 'Composed_of', 'Field', or 'Origin', then this PP is considered to convey relevant differentia information, and its case-frame can be added to the PN buffer. If the case label is not one of these crucial three, then the present PP is more likely to be part of the case-frame for the previous PP, i. e. the present PP is simply a continuation of the differentia information contained in the previous PP. In this case the PP itself is simply added to the buffer, rather than the whole case-frame. For example, in the PN

**9.17** 'National Society for Prevention of Cruelty to Children'

the case-frame for the first PP is '[field(prevention)]'. The case-frame for the second PP is '[of_theme(cruelty)]', and that for the third PP is '[towards(children)]'. Both of these second PP's are actually part of the field information, so the last two PP's are added to the argument of the first case-frame to give '[field([prevention, of, cruelty, to, children])]'. However, in an example like

**9.18** 'Action Group for Re-Development in Liverpool'

both the PP's give informative cases, the first gives field information, and the second origin information. Here therefore both the actual case-frames are joined to give '[field ( [re, development]), origin(liverpool)]'.

The corp PN analyser can afford to assume every noun-attached PP forms part of the corp PN because of the PP-attachment heuristics employed in the syntactic phase. These were constructed with the nature of corp PN's in mind, so for instance 'on PP's' will be attached to a noun only if its semantic category were corp, event or abstract. The only PP's which may 'creep into' the corp PN are temporal PP's. These are kept out by a simple rule that says don't attach PP's whose case label is 'time'.

In FUNES when the analysis of noun-attached PP's is complete, the corp PN buffer is inspected. If it contains a corp PN then several actions are taken. Firstly it is looked up in the lexicon, and if not found it is entered into the unknown and genus DB's (using the semantic category stored in the buffer, one of [corp,name], [legis,name], or [event,name]). Then the bracket register is checked, using the last word in the corp PN as index. This register stores bracketed words, where the conditions described in section 1 are met. If bracketed information is found, an 'abbrev' entry is added to the lattr DB and if the bracketed abbreviation is unknown then it is added to the unknown and genus DB's, and a 'Name' entry added to the lattr DB. Next 'Works_for' and 'Staff' entries are made in the lattr DB, as it is only now that the full corp PN is available. For example in the phrase

**9.19** 'Jalal Tabani, head of the Patriotic Union of Kurdistan, '

we would now be able to enter

9.20 (tabani, works_for, [patriotic, union, of kurdistan]) **and** ([patriotic,union,of kurdistan], staff, chairman(tabani)).

Finally the Case-Frames which have been stored in the corp PN buffer are removed, and used to create corresponding entries in the lattr DB. Again this process has to be delayed until the whole PN is finally detected and formed. So in the above example we would create an entry '([patriotic, union, of, kurdistan], origin, kurdistan)'.

### 9.4.4 Handling of Conjunction within Corporation PN's

Chapter 6 described the complications that can occur when conjunctions occur within corp PN's. These complications stem from the fact that a conjunction can link two different corp PN's, often with ellipsis, (like personal PN's), and in addition, a conjunction can occur within a single corp PN.

As we mentioned above the apposition/conjunction problems encountered with personal PN's do not occur with corp PN's, or if they do it is even less frequently than with personal PN's. We will not dwell on such constructions here due to this. The handling of apposition is a category-independent process, so the discussion contained in the previous chapter applies to all categories of PN. As with the analysis of corp PN's that do not involve conjunction, our description of conjunction cases splits corp PN's into two groups — those occurring with PP's and those without PP's.

In the latter group we can have two different situations in which a conjunction is present.

1. Two separate corp PN's joined by a conjunction. We are not concerned with situations when the two PN's are both known (e. g. 'Hanson and ICI'), or both contain KW's (e. g. 'McDermot International and NL Industries'), as in these cases they are easily separable. The situation we are concerned with is when only one of the PN's contains a KW, the other KW having been ellided. Examples: 'the building societies Halifax and Nationwide', 'Halifax and Nationwide building societies', 'the Scottish and Welsh Communist Parties'.

2. A single corp PN which contains a conjunction. This may be defined by a preceding or following KW, or by an appositive NP. Examples: 'the building society Bradford and Bingley', 'the Competition and Service Bill', 'the quality food group, Marks and Spenser'.

This duality is mirrored in the former group. However here things are simpler, as the number of the initial corp NP is the sole focus of analysis. Thus we can have:

1. A plural corp KW, followed by conjoined PP's, e. g. 'The Departments of Health and Social Security'. The plural KW indicates we are talking about two departments, where the second occurrence of 'department' is ellided.

2. A single corp KW followed by conjoined PP's. e. g. 'The Department of Health and Social Security' (now defunct). Here we have an identical structure except the number of the corp KW.

We begin by looking at conjunction of separate Noun Group corp PN's.

### Conjunction of Separate Noun Group Corp PN's

Here we essentially have two cases to consider — preceding KW and following KW. The analysis of these cases corresponds to that of their non-conjoined counterparts. Thus the former case (which does not apply to Legis PN's) is best handled in the syntactic stage, utilising the presence of a preceding KW to create the correct Genus. When the corp KW is plural we have the same sort of construction as we encountered in the last chapter for personal PN's (e. g. 'Presidents X and Y'). It can be handled by monitoring the number of the KW and passing this information between NP parser calls. In the first example above — 'the building societies Abbey National and Halifax' — when the noun 'Halifax' is analysed, the information contained in the KW 'societies' can be used to create genus and differentia (supertype) classifications.

Where the KW follows the corp PN, analysis is postponed until the semantic stage. The first NP will have no corp KW to reveal it as a corp PN, so no genus or differentia information can be derived. This is not so for the second NP, which will have a corp KW, and for which a corp PN genus can be derived. The number of this KW should also indicate that it applies to the NP to which it was conjoined. [7]

We now look at conjunction within a single corp PN, for non-PP PN's.

### Conjunction within a Single Corp PN

When we have a corp PN containing a conjunction it will be handled in the syntactic stage as two separate NP's (as indeed it is). In the semantic stage we attempt to detect such types and form them into the single PN which they are. The main difficulties for analysis are that one of the NP's will not be defined as a corp PN, and the possibility of conjoining NP's that are indeed separate entities.

The procedure attempts to join the constituents together in the following situations:

1) Preceding KW, e. g. 'the building society Bradford and Bingley'
2) Following KW, e. g. 'the Bradford and Bingley Building Society'
'The National Aeronautics and Space Administration'
3) Following KW in apposition, e. g. 'Bradford and Bingley, the building society,'

In the first case the word 'Bradford' is given a semantic category by the preceding KW, but the word 'Bingley' remains undefined. In the second case the word 'Bingley' will have received a semantic category from the following corp KW, but the word 'Bradford' will be undefined. In the last case 'Bingley' will have received a category from the following appositive NP, 'Bradford' will not.

There are several initial conditions for two NP's to be considered a single corp PN. Firstly (and obviously) they must be conjoined. Secondly, so that the procedure is only applied to PN's, both head nouns must be capitalised. Thirdly, the head noun under consideration must not be plural. This is the crucial step in differentiating a single corp PN from two separate PN's.

Finally one of the head nouns of the two NP's under consideration must have a semantic category of corp or legis. This ensures that we do not attempt to merge non-corp PN's. This is a condition that can be enforced successfully due to the fact that it is extremely uncommon to meet corp PN's in this form without an accompanying description that

[7] In FUNES the application of the plural corp KW to the preceding NP is not yet implemented. Thus only the second NP is analysed correctly as a corp PN.

will have provided a semantic category for one of the component NP's. [8] Where the conjoined PN occurs with no description, neither word will be corp or legis, so a join is not attempted. This is due to the low frequency of corp PN's occurring in this form with no description, and the high possibility that the two conjuncts could be something entirely different, e. g. 'Sarajevo and Dubrovnik', 'Smith and Jones'.

Finally we must rule out the case mentioned above of linking two known corps or two corps which both contain KW's. We must also avoid linking unrelated constituents, just because one of them happens to be a corp PN, e. g. 'President Bush and Congress were for once agreed ...'. The first two categories are ruled out by preventing the join of two corp PN's, or of two NP's that both have corp KW's as headnouns. The final category poses greater problems.

Examination of the news text we possess has shown that conjoined names of this type take two forms (note that here we are talking about the form of the actual names, rather than the nature of the descriptive contexts which the names occur in, such as preceding of following KW, which is what we described above):

1) A Conjunction of Proper Nouns, where the NP following the conjunction is a single Proper Noun, e. g. 'Miers and Carmichael'
2) Conjunctions of Capitalised Constituents, with the NP following the conjunction possessing a KW, and the NP preceding it not,
e. g. 'National Food and Drug Administration'
or 'National and Community Service Act'

Given this fact, two NP's are considered to form a single corp PN, if all the above criteria are fulfilled and :

1) the non corp NP is a single capitalised word, OR
2) the non corp NP is a string of CC's the head noun of which is not a personal PN

This heuristics has been found to correctly process all observed cases. Having now considered the conjunction of complex Noun Group corp PN's at some length, we move on to describe the nature of PP corp PN's involving conjunction.

### 9.4.5 Analysis of PP Corporation PN's involving Conjunction

This type of conjunction poses somewhat different problems from the above. Some of these problems stem from the syntactic problems of parsing conjoined PP's. When a conjoined NP occurs after a PP, the NP may attach to the same NP as the PP or it may begin a totally unrelated NP. Thus we may compare:

**9.21** 'National Society for the Care **and Resettlement** of Offenders' , **and**

**9.22** 'Richard Wilson of London **and John Stevens** of Birmingham'

In 9.21 the conjoined NP is really a conjoined PP. The whole phrase should actually read 'National Society for the Care and for the Resettlement of Offenders'. Both the PP's attach to the main NP. In 9.22 the conjoined NP is not part of the initial NP, and there is no ellipsis.

[8] It is quite common to meet corp names linked by an ampersand with no accompanying description, but in that case the ampersand is providing definite information that we have a corp name. With the word 'and' acting as the link this is not the case, and therefore either a preceding KW or an appositive will commonly occur.

In FUNES we process all cases of PP conjunction as 9.21. When a conjunction NP occurs in this way, the preposition from the previous PP is handed into the recursive call to the NP parser, so that the conjoined NP is actually stored as a PP. This means that it will be stored in the PP register as if it qualified the preceding PP. When we actually have a case like 9.22, the parse will not fail, it will simply be slightly incorrect. The PN's will however be processed correctly. A more advanced system would consider the semantic category of the two conjuncts before deciding on the nature of the conjoined NP's attachment. If the category of the two NP's is the same it is likely that we have a case like 9.21, if not a case like 9.22.

In essence therefore, conjoined PP's are handled as if they were normal PP's. This facilitates the analysis of corp PN's of this type, as the corp PN (PP) analyser can handle them as it handles non-conjunction types. We therefore say no more about single PP corp PN's, but move on to consider a conjunction of different PP corp PN's.

### Semantic Analysis of more than one PP Corp PN

This class can be differentiated from a single PN by examining the number of the corp KW in the initial NP. Thus, while we speak about the 'National **Society** for the Care and Resettlement of Offenders', we talk about the 'The **Departments** of Social Security and Transport'. We have found no examples of Legis PN's being conjoined in this way.

In such cases the initial NP that contains the plural KW is ellided from the conjoined NP. The full version of the second example above is 'The Department of Social Security and the Department of Transport'. In FUNES we detect the plural number of the corp KW from within the corp PN (PP) analyser. During processing of the second and subsequent PP's the number of the initial KW is checked. If it is plural, and the PP is conjoined, then we assume we have a second corp PN. We first locate the ellided corp KW (from the corp PN buffer), and then make a recursive call to the corp PN analyser, which deals with the current PP as if it were the first PP following the corp KW. This will eventually lead to the creation of two corp PN buffers, which will both be processed in the standard way.

We will look at the above example to clarify the process. When the post conjunction NP 'Transport' (which is returned by the parser as a PP — 'of Transport') is analysed, the number of the initial corp KW 'Department' reveals that the PN is in fact two separate PN's. The fact that 'of Transport' is flagged as a conjoined PP indicates that the new corp PN begins here, but of course the corp KW has been ellided. This is retrieved from the corp PN buffer formed for holding the first corp PN, and the corp PN analyser is now called again with the ellided NP (department) and the present PP (of transport). The case frame supplied to this recursive call to the analyser is formed from the case label found in the first corp PN buffer, and the present PP. So here the CF would be '[field(transport)]', formed from attaching the PP 'transport' to the case label 'field' which is the case label for the first PP.

At the end of processing the post-noun PP's, two corp PN buffers will be available:

**9.23** [[department,of,social,security],[corp,name],[field([social,security])]] , **and**
[[department,of,transport],[corp,name],[field([transport])]]

Each of these will eventually be entered into the lexicon and KB.

## 9.5 Handling of Variant Forms

In chapter 6 we described the variety of forms in which a corp PN can occur. In essence a corp PN (this variant form problem does not apply to Legis PN's) can occur in three forms:

1. Its full form, e. g. 'Polly Peck International', 'National Oceanic and Atmospheric Administration'.

2. A shortened form, in which various constituent words have been removed, rightmost first, e. g. 'Polly Peck'.

3. An *acronym* form, in which each of the constituent words has been replaced by its first letter, and prepositions and conjunctions removed, e. g. 'NOAA'.

The variety of forms described in chapter 6 can be reduced to these three alternatives. Unlike personal PN's, with corp PN's we do not have to be concerned with the issue of matching a variant form to an existing KB entry, as corp PN's always occur initially in a story in their full form. It is only on subsequent occurrence that they change. Thus the matching process need only be sensitive to these variations when we carry out NP reference. For the most part we are only concerned with matching unknowns here. If we come across a known PN which is a variant form it will have been mapped to its full form on the occasion on which it was learned, as variant forms do not occur without a previous full form use, and this full form information will be contained in its KB entry.

There is one exception to this. It is possible that a corp will have occurred in a story without a variant form, and so only the full form will have been acquired. On a subsequent occurrence both the full form and a variant form may occur. In this case the full form will be known, but the variant form will not.

In matching unknown NP's to previous NP's, the following approach is adopted. If a total match should fail, the following partial matches are tried:

1) A match of the present noun to previous acronym forms.
2) If this fails an attempt is made to match previous NP's which share the same first word as this NP

The first match is enabled by a procedure which creates an acronym form for any corp PN which occurs in the input text. This procedure simply takes the initial letter of every component word, abandoning any prepositions or conjunctions. For the great majority of PN's this works, although there are obviously exceptions, e. g. 'British Aerospace' abbreviates to 'BAe', 'Grand Metropolitan' abbreviates to 'GrandMet'. Due to these exceptions, the acronym form created is not entered as a name-frame slot, unless there should be a subsequent occurrence of it in the text, which acts as a confirmation of the heuristically-derived form. This approach is in contrast to that taken in [137] which automatically enters the algorithmically-derived acronym forms into the lexicon. Such an approach will lead to incorrect lexical entries.

The second match will work for all of the remaining foreshortening heuristics described in chapter 6. If a match is made, it is preferable to adopt the semantic category of the full form, rather than carry out a comparison. This is because the shortened form is unlikely to contain a KW, and so its semantic category will have been derived from the far vaguer verb or preposition selectional restrictions. The semantic category for the full form is likely to be the more accurate, as it was more likely to have contained a corp KW.

There are two problems with this simple use of matching against previous nouns with the same first word. The first concerns the occurrence of corp subparts — some companies have sub-divisions which share the name of the parent, but add some qualifying information to the end of the name, e. g.

9.24 'Saatchi & Saatchi' **and**
'Saatchi & Saatchi Advertising Worldwide'.

This problem is solved by checking if the present unknown is longer or shorter than the previous occurrence it matches. If it is longer then it is assumed to be a sub-division, if it is shorter a namehead.

The second problem concerns different PN's which share the same first word. This situation occurs very rarely. This is because when we have a shortened form it will be a Proper Noun. (As explained in chapter 6, those corp PN's composed entirely of CC's invariably shorten into acronyms). The chances of having another PN commencing with this Proper Noun which is not its full form are very remote, due to the confusion this would cause a human reader. Moreover when such a match is made the semantic category of the NP recovered must be 'corp name'.

This heuristic approach to matching variant forms has worked well, proving to be relatively simple to implement and handling the majority of examples encountered. In chapter 11 we present the results for an unseen corpus of test stories, which support this claim. Appendix K contains examples of the sort of cases that are noted in news text.

The final section in this chapter looks at the handling of known corp names.

## 9.6 Dealing with Known Corporation PN's

To a large extent the description of handling known personal names applies to corp names as well. Whenever a corp PN is detected, whether it be in the syntactic KW analysis, or the semantic corp PN analysis, it is looked up in the lexicon. If known the name is not entered into the unknown DB. Unlike known personal PN's, known corp PN's do not undergo processing and comparison in the name-frame compilation stage. This is because corp PN's do not receive much additional description in a text beyond that given in their name, or an accompanying appositive. The crucial information for a company tends to be origin and field information, which does not change from text to text. There is thus little opportunity for building up an evolving picture of a corp PN, in the same way as for a personal PN, and FUNES does not attempt to do so. [9].

In the first section of the chapter we mentioned the desirability of detecting known corp PN's at the earliest opportunity, to facilitate and speed up subsequent processing. Even though a PN like

9.25 'Bank of Credit and Commerce International'

will be detected as known (if indeed it is known) by the corp PN analyser, this will only be after considerable work has been expended — the PN having been processed as a NP and two attached PP's. If this could have been detected at an earlier stage, then it would have been processed as a single known NP, leading to much less time and effort being expended on its analysis.

We have developed a procedure which attempts to do just this. After each word is looked up in the lexicon, it is passed to a special procedure which inspects a buffer and the word's case. If the word is capitalised it is appended to whatever is in the buffer, or if there is nothing in there it becomes the first word in the buffer. Additionally, if there are already words in the buffer and the current word is 'the/and/of/for/to/on' then it is added. When

[9] In business news this is not necessarily true, as often information on sales, acquisitions and mergers is given. If a system were able to analyse the verbal constructions used to convey such information, an evolving picture could be created. This appears to be the approach taken in [96, 33]

a word is encountered which is not capitalised nor one of these corp component words, the contents of the buffer are looked up in the lexicon. If found, the constituent words and their definitions which have been added to the list output by the lexical look-up procedure are replaced by a single NP entry, consisting of the buffer contents and the definition found in the lexicon.

This procedure is quite effective, but some further constraints are needed to prevent incorrect constituents being added to a known corp name. If the current word is a known place name or of semantic category 'time' then it is not added, and the buffer is immediately looked up. This is because such words can occur next to a corp PN, and if included in the buffer will then prevent the actual corp name from being found, for example 'Last July Institute of Earth Science members announced ... '. If the preceding word has an apostrophe s then the existing buffer is looked up, and, if the current word is capitalised, a new buffer started.

These facilities make the procedure more effective, while still being relatively cheap to operate. However there are two problems that are not solved. The first is the word 'and' which can link two PN's or occur in a single one. If it is linking two separate PN's then neither will be found. The second problem is the role noun. As explained before, this can occur between a corp PN and a personal PN (e. g. 'Virgin Boss Richard Branson'). As the role word is often capitalised, it will cause the two different PN's to enter the buffer, and so prevent either being found. Exclusion of role words from the buffer would then fail to find those corps PN's that contained them (e. g. 'National Boilermakers Union'). A solution to all these problems would be to look the buffer up after each addition. This would have the benefit that it would find all known names, but at the cost of many extra lexical look-ups. The adoption of this solution would depend, therefore, upon lexical look-up times being negligible.

The effect of this procedure is to detect many known complex corp PN's at the earliest possible stage of processing, and so to increase the efficiency of any system adopting it. The fact that the buffer is always looked up before any of the contents are returned as a single unit, and that place and temporal nouns are not admitted, means that there is no danger of obtaining meaningless false positives, a problem which has plagued some simple PN detectors which rely just on letter case.

## 9.7 Summary

In this chapter we have discussed the problems which corp and legis PN's present for computational processing, and have showed how the FUNES system overcomes many of these problems. Topics we have discussed are:

- Pre-Processing Issues — the detection of complex known corp PN's.
- Syntactic Issues — corp PN's defined by preceding KW's and ampersands, detection of appositive definitions.
- Semantic Issues :
  - analysis of corp PN's defined by following KW's
  - analysis of corp PN's defined by appositives
  - analysis of corp PN's which occur as complements to other types of PN.
  - analysis of corp PN's containing PP's

- differentiation and analysis of single corp PN's containing conjunction from separate corp PN's linked by conjunction.
- referral and acquisition of variant forms

We have described solutions to the majority of these problems, and shown how the FUNES system implements these solutions to enable it to analyse and acquire the majority of corp PN's encountered in news text. We have also discussed those cases which exceed FUNES' capabilities.

In the next chapter we continue the account of PN processing with a description of the remaining categories of PN introduced in chapter 6.

# Chapter 10

# The Structure and Analysis of Place, Object, Information Source and Event Names

## 10.1 Introduction

In this final chapter on the analysis of PN's we look at the remaining four categories, and also briefly consider origin PN's. Much of the material covered in the last two chapters is also of relevance to these categories. It is hoped that the reader is now familiar with the nature of appositive and KW analysis, thus facilitating the briefer description given here. As we made clear in chapter 6, the coverage of Isource and Event PN's within our model is limited, due to the limited nature of their occurrence in news text. This is reflected in the present chapter. Place PN's (and to a lesser extent Object PN's), being more common, have received more investigation and analysis, something which is reflected in the space accorded to them.

We shall look at each category of PN separately, making the now familiar distinction between pre-processing, syntactic and semantic issues. We begin with a short description of the problems of origin PN's. These are not accorded the same status as other classes within our model, due to the lack of within-text descriptions used to describe them. For the most part they simply occur as adjectives or noun complements, usually giving the nationality of people or companies.

## 10.2 The Analysis of Origin PN's

### 10.2.1 Pre-Processing of Origin PN's

In chapter 7 we discussed morphological issues related to origin PN's, and we will not repeat that discussion here.

The only issue not discussed in chapter 7 was that of hyphenated words. Some hyphenated word contain origin PN's, which can often be revealed by the occurrence of certain verbs as the second word in the hyphenated construct.

Therefore, if the first word in a hyphenated word is capitalised, then the second word is inspected to see if it is a verb indicative of an origin PN. The words 'based' and 'born' indicate that the first word is an origin PN, (which in FUNES is entered into the origin DB for use in ascribing origins to those PN's with no origin slot at the end of processing). The words 'sponsored', 'led', 'backed', 'made' and 'registered' also indicate an origin PN, but have not found to be reliable entries in the origin DB. The first three words in fact indicate a 'Works_for' relationship between the following PN and the origin PN. Finally, if the second word is 'speaking' it also indicates an origin PN, specifically a language.

The hyphenated word itself is returned as an unknown adjective and will not be entered into the lexicon. As these compounds can be easily analysed compositionally, addition into the lexicon would only add a great many un-needed words.

### 10.2.2 Syntactic Processing of Origin PN's

Unknown origins receive minimal analysis in this stage. However known origin PN's must be detected and removed from the NG (noun group), to facilitate the application of KW patterns. The removal of any origin PN means that in the analysis of other PN categories a simple letter-case heuristic can be more effectively used to locate the other PN. It also prevents the origin PN being erroneously mis-classified as part of this other PN. The removal of known origin PN's is a straightforward process, as origin PN's always occur at the start of the NG. However there is an exceptional, but relatively rare, case that must also be considered.

This is illusrtated by examples like 'the Patagonian south' or 'the Canadian north'. For such cases to be detected the PN must be a known origin, or an unknown origin that has been revealed by morphology. When this is so the words must be reversed if the case frame for the NP is to be accurate. However the PN cannot be left in its present form, as 'the north Canadian' is incorrect. So the root place-name form must be used when it is transformed into the head noun, so 'Canadian North' becomes 'North Canada'.

The final issue we consider is the contribution origin PN's make to the semantic analysis of a sentence, and the formation of PN lexical and Knowledge Base entries.

### 10.2.3 The Semantic Analysis of Origin PN's

In the semantic stage origin PN's play a part in the contribution of origin differentia information for other types of PN. This process has been partly described in previous chapters. Origin PN's only occur in the Adj slot of an NP — although a PP may contribute origin information it will do so from the occurrence of of a preposition plus a place PN (e. g. 'of Chile', 'from Pakistan'). The different cases that can be returned upon detection of an origin adjective in the Adj slot depend on the nature of the head noun and the nature of the adjective. The process is shown in Table 10.1. The 'Superpart' CF leads to a Superpart slot in the lattr DB, the 'Origin' CF to an Origin slot. People and Corps are often given explicit origins in this manner. However, in many cases they are not given an explicit origin, this being left implicit in the text. An approach to deriving this information is described below.

#### Derivation of Implicit Origin Information

Our use of implicit origin information rests on the fact that often origins are not given explicitly in a text, as they are obvious from the location of the events described in a news story. For example, if the origin of people is not given in a story about India, then it is likely that this is because they are clearly from India. Such origin information can be derived from the origin and place PN's that are present elsewhere in the story. However, some care must be exercised here, as not all such PN's that are mentioned in a story may be directly relevant to the main actors of a story, they may simply be mentioned in passing, or be the location of some other event in the text.

```
if Adj (hence Adj1) is directional
  then if there is an additional origin Adj (hence Adj2)
         then return the CF 'from_loc(Adj2),region(Adj1)'
         else return CF 'region(Adj1)'
else if head noun is a place KW/PN
       then return CF 'superpart(Adj1)'
else if origin Adj is shown as a place PN
       then return CF 'based_in(Adj1)'
else return CF 'origin(Adj1)'
```

Table 10.1: Heuristics for Semantic Analysis of Origin Proper Names

We have identified the following types of information as being the most reliable in indicating origins in this way:

- All known origin PN's
- All unknown origin PN's
- The subject of the sentence, if this is a place PN
- Any word returned from a PP module as origin(X), e. g. 'from' or 'of' plus a place PN

Any such words are considered to contribute implicit origin information. If a PN has no origin information derived for it at the end of processing the story in which it occurred, then an origin slot can be built using those origins which fulfill the above criteria. In FUNES this is achieved by entering all such words into an origin Database, and in the name-frame compilation stage of processing, if a PN has no origin then an origin slot is created from the contents of this database.

This approach frequently leads to greater than one origin being given, but these can be filtered down on subsequent exposure (see chapter 8, section 8.5.1). An example of the utility of this heuristic can be seen in the following story, in which origin information can be derived for 'Paul Touvier', even though it is not given explicitly:

**10.1** 'The Paris Court of Appeal turned down an application for the release of Paul Touvier, 75. He was charged with crimes against humanity for his role as intelligence chief for the pro-Nazi French paramilitary police during the War.'

This heuristic approach has worked reasonably well. The main drawback is in home news, where the 'UK' origin of most actors is not given, but the origin of any foreign actors is. This will then lead to the erroneous creation of foreign origin slots for the British actors in the story. This does not happen as often as it may seem, as home news is mainly about UK events, and so does not mention foreign Origin PN's. However, it does occur sometimes, as shown in the example below:

**10.2** 'Two men were injured and 19 others arrested after a street fight between rival gangs in Bradford, West Yorkshire, early yesterday. The disturbance involved about 50 Asian youths in the city's Horton Grange area.'

This led to a 'superpart' slot of 'asia' for 'Horton Grange' — rather an interesting reading of the story. The entry of place PN's that occur in appositives would be a partial solution to this problem. It might be thought sensible to allow 'in + Place PN' to lead to an origin Database entry, but in practice this has been found to produce too many incorrect entries. Although 'in + place PN' certainly gives the location of events, it does not necessarily give the origin of actors.

This completes our account of origin PN issues. We now move on to consider the remaining four classes of PN, starting with the most important — the place PN.

## 10.3 The Analysis of Place PN's

Place PN's present a wide variety of types, as was shown in chapter 6. As with corp PN's, we find a variant form problem in this category as well, due to the optional attachment of many KW's to their PN constituent. The majority of place PN analysis occurs in the syntactic stage, as it is here that KW analysis takes place. In the semantic stage appositive definitions and the directional patterns described in chapter 6 are analysed. Prepositional selectional restrictions can also also contribute a place Genus to unknown PN's, should they occur in the right context. These are also applied in the semantic stage.

We begin, however, with a discussion of pre-processing issues.

### 10.3.1 Pre-Processing of Place PN's

The morphology of place PN's has not been examined in any great detail as it is not felt to be a consistent indicator of the class to which the PN belongs. In chapter 7 we described how some apparently quite reliable word endings had to be abandoned due to the occurrence of too many counter-examples. Even if the morphology of place PN's were to be analysed convincingly, one would be faced with the problem that morphology would depend on language group, and this can only be known when the PN is already known. [1]

We have found certain 'directional' words to be very useful in classifying following unknown capitalised words as place PN's, however. These words are:

> northern, southern, western, eastern, north-east/west, south-east/west, neighbouring, central, outer

There is one problem in the use of these words, in that certain corp PN's can include directional words of this sort, e. g. 'Eastern Utilities', 'Southern Water'. In the vast majority of these corp names, the directional word will be followed by a corp KW or word that can be VIEWED as a corp KW, and so the heuristic will not apply as the following word will be known. However, it is possible that such a corp PN could contain an unknown name constituent, e. g. 'Eastern Edison Co'. To cope with these cases we insist that the directional word be lower case if the following unknown word (which of course must be upper case) is to be returned as a place PN. Examination of the news corpora available has shown that, in the vast majority of cases, upper case directional words indicate a corp PN, while lower case ones indicate a place PN. All the occurrences of such words were extracted from the on-line AP news corpus, and the only exception to this upper case/

[1] Although Oshika et al. [131] have carried out some work utilising hidden Markov Models to determine the language group of personal PN's.

lower case distinction was 'Northern Ireland'. This is already in the FUNES lexicon. Such a finding further supports the inclusion of country names in the initial lexicon.

We now move on to consider the far more detailed analyses that can be applied in the syntactic stage, due to the variety of place PN and place KW constructions.

### 10.3.2 Syntactic Analysis of Place PN's

The first area we shall look at here concerns the problems caused by the occurrence of a place PN with an apostrophe s.

#### Place PN's with Apostrophe S

In FUNES we have only tackled the problem of 's in place PN's when it occurs in building names that begin with 'St'. These are detected by the parser which does not end the NG parse when they occur (as it would normally do), but continues until there are no more nouns to parse. The place PN contained in the NG is then analysed in the KW procedures as explained below.

There is another group of place PN's that occur with 's, but these are not yet handled in FUNES (they do not, in fact, occur very frequently). However some consideration is in order, should they be found to cause greater problems in the future. The sort of names we have in mind are building and place names, such as 'Pike's Peak', 'Sportsman's Park', 'Caesar's Palace' and 'Bob's Big Boy restaurant'. These would be relatively easy to check for in the NG parse, by checking if the next word is a capitalised noun of semantic category 'loc'. Such an approach would just leave examples of the last type uncovered. Here the 'loc' noun is arbitrarily far removed from the apostrophe 's noun, and is not necessarily capitalised, so checking for this case would not be easy. We would have to consider how far a system should be extended to cope with rare and esoteric examples, given their low frequency. If such cases are very rare, then to have to check all the following nouns upon every occurrence of an 's PN is not justifiable. However, if such cases are noticed more frequently then such an extension would become necessary.

The 's problem is the only problem which place PN's cause in the actual parsing of a NG. Once the NG has been parsed, however, the analysis of any place PN within that group throws up many interesting problems. The first issue we look at is the use of directional nouns in the detection of unknown compound place PN's.

#### Compound Place PN's

The exhaustive listing of all the world's countries and states in an initial lexicon is a herculean task. In FUNES we have listed many of the more common countries, but this is all. We have found that the use of directional words, such as those described above, greatly facilitates the detection of place PN's, and can thus compensate for an impoverished lexicon.

Whereas directional adjectives such as northern and central were utilised in the preprocessing stage, we have found it easier to carry out the processing of directional nouns (such as 'North'), in the syntactic stage, where they can be treated similarly to place KW's. The words 'North', 'East', 'South', 'West', and 'New' not only indicate that a following capitalised word is a place (or origin) PN, but if the directional word is also capitalised then this indicates that the two words form a compound PN, e. g. 'North Korea', 'New

Jersey'. [2] If the directional word is not capitalised then the two words do not form a compound, but the following word can still be classified as an origin/place PN, (not in the case of 'New'). [3]

In the next section we consider the role of other place KW's in the analysis and acquisition of place PN's.

### The KW Analysis of Place PN's

Place PN's can be defined both by preceding and following KW's. The following KW is easily the more complex of the two, and it is this which we will describe first. It is initially signalled by the head noun in a NG being of semantic category 'loc'. As place PN's can exhibit a variety of types, the variety mainly being a factor of the namehead phenomenon discussed in chapters 5 and 6, the capitalised nouns in the NG must be examined to decide their precise relationship to the head noun. [4] In the FUNES system, when a place PN is detected as unknown, it is entered into the Genus DB with a semantic category of 'loc name', and a supertype triple entered into the lattr DB, linking the PN and its KW. The different possibilities for post-PN KW's are discussed below:

### A) PNcon* KW KW

Example of this type are:

**10.3** 'Crumlin Road jail'
'his Black Sea villa'

In this type the place PN is actually acting as a noun complement to the place head noun, giving its location. It is a common pattern, revealed by the presence of two KW's, the first of which is capitalised. Identification of this pattern enables the actual place PN to be isolated. It can then be looked up in the lexicon to determine if it is known or unknown. It should be noted that we do not consider the whole NG as a place PN in these cases, e. g. we are not classifying 'Crumlin Road jail' as a place PN, but only 'Crumlin Road'.

If the headnoun (i. e. the second KW) is also capitalised this indicates that the entire NG is a single place PN, e. g.

**10.4** 'the Bayswater Road Hotel'

In this case, the entire NG can from now on be treated as a single unit, whereas above we had a separate headnoun, and a place PN acting as a noun complement. The determiner could be used to help in the analysis (if the text were all in single case for example). A definite determiner will indicate a single PN, whereas an indefinite determiner will indicate a PN + separate KW headnoun. However this is not entirely reliable, as in cases of NP reference a definite determiner can be used to describe the latter case, e. g. 'the Edgware Road shop was littered with debris'.

[2] The rules governing the use of the word 'New' are actually stricter than those for the other words. 'New' must always be capitalised, and the following word either a known place PN or unknown. This is because it can also occur in other types of construction.

[3] There is a problem in classifying the following word (or the whole compound) if it is unknown, as it could be a place PN or an origin PN, e. g. 'the south Lebanon security zone' or 'the west African relief post'. We make the assumption that if it was an origin PN it is more likely to have been noted by the morphological heuristics and so will have received a semantic category there; therefore, at this stage, if it has no semantic category it is more likely to be a place PN, and in FUNES is returned as such. (In both cases though it will be added to the origin DB, as it is being used to give an origin).

[4] If there are no capitalised nouns then there is no place PN.

### B) PNcon* Unit-KW

In this pattern the 'PNcon* KW' forms an indivisible unit, e. g.

**10.5** 'Oxford Street'
'Buckingham Palace'

As discussed in chapter 6, different KW's have different attachment priorities as regards their attachment to the PNcon. 'Street' and 'Square' for example cannot be separated from their PNcon. Our analysis of news text has enabled us to identify those place KW's which cannot be separated from their PNcon. This can also be seen as the identification of those place PN's which do not form nameheads, to use Carroll's terminology [30]. This information is included in the lexical entry for the place KW, and can be used to identify this pattern. All the place KW's in the FUNES lexicon are shown in appendix L.

### C) Apostrophe s PN's

These are treated as indivisible units, just like the above type. Although it is possible in conversation where the listener is familiar with the referent to utilise a namehead (e. g. 'St. Paul's Cathedral' — 'St Paul's'), this is not commonly done in news text.

### D) PNcon* Flexible-KW

In this pattern the 'PNcon* KW' does not form an indivisible unit — the PNcon* can occur with or without the KW, hence the term 'Flexible-KW'. As the KW is optional we consider that the lexicon should only hold the PNcon, e. g. in 'Sahara Desert' only 'Sahara' should be held. To ensure that known PN's are located, only the PNcon is looked up in the lexicon. Thus, if we encountered 'Sahara Desert' we would now look up 'Sahara'. Obviously the success of this strategy depends on accurate lexical information on the nature of the KW, and the consequent ability of the PN it occurs with to form a namehead.

There is also an ambiguity problem here in that the PNcon could be a known common noun (e. g. 'the River Pattern'), or a known PN (e. g. 'Devon Island'). This ambiguity can be revealed by a comparison of the semantic category of the KW, and the semantic category found in the lexicon for the PNcon. The semantic category of the KW should take precedence, and if it gives a different semantic category then this will lead to a new lexical entry, for the PNcon in question. Where this new lexical entry leads to a PN/non-PN meaning for a noun, then the letter case of the noun can be used in future to select the appropriate meaning. Where we have two alternative PN meanings, we must make reference to the KW to choose the correct meaning on subsequent occurrence.

If the PNcon is not found in the lexicon our analysis depends on the exact type of KW, and its letter case. However, we must also consider the nature of the preceding PN. If the KW is part of a 'town of X' construction, then the name of the town will be given by X. Therefore, any preceding PN will not be naming the KW. This is another reason for removing origin PN's from the start of the NG as it greatly reduces the problem. However, as we can never be sure of removing all possible origin PN's, we must check for a following 'of'. If one is found then it indicates that any preceding PN is an origin PN, giving the superpart of the KW, e. g. 'the Bosnian town of Bunja Luka'.

If there is no following 'of', and the KW is one indicating optional attachment, then the PNcon only should be retained and used for lexical update. This is because it is capable of occurring as a namehead. Updating with just the PNcon, and not the PNcon plus KW, will enable future detection of both the namehead and the full form, e. g. 'Malaki' and 'Malaki province'.

If the KW is a building type, then there are various possibilities for its relationship with the PNcon. If it is capitalised, then it indicates that the PNcon is a name for the KW (e. g. 'the Savoy Hotel'). In this case only the PNcon should be retained, as the KW is optional. [5]. If the KW is not capitalised additional lexical information can be used to decide upon the relationship. Most KW's are ambiguous in that they can exist in two relationships to the PN — the PN can give the location of the KW, or the company that owns the KW. There are a few that seem to always take a place PN, e. g. station, town hall, university. If the KW is of this type then the PNcon is returned as a place PN. If not no definite conclusion can be reached, the PNcon could be a place PN (e. g. 'a Newcastle factory') or a corp PN (e. g. 'a Nissan factory'). Again this strategy depends on adequate lexical information as to the nature of the KW.

In all of the above patterns, the KW may be plural. If this is so then its meaning also applies to the preceding NP. This preceding NP will not have received a definition, as it will have occurred without a KW, e. g.

**10.6** 'the Hudson and Mississippi rivers',
'Marylebone and Edgware Roads'.

It is only when we come to process the KW NP that we obtain the KW, which must be applied backwards to the preceding PN. In FUNES this is achieved by a procedure which activates if the present KW is plural and flagged as part of a conjunction NP. The previous noun is located from its register, and the present KW utilised to assign it a semantic category (of course this will only be necessary if it is unknown). We must also consider the type of the present KW, as if it is one that cannot separate from its PNcon it must be joined to the preceding PNcon, before lexical update.

### E) Preceding KW's

If the PN-KW patterns fail to match, the KW-PN pattern is applied. This will detect examples like

**10.7** 'Lake Windemere'
'the river Darent'

These types are far simpler — the KW clearly defines the capitalised noun(s) which follows it, providing Genus and differentia information. Place PN's which take preceding KW's can all occur as nameheads, so the type of the KW need not be considered. There is a problem concerning ambiguity, in that the PNcon may be known, but this can be resolved in the same manner as for following KW's. Preceding KW's can, like following KW's, be plural, in which case they define the following NG as well.

It is possible that place PN's of this type may occur as noun complements in PN's of other categories, in particular role and corp PN's, e. g.

**10.8** 'Lake Windemere Noise Control Inspector X'
'the River Darent Conservation Group'

[5] Some KW's which are building words are actually held as 'unit' types, as they must attach to the PNcon, e. g. palace, museum

Such types have not been observed in the news text examined, although they are perfectly feasible. In FUNES, the PN which is the head noun will be detected, due to the heuristic of always utilising the rightmost KW in the NG, so there is no danger of mis-classifying these types. The place PN which is a component of the role/corp PN will not be detected. This is not viewed as a great problem.

This concludes our discussion of the use of KW's in the detection and analysis of place PN's. Like personal PN's, place PN's can also occur on their own in a NG. Below we discuss heuristics for their detection.

### F) Heuristics for Detection of Non-Described Place PN's

Before any of these heuristics are applied we must check for the presence of a plural KW in a preceding NP, e. g.

10.9 'will fly past **the planets Jupiter and Saturn** before ...'

This is handled in exactly the same way as for personal PN's. If there was no such KW then the following heuristics are used to decide if the NG is a place PN occurring on its own:

```
if there is no determiner or a definite determiner
and (the NG begins with 'st' or the NP is in an 'at type' PP)
and the NP is not following an apostrophe s
and none of the words are known personal names
then return as a Place Name.
```

The use of 'St' may seem surprising as an indicator of a place name, given the number of human saints. However, we have found that human saints are rarely mentioned in news, whereas places (such as 'St Louis', 'St Petersburg') frequently are.

If none of the above patterns/heuristics has derived a semantic category for the NG, it may still be a place PN. The most common form which will have eluded detection is a single capitalised word. The definition of such types is left until the semantic stage, where either an appositive or a preposition selectional restriction can provide a category. This is described below.

## 10.3.3 The Semantic Analysis of Place PN's

### Appositive Definition of Place PN's

As discussed in chapter 6, place PN's do not commonly receive a straight definition through an appositive, more often the existence of a place PN can be inferred from an appositive giving locational/directional information. If a straight definition does occur (e. g. 'Rotterdam, the largest European port,') then it will be handled in the manner described in chapter 8.

Locational information can be given in two ways. Firstly it can be quite detailed, giving the distance, and possibly direction, from a referent, e. g.

**10.10**'Lac, north-east of Tirana, '
'Baku, some 25 miles from Makhachkala, '

These types are revealed by the head noun of the appositive being either a measure noun or a directional noun. [6] The appositive NP gives distance or direction information (or

[6] Measure nouns are simply those defined as [measure] in the lexicon. Directional nouns were described on page 82.

both), and the PP(s) give either direction from a reference point, or just a reference point. The CF's returned describe the location of the place PN in these terms. In FUNES these location CF's are entered into the lattr DB and eventually form part of the KB entry for the place PN. The form of the location CF's for the examples above are:

```
10.11   (lac, location, [direction((north,east)),
                         of(tirana)]

10.12   (baku, location, [distance(measure(mile),
                                   number(25)),
                          from_loc(makhachkala)]
```

The occurrence of such an appositive pattern clearly reveals the PN it is attached to as a place PN.

In the second type of locational pattern, the appositive simply gives the superpart location of the preceding place PN, e. g. 'Paris, Texas,'. Such types are revealed by the presence of two PN's with no determiner, which are either unknown or one of which (or both) is a known place. In this situation both of the PN's are clearly revealed as place PN's, with the PN in apposition giving the superpart of the main PN.

The construction used most often to fully define a place PN, however, is an 'of PP'. This is described below.

### Definition through of-type PP's

In this construction the place PN occurs in an 'of PP' which follows a place KW, e. g.

**10.13** 'the city of Liverpool'
'the German border town of Eisenhuettenstadt'

This construction seems to be much preferred in news text to use of an appositive, and is peculiar to place PN's. The analysis is relatively uncomplicated, but there are some factors which render it less simple than it may appear on the surface. One of these was mentioned above in the description of KW patterns, and concerns the effect of an 'of PP' construction after a PN-placeKW construction.

The other complicating factor is that only certain place KW's permit this particular construction. The separation of those which do from those which do not follows no clear rules. Some KW's clearly do permit it (e. g. town, village, republic, city), and some clearly do not (capital, river, hotel). However some words, such as region or province, are more ambiguous, in that sometimes a following 'of PP' conveys a name, whilst other times it may convey a superpart. The main aid in disambiguation here is the type of determiner — if definite this indicates the 'Name' case, if indefinite the 'Superpart' case. It is possible to have a 'Superpart' case with a definite article where a relative clause follows, e. g. 'In the region of Afghanistan where I was born', but this is so rare it is ignored. We have identified the following words as clearly indicating a 'Name' case when there is no indefinite determiner:

'city, town, village, island, republic, region, enclave, province, state.'

In the second example above, the PN 'Eisenhuettenstadt' is revealed as a place PN, with supertype 'town'. In FUNES, the additional differentia information conveyed by 'German' and 'border' would be picked up in the name-frame compilation stage, leading to a final name-frame:

10.14 [noun(eisenhuettenstadt,[loc,name],n),
isa: [town,property(border)],
superpart: germany]

Like the NG KW patterns, this type of KW pattern can occur in the plural with a conjoined NP, in which case the defining information applies to both the NP's, as in 'the cities of Rome and Florence'. In FUNES, this is handled by simply checking the number of the KW in such constructions, and if it is plural utilising it to define the conjoined NP as well as the initial 'of PP'.

PN's can also receive a place Genus through occurrence in other types of PP. The handling of these different patterns is described below.

### Further Uses of PP's

In chapter 4 we described how PP selectional restrictions can only be used to give very broad semantic categories. However, when the head noun of the PP is a PN, the restriction is much tighter. This particularly applies to place PN's. As place PN's very often occur within PP's, a semantic category can still be derived even when there is no other source of description. 'Place PN' is the default category for any unknown PN for many prepositions — within, around, in, on, at, inside, outside, near, through, from, to, towards, and into. The preposition 'of' demands more complex analysis (as usual). The 'placeKW of PN' construction was explained above. However, 'of' can indicate a place PN in other circumstances, in particular:

1. when the preceding noun is directional, e. g. 'north of Exeter'.

2. when the preceding noun selects for a place PN. Many role KW's do this, for instance 'Prime Minister', 'Foreign Minister' and royalty nouns. Some words (e. g. 'president') are ambiguous, in that they can be followed by a corp PN or a place PN. We let them select for place at present, as so many corps contain defining KW's which will not be over-ruled by a preposition selectional restriction. Here, as with the use of place KW's, it can be seen that detailed lexical knowledge of how KW's interact with PN's is crucial for the effective analysis of PN's.

PP's can also give the location of a place PN, in very much the same way as the NP appositive described above, for example

10.15 'Reading, to the west of London,'
'Majdel Silm, at the western tip of Israel's self-declared security zone'.

These PP's are handled by the modules in the customary manner, and will be returned with very similar CF's to those described above for the appositive cases. These CF's then give rise to a lattr DB location entry. Such CF's also lead to a Genus DB entry of 'loc name' for the main PN and the PN from the PP (if unknown).

Finally, PP's can give rise to origin slots for other PN's (e. g. 'X, a doctor from Paris' or 'Y, the President of Togo'), as explained in previous chapters.

This concludes the account of place and origin PN's . All of the patterns described in chapter 6 can be analysed by FUNES, which has acquired a large number of place PN's in its news processing. Examples of these can be found in appendix L. We now turn to look at the analysis of object PN's.

## 10.4 The Analysis of Object PN's

We have given no consideration to the topic of object PN morphology — their detection and analysis occurs entirely in the syntactic and semantic stage. No specific object appositive patterns exist, as they do for place PN's, so the emphasis in this section will be on syntactic analysis. If an object PN is described with an appositive, it will be handled in the standard way. Although some cases of object PN's spreading over several syntactic constituents do exist, none have been observed in the news text studied, and thus we shall not discuss them here.

### 10.4.1 The Syntactic Analysis of Object PN's

Object PN's present problems due to the variety of relationships that can hold between their PNcon and KW, as described in chapter 6.

As with place PN's, a potential object PN is revealed by the presence of an object KW as head noun. Objects are quite commonly accompanied by their manufacturing corp, the removal of which will facilitate the application of KW patterns, as does the removal of origin PN's. In addition, the presence of a corp PN can also aid in the analysis of the nature of the object KW. Corp PN's are detected at this stage using the following heuristic:

1) a known corp PN is detected **or**
2) a word all in upper case is detected **or**
3) a corp KW is detected

In the last case all the words up to, and including, the corp KW are assumed to be part of a corp PN.

Before classifying the remaining elements of the NG as an object PN, it must be determined if the object KW is actually being used as a component of a corp PN. For an object PN to be present we must observe the following:

A determiner is present **or**
(the KW is plural **and not** capitalised) **or**
the NP was preceded by an 's **and** the KW is not capitalised

When an object KW is used as a corp PN component (e. g. 'Brixton Cycles', 'International Business Machines'), the KW will invariably be both plural and capitalised. When we have a pattern of 'Corp + object produced' we will either have a determiner and a singular noun (e. g. 'the Panasonic amplifier') or a plural but lower case KW (e. g. 'Amstrad computers').

The various different types of object PN are described below. As with place PN's, only the capitalised elements of a NG can form part of an object PN. We have found no cases with more than two PNcons preceding the object KW. Therefore all cases below consider the problems of 1 or 2 PNcons.

#### A) Serial-Number Object PN's

These are detected as a combination of letters and digits. The possibility of two serial-numbers is unlikely and therefore not considered. If we have two capitalised words, the most likely case is of a maker and a serial-number (e. g. 'the Amstrad AT-250 computer', a 'Vickers SG-85 missile'). However the semantics of the the non serial-number PNcon are dependent on the presence of known corps. If a known corp was detected in the NG then the non serial-number PNcon and the serial-number are returned as a compound object

PN. If there were no known corps, then the non serial-number PNcon is returned as a corp PN, and the serial number as an object PN.

It may be felt that the entry of serial-numbers into the lexicon (and KB) is questionable, as it could result in vast and un-needed additions to the lexicon This is not the case. In news text, certain serial numbers occur quite frequently (especially those of aircraft), and thus their presence in the lexicon will speed up analysis, making it unnecessary to go through the process of analysing the serial number each time. Moreover, as serial numbers become more well known, their accompanying KW will appear less frequently. This has happened over the past few years with 'AK-47', which always used to be accompanied by the KW 'assault rifle', but which now often occurs on its own.

**B) (PNcon) PNcon (Digit) KW**

In this pattern the PNcons may be naming the object KW (e. g. 'the Black Bird spy plane'), or they may be naming the maker (e. g. 'a Rolls Royce convertible'), or owner corp ('a PanAm aircraft') of the object KW. When the PNcon is unknown we have no way of knowing which is the case. The presence of digits can help in the same way as it did for serial numbers above.

If the KW is an 'artwork' KW, (e. g. painting, film, play), then the PNcon(s) is held to be the human creator, and classified as a personal PN.

If not, the last PNcon is checked to see if it is a digit or a roman numeral. If so, it is decided that the entire group of PNcons are naming an object, e. g.

**10.16** 'a Scud 7 missile'
'the Carlos II cruiser'

If there were no digits present, the presence or absence of corps can be used, as above, to determine the nature of the remaining PNcons. If a corp PN was present, then the remaining PNcons are returned as object PN's, while if no corp was present two hypotheses must be entertained for the PNcons — that they are corp PN or object PN.

As with place PN's, objects may be defined by plural KW's. If so, they are handled in the same manner.

If a post-PN KW has not been located, then a preceding KW must be checked for. Object PN's can be revealed both by a preceding object KW and a preceding corp PN:

**10.17** 'the space shuttle Columbus'
'the Ford Fiesta'

The former case indicates a 'supertype' relationship between the PNcon and KW, the latter case a 'made_by' relationship.

As we mentioned above, object KW's receive no special semantic processing, any appositive description being handled in the standard manner (described in chapter 8). This therefore concludes our account of object PN analysis. We next turn to isource PN's.

## 10.5 Information Source PN's

This is a simpler class to analyse than the preceding ones. The main isource PN we meet in news text is that which names newspapers, and these tend to have simple names, invariably described by a preceding or following KW.

The method of analysis is exactly the same as that already outlined for place and object PN's, and is summarised below:

- Check for post-PN KW patterns (e. g. 'the Herald newspaper'), including any plural KW's
- If fail check for preceding-PN KW patterns (e. g. 'The Saudi daily Ashaar al-Awsat'), including any plural KW's
- Process Appositive Definitions in the standard manner (e. g. 'Hurriyet, another Istanbul newspaper')

The problem referred to in chapter 6, wherein a following appositive could occur giving the name of the paper, which would mean that the PN originally taken as the name of the paper is in fact the name of the owner or publisher has not been completely addressed. This is because it has never been found to occur. Moreover, removal of any preceding corp PN's before analysis of the post-PN KW pattern will prevent mis-classification occurring in many cases were this pattern found to occur.

We do find a problem when the isource PN contains a determiner, e. g.

**10.18** 'the top-selling quality newspaper the Daily Telegraph'

More often than not this is dropped, but when it is not, it will prevent the 'KW PN' pattern being parsed as a single NP, as a determiner cannot follow a noun. Our solution is to check for the presence of a determiner if an isource (or object) KW has just been parsed, and if the word following the determiner is capitalised. If detected, it should be removed.

We have not considered other types of isource PN to the same extent. Other types which do occasionally occur include television and radio programmes, films and books. Examples which occur with descriptive KW's (in the same NG or an appositive) will be handled as the above. However more obscure cases (e. g. '... who starred in Aguirre, Wrath of God') have been ignored as beyond the scope of this thesis, which is concerned with handling the sorts of PN that commonly occur in news text likely to be examined by computer (corporate news, foreign news and political news). As the evaluation described in the next chapter shows, few isource names of this sort occur in such news items.

The final category we examine is event PN's.

## 10.6 Event PN's

Event PN's cannot be described by preceding KW's, only by following KW's. So their detection simply relies on the presence of an event KW as head noun. Normally this KW will be capitalised and a definite determiner will be present. However when the KW is plural these requirements are not enforced as we can meet cases like 'the Vietnam and **Korean wars**' in which the KW may be lower case and a determiner not present.

It is highly unlikely to have any other elements in the NG that are not part of the event PN, so, if the first noun in the NG is capitalised, it can be assumed that the whole NG forms an event PN. Plural cases are handled in the standard manner.

The only problem in the analysis of event PN's comes from the origin component (should there be one). As explained above, removal of origin PN's facilitates the detection of PN's described by KW's. However, this process will make the detection of event PN's with an origin component (e. g. 'the French Revolution') more difficult, as a single event KW (e. g. 'Revolution') cannot be an event PN. One solution to this problem would be to prevent removal of origin PN's if the head noun is a capitalised event KW. However, as we mentioned above, in the case of conjunction patterns this may not be the case. Therefore we leave event PN's in their ill-formed state until the semantic analysis of accompanying adjectives. Here a procedure is included which activates in the analysis of origin adjectives if the head noun is a capitalised event KW. It appends the origin PN to the head noun, and enters two triples into the lattr DB (e. g. ([french,revolution],isa,revolution]), ([french, revolution], origin, france])).

Event PN's that contain PP's are handled by the corp PN (PP) analyser. The component PP will invariably be an origin case, thus the analyser will append it to the preceding NP as a component of the PN (e. g. 'The Great Fire of London', 'The Battle of Bosworth Field'). There are a few cases which do not fit this analysis ('The Retreat from Moscow', 'The Battle of the Bulge'). These could be handled by an extension of the analyser so that it builds a PN if the main NP is a capitalised event word, regardless of the case relationship of the PP.

## 10.7 Summary

This chapter has described the computational analysis of four categories of PN — places, objects, isources and events. Only the first of these is very common in news text, and this has been reflected in the coverage it has received in this chapter. We have considered:

1. Place (and Origin) PN's:
   - Morphological processing of origin PN's, and the use of hyphenated constructs to identify unknown origins.
   - Removal of Place and Origin PN's from compound NG's.
   - The use of place KW's in syntactic analysis.
   - The analysis of appositive constructions.
   - The analysis of PP constructions.
2. Object PN's: the use of object KW's in syntactic analysis, and the problems of differentiating corp from object components.
3. Isource PN's: the syntactic and semantic analysis of newspaper names, involving the use of KW's, both in syntactic and semantic (appositive) stages.
4. Event PN's: the use of event KW's in syntactic analysis, and the problems of component origin terms.

We have shown that an important prior requisite for the effective analysis of PN's is detailed information on the KW constituents of these nouns. This especially applies to place KW's, which can be grouped according to whether their PN component forms a namehead or not. But it also applies to role KW's, which can select for a place or corp complement, and to corp KW's which can select for a place or name component.

In the following chapter we turn to an evaluation of the mechanisms described in the preceding chapters and consider the validity of our approach which attempts to process PN's as they are encountered during processing.

# Chapter 11

# An Evaluation of the FUNES System

## 11.1 Introduction

An evaluation of some sort is an important part of any work presenting a new theory or system. A common way of carrying out an evaluation is to test one's system or theory against existing work of a similar nature. We are hampered in this endeavour by the fact that there is very little work which can provide a benchmark against which to judge FUNES. The existing work on PN's (discussed in chapter 5) is very diverse, and explores the topic from many different angles, but very little of this work has received evaluation through testing on unedited texts. The systems of Rau [137] and Katoh [99] are exceptions, and the performance of FUNES' is compared to these systems. We also compare the methods used in the FUNES system to other PN and word learning work, and also to work on database creation from text. Although this is a less related area, there are some interesting comparisons to be made between FUNES' acquisition of PN's, and acquisition of other types of information from text.

We begin by describing some studies aimed at determining the precise number of PN's in samples of news text, and how many of these PN's receive some sort of description within the text. We then move on to describe the final evaluation of the FUNES system, utilising a small corpus of 200 previously unseen news stories. Based upon the system's performance on this text sample we describe weaknesses in our approach and suggest possible solutions.

## 11.2 Studies to Assess the Nature and Frequency of PN's in News Text

There are two crucial facts that must be true regarding the nature of news text for the work described in this thesis to have any validity. The first is that PN's occur often enough in news text to cause a problem for any system seeking to analyse it. The second is that enough of these PN's are described within the text to enable the FUNES method of utilising these descriptions to be a viable solution to the 'PN Problem'.

Although a cursory glance at today's newspaper would seem to answer this question, some hard figures would be more desirable. There appear to be very few figures available, however, and given the problems that PN's pose for automatic extraction programs or taggers, any figures produced using such aids will be unreliable. Moreover, to the author's knowledge there are no published studies which consider the number of PN's in news text that receive some sort of description. The work reported in chapter 2 on the adequacy of MRD's gives some indicators concerning the question of the number of PN's that occur in news text. Walker and Amsler [172] and Seitz et al. [151] both used news text in assessing the adequacy of MRD's, and both these studies concluded that a large number of the non-covered items from the news text were PN's. Rau [137] mentions an evaluation of the number of unknowns found in a large sample of news, and the fact that some 4% of these were corp PN's (see chapter 5). However, nowhere in these studies do we find hard figures on the percentage of PN's in the text, and no consideration is given to the number of PN's which are described.

Therefore, we have been forced to carry out some studies of our own. These have been small-scale, due to the extremely time-consuming and tedious nature of the work required. This work has to be done entirely manually, as we need to assess the number of PN's and the actual number of constituent words that are part of these PN's. So the words 'Consortium for Lexical Research' form a single PN, yet this single PN comprises four words. Simply locating all the capitalised words in a text is not sufficient for this, even if one had on-line all the text needed for the evaluation. As we wished to carry out the study on English news, and be able to determine the type of news examined, we did not even have this facility.

The first study (originally reported in [45]), was aimed at determining the percentage and degree of within text description for PN's. Three headline stories from three British newspapers[1] were selected at random over the course of a week. This gave a total of 9211 words, of which 846 were PN's, giving a (surprisingly high) percentage of 9.1. The description accompanying each PN was noted and split into three classes : well described, described and not described. To be 'well-described' a PN had to have received sufficient description to enable it to be allocated to a Genus category, and to permit the creation of at least one differentia slot. The criteria for the 'described' class was simply enough information to allocate it to a Genus category. The 'not described' class is self-explanatory, no accompanying information was provided. The percentages for each class were as follows :

- Well-described 38.8 %
- Described 42.1 %
- Not described 19.1 %

These findings, whilst based on a very small sample, show that over 80% of PN's were described. It is informative to consider the nature of the not described items. These were all well-known names — either countries, capital or major cities of well-known countries, or well-known institutions (such as 'the White House' or 'the Kremlin'). The figure of 19.1% is largely due to the frequent occurrence of these well-known items.

What this preliminary study would seem to indicate is that entry of the world's countries and nationalities into the initial lexicon would be worthwhile. Indeed this was done for the FUNES lexicon.

The second study examined a single story from the same papers as above, plus the Guardian and the Sunday equivalents, for a month. This comprised 30 news stories, and a total of 18,320 words. This study did not concern itself with differentiating between well-described and described classes. It was more concerned with determining the number

[1] The Times, The Daily Telegraph, and The Independent

| Words | PN's | Unique PN's | PN+Cons | PN+Cons+WTD's | Desc | Not |
|---|---|---|---|---|---|---|
| 18,320 | 1427 | 768 | 2145 | 3039 | 645 | 123 |
| | 7.8% | 4.2% | 11.7% | 16.6% | 84% | 16% |

Table 11.1: Figures for Number of Proper Names in News Text

of repeated PN's, and the percentage of actual words that the PN's comprised. The results are shown in Table 11.1.

The most important figures in this table are the percentages for described and not described PN's, and the percentage for PN+Cons. This figure counts the actual words comprising a PN, so 'Centre for Medical Analysis' would be counted as four words. It reveals the true extent of PN's more accurately than just counting a PN as a single constituent, as each PN will often comprise several words. The column for unique PN's excludes repeats. It also excludes constituent words (as does the PN column), so does not give an accurate picture of the percentage of a text which PN's comprise. The column for PN+Cons+WTD's counts constituent words and words included in within text descriptions. It thus shows the real percentage of a text that is given over to PN's and their descriptions.

The figures for 'not described' and 'described' PN's were based around the unique occurrences of each PN, as a PN only needs to be described once for FUNES to be able to obtain a definition. We examined the nature of the not described items in more detail in this study. Of the 123 PN's which were unclassifiable by any of the means described in preceding chapters, 62 were countries or nationalities (i. e. origin PN's). Of the remaining 61, 12 were items that could be considered well known enough to be in a reasonable lexicon — religious groups (such as 'Christian' and 'Muslim'), and the words 'MP', 'Treasury', 'Pentagon' and 'Nato'. This left just 49 PN's that were truly obscure and were still not given a description, a figure of 6.4%. It was also noted that 10 of these could have been derived from contextual clues by a human reader, although not by FUNES. For example, one story began 'Five rivers have been highlighted by the National Rivers Authority as in serious danger of drying up. The Pang, Misbourne, Wharfe, Darent and Ver .... '. The contextual device which at present only handles people in FUNES could be expanded to handle such cases. So, the final reckoning brings this figure down to just 5.1% of PN's occurring with absolutely no description.

These unclassifiable types were a mixture of corp PN's and various other categories. As the default for a PN occurring alone is personal PN, when a corp PN occurred thus it was considered unclassifiable. It would appear that in some types of financial news (e. g. market reports), this default could be more fruitfully set to 'corp'. The problem for an NLP system, of course, is to determine the news type first. The other unclassifiable types were a truly mixed bag, defying efficient rule formation. Some examples are shown below:

- A class of 'human groups', where the head noun is a number, e. g. the Maguire Seven, the Birmingham Six.

- Road types, e. g. A2, M62.

- English Places, e. g. Dorset, Essex, the Pennines

- Exceptions, e. g. Trooping the Colour, Footsie, FT-SE 100 Index, Third World, a David and Goliath situation.

Some of these could be identified, e. g. road names are clearly signalled by a capital 'M', 'A', or 'B' and a number. Some of the exceptions might be added to a lexicon (e. g. 'FT-SE 100 Index'), others should just be left. Place names are usually revealed by locational prepositions, it is only irregular types or those that occur as noun complements that may avoid detection (e. g. 'Piccadilly Circus', 'Essex man').

These studies, while admittedly small-scale, do offer support for the utility of the FUNES approach. It is clearly shown that PN's are both extremely frequent in the news, and that the large majority of them are classifiable into at least a Genus category using the FUNES approach. In addition they have been valuable in highlighting those PN's for which initial lexical entry is advisable, due to their common occurrence and infrequent definition (e. g. nationalities and religious groups).

## 11.3 Evaluating FUNES' PN Handling

The very nature of the FUNES approach to handling PN's makes its assessment difficult. This is because FUNES handles PN's in the context of its general language processing. However, its general language processing is limited, as is that of every system, by the extent of its grammar and its semantic analyser. No system can run with a high success rate on free (unedited and unseen) text. The MUC evaluations ([50]) have clearly shown this to be the case, and the systems under evaluation there were the product of many person years work. Thus to run FUNES on free text is to subject it to a very difficult task, in which it will be hampered by the fact that it will not be able to process large portions of the text.

The alternative is to somehow extract a large number of PN's and their surrounding context from text, and then test FUNES solely on these. However this is a chicken and egg situation — to extract these we need a good PN detector. Moreover, how much context would we extract with each PN ? In addition to being rather infeasible to implement, this approach would not test what FUNES is meant to do — analyse PN's in the context of analysing the text in which they occur. [2]

We are thus forced back to the original approach — running FUNES on a test sample of free text. Due to the difficulties presented by this task, we have monitored the analyses produced by FUNES and split the results into two categories. The first shows the performance when considering the whole text. The second shows the performance when only considering those PN's that occurred in text FUNES was able to analyse. This has several advantages. Firstly, it enables FUNES to be tested in a realistic manner, and one which tests the ability to handle PN's while processing the text in which they occur. Secondly it gives us an idea of how well FUNES actually does analyse free text (which, it must be said, is not the central focus of the thesis). Thirdly, using only the text which FUNES was able to analyse will enable us to accurately judge how well the PN handling heuristics are working, without this being confused with problems arising from limited grammatical coverage. In the next section we discuss performance in acquiring genus information, and then move on to discuss the acquisition of differentia information.

[2] Despite the unattractive nature of this approach, it was used to test the system described in [99]. This performance will be described in the next section, when we compare FUNES and other systems.

| News | Words | Average Sent Length | Unknown PN's | Acquired1 | Acquired2 | Incorrect Genus |
|---|---|---|---|---|---|---|
| UK | 6250 | 22.5 words | 535 | 340 | | 12 |
| % | | | | 64% | 77% | 2.2% |
| US | 8800 | 20.6 words | 557 | 310 | | 21 |
| % | | | | 56% | 66% | 4% |
| Total | | | | 60% | 71% | |

Table 11.2: Results for the Acquisition of Proper Names from Unseen Text

### 11.3.1 Evaluation of the Acquired Genus Information

The evaluation [3] used two samples of 100 stories each. The first batch of 100 were taken from the four UK quality newspapers mentioned above. They were all relatively short, between one and six sentences long, and comprised in all 6,250 words. Individual sentences can be very long and convoluted in news stories, and these were no exception, with some being between 30 and 40 words long. The stories were arbitrarily selected from among home, foreign, and business news for the month of January 1992. The only pre-editing that was done was to add some vocabulary items (mainly phrasal verbs and compound nouns) to give a greater chance of parse success.[4]

The results are summarised in Table 11.2. For the UK news we have included an earlier test run on 30 unseen stories, from the same papers. The average sentence length in the UK corpus was 22.5 words. This shows the complexity of the sorts of sentence we were expecting FUNES to analyse. The stories contained 825 occurrences of PNs, of which 712 were unique (the other 113 being repetitions). Of these 712, 176 were already known, leaving 535 unknown PNs to potentially be acquired. [5] A PN was considered acquired only if a unique and correct genus category was produced. In many cases, two or three different categories were returned. Even though these are retained for future refinement, we did not consider them as successful acquisitions.

FUNES successfully acquired 340 of the 535 unknown PNs, giving a percentage of 64% (the entry for 'Acquired1' in Table 11.2). Of the 195 PN's not acquired, 92 were in portions of the text that were not analysed due to parse failure. Therefore, if we only consider PNs that were analysed, the percentage successfully acquired rises to almost 77% (the figure for 'Acquired2' in the table).

As the test sample was taken from the same source as the data used for developing FUNES (i. e. UK newspapers), it was felt appropriate to run a test with data from a new source. For this purpose, 100 stories from the ACL/DCI CD-ROM of Wall Street Journal news were selected. This provided a challenge to the generality of the PN handling heuristics by moving to a completely new source and a new dialect. These 100 stories were longer, comprising 8,800 words. The average sentence length was similar to the UK stories - 20.6 words long. The corpus contained 820 PN occurrences, and 617 unique occurrences. Of these, only 60 were known, leaving a total of 557 unknown PN's that could potentially

[3] originally reported in [46]

[4] FUNES has a base lexicon of only 2,000 roots, so this addition was considered acceptable. A lexicon of around 10,000 roots seems to be generally considered the minimum for effective news text processing.

[5] This may seem like a large number of known PN's, but it must be remembered that in addition to having a large number of origin PN's manually entered in its lexicon, FUNES had already automatically acquired many PN's from its Development Corpus.

be acquired. Of these, 310 were successfully acquired, giving a percentage of 56%. Of the 247 not acquired, 86 were in portions of the text that were not analysed due to parse failure. Therefore, if we consider only PNs that were analysed, the percentage successfully acquired rises to 66%.

The combined figures give percentages of almost 60% for all PN's, and 71% considering only those PN's that were analysed.

The percentages of PN's returned with an incorrect genus were very small in both categories, 2% for UK and 4% for US news. The majority of these were due to ambiguous KW's (such as 'forum', 'accountant' or 'shareholder').

It is slightly strange that the percentage of PN's occurring in unanalysed text in UK news should be higher than in US news, given that the US news is more complex. The only explanation is, that although more of the US news was actually being parsed, and thus more PN's were being parsed, many of the parses were incorrect. In addition, the semantic analysis performed was less correct than that for the UK news. For instance, the 'interposed' appositive construction discussed in the next section is parsed, but incorrectly, and the semantic analysis derived is therefore incorrect. But as the sentence in question was 'parsed', the PN's in the construction are counted as analysed.

As we stated in the introduction there are no comparable studies we know of which assess performance in this manner. The systems of Katoh and Rau reviewed below are both PN extractors, which do not run in tandem with an NLP system. To give an idea of the difficulties of processing real text we provide some figures from other systems aimed at this task — specifically the Core Language Engine [9] and the MUC-3 systems [50]. This data is not directly comparable as it concerns performance on natural language analysis as a whole, and not just PN analysis. However, the FUNES figures for PN analysis do give us an idea of the level of performance of the more general language analysis of which the PN analysis is a part. We are not claiming that the PN figures are in any way directly transferable into figures for language analysis, just that successful analysis of PN's in a sentence is unlikely without a reasonable syntactic and semantic analysis of the sentence.

The Core Language Engine was given an initial evaluation on 1,000 sentences from the LOB corpus, **which were all of 10 words or less in length**. Thus the complexity of the test sentences would be many times less than those used in the FUNES evaluation. However, the semantic analysis of the CLE is far more powerful than that of FUNES, with much attention being paid to quantification. 63% of the test sentences were considered to receive a correct semantic analysis.

In the MUC-3 evaluation the main evaluation metrics were recall and precision. Systems were evaluated as to their performance in filling templates summarising the terrorist stories under analysis. These templates consisted of various slots such as location of incident, perpetrator of incident, target etc. Recall was assessed by dividing the number of correct slot fills into the number of 'answer key' fills, e. g. in a story giving five human targets of an incident, a system which named four of them would receive a recall score of 0.8. Precision was assessed by dividing the number of correct slot fills into the total number of slot fills. So if a system had actually produced eight human targets its precision would only be .5, whereas if it had produced just the four, its precision would be 1. The highest recall score obtained was around .5, the highest precision was also around .5. If penalties for spurious slot fills are lessened then the best precision improves to around .65. These figures show how difficult accurate text understanding of real text is.

| Corpus | All info | > 50% | < 50% | Incorrect Info |
|---|---|---|---|---|
| WSJ | 43% | 33% | 24% | 20% |
| UK News | 54% | 34% | 12% | 17% |
| Total | 48% | 33% | 19% | 19% |

Table 11.3: Results for the Acquisition of Differentia Information

### 11.3.2 Evaluation of the Acquired Differentia Information

The analysis of differentia information is a complicated and somewhat subjective business. To give an impression of how well FUNES performed we examined those PN's which were successfully acquired and which had one or more differentia slots. Each of these slots was evaluated against the information held to be contained in the text. The results are shown in Table 11.3.

Given that the evaluation cannot be exact, we used three fairly broad groupings. 'All info' shows the percentage of PN's for which all the information in the text was considered to have been extracted. The second column shows the percentages where more than half the relevant information was extracted (but not all). The third column shows the percentages where less than half the information was extracted (but more than zero). The 'Incorrect Info' column shows the percentage of PN's which contained incorrect differentia slots, whether these were PN's which had had all the information acquired or only some.

On the whole we find these results to be quite good. Around half of the PN's that were successfully acquired as regards their Genus category had all the differentia information correctly extracted. Another third had more than half of this information extracted, while the remaining fifth had less than half extracted. The main drawback is the relatively high percentages for incorrect differentia information. The creation of incorrect differentia slots is due to breakdowns in the syntactic and semantic analysis, some of which will be discussed below. However, a large number of incorrect entries were found to be due to the global origin heuristic (described in chapter 10) and the novel US appositive construction referred to below. So this figure is not quite as problematic as it may first appear.

The above results purely concern performance on PN's. We have not attempted to evaluate the performance of the system's parser or semantic analyser, as this is not the focus of the thesis. Moreover, as the analysis of PN's takes place within the context of general language analysis, the performance on PN's can give some idea of the performance of the system as a whole (as mentioned above).

In the following section we will examine the main reasons for failure in the above evaluation, and their implications for improvement.

## 11.4 Problem Areas Revealed in the Evaluation Study

### 11.4.1 Problem Areas in General Language Processing

The main reasons for performance failure in this area, as opposed to the more specific area of PN analysis, were :

1) Lack of Vocabulary
2) Lack of Syntactic Coverage
3) Lack of Semantic Coverage

As might be expected, these shortcomings correspond to the three main areas of the FUNES system. The performance of the pre-processing unit has been found to be very good. Only a few failures were traced to misclassification of ambiguous words or unknown words.

As stated above, the FUNES lexicon only contains about 2,000 word roots. In every story that it processed there were unknown common words, in addition to the unknown PN's. The numbers varied from one or two up to twenty or so. We feel that FUNES has exhibited high performance in view of this. However, a larger lexicon would permit a much improved semantic analysis. It would also lead to improved acquisition of common nouns. The more words that are known in a sentence the greater the chance of acquiring a semantic category for an unknown. For instance, if both a verb and its object are unknown, then it is not possible to utilise the verb's selectional restrictions to acquire a meaning for the object noun. If the verb is known and the object unknown then selectional restrictions can be used. In addition to a lack of single word vocabulary, lack of compound word vocabulary was also an important factor. Not only does this lead to incorrect semantic analysis, but it can also lead to parse failure when a compound noun is handled as a noun plus a verb for example (as was the case with 'chief operating officer').

The decision to hold noun/adjective ambiguous words as only nouns did indeed facilitate the task of lexical ambiguity resolution, but it also led to some parse failures when such a word occurred as an adjective, and was followed by another adjective. As the FUNES grammar does not accept a NP consisting of a noun followed by an adjective the parse failed. Such words should thus be returned as nouns, unless followed by an adjective.

Syntactic and semantic failure was due to a variety of reasons, the following being identified as the most common:

1) Conjunction of non-NP constituents
2) Prepositions followed by VP's
3) Reduced Relative Clauses
4) Speech
5) Topicalisation

Depending on the contexts in which these constructions occurred they may cause parse failure or inconsistent semantic analysis. For instance, a reduced relative clause on a subject NP would be taken as the main VP. The parse would then fail when the main verb was encountered. If it occurred on an object NP then the parse would fail at that point. Conjunction of sentences leads to the subject NP of the conjoined sentence being taken as a conjoined NP on the final NP of the first sentence.

Each of these constructions would need to be covered if performance is to dramatically improved. It is felt that an improved FUNES could perform to a high standard on the sort of news encountered in the UK news sample. Although this is complex in structure, it is within the grasp of an advanced NLP system, as it is grammatical, and uses a restricted set of constructions. However the sort of news encountered in the WSJ sample is felt to go beyond that analysable by a syntactically-oriented, essentially Top-Down processor. It is both telegraphic and agrammatical. Examples of the corpus are shown in Appendix M. Effective analysis of this sort of text would need either a system constructed especially for it, or an immensely complex and powerful tool of the sort used in the MUC trials, e. g. GE's NL Toolset or SRI's Tacitus. Even then, performance would be far from perfect, as shown by the results quoted in the previous section.

All of the above inadequacies contributed to PN failure in a general way. However it is possible to pinpoint some problems that are more specific to PN's in particular. We discuss these below.

### 11.4.2 Problem Areas in PN Analysis

- Elliptical conjunction. This leads to the loss of differentia information, e. g. 'Hudson General Corp president and chief executive officer, Alan Stearn.' In this expression only the NP 'chief executive officer' is matched to 'Alan Stearn'.

- Hyphen problems. At the time of the evaluation capitalised prefixes were not identified as name components, so names like 'All-Burma Young Monks Union' were not processed. This has now been corrected.

- Lack of KW vocabulary. As the correct classification of PN's depends on knowledge of their accompanying KW's, if a KW is unknown then the accompanying PN will not be classified. The unknown KW's from the test corpora have now been added to the lexicon.

- Reduced Relative clauses. These are commonly used within appositive NP's to provide further descriptive information. The failure to handle these meant that this information was lost.

- Some role words can indicate that any preceding unknown capitalised word is a place PN, rather than a corp PN. So we will talk about 'a W. R. Grace chairman' and infer that 'W. R. Grace' is a corp PN, but also talk about 'a Salisbury magistrate' and infer that 'Salisbury' is a place. Facility to cope with these has now been added.

- The Global origin heuristic often produced incorrect entries. When the correct origin was among these we do not consider it a problem. However, in home news, the location of places and the origin of actors is often left unmentioned as it is considered obvious that it is the UK. The origin of foreign entities will be mentioned, and this can lead to the incorrect application of the foreign origin to those entities which were not given one.

- Distal reference. Although a preceding description can be applied to a following undescribed human name (see chapter 8), this construction is occasionally used for corp and place PN's in which case it cannot be handled. We also find reference through pronouns to preceding corps or groups, e. g. 'Their leader, X,' or 'The chairman, Y,'.

- Some categories of PN that occurred are not handled by FUNES, e. g. time periods ('Remembrance Sunday'), diseases ('Guillian-Barre syndrome') and medicines ('Retin-A acne medicine'). Some PN forms are also not handled, e. g. film and television titles such as 'The Trials of Life' and 'Lethal Weapon II'. Finally there are peculiar forms even in those categories that have received deep analysis, e. g. 'the United Steel Workers Local 3057', 'Toys R US'.

- Corp PN's with PP's or with conjunctions, that are used as noun complements. This is not common, but does occur, e. g. 'Housing and Urban Development Secretary Jack Kemp'.

- Various case relationships for PP's were not handled, leading to a loss of differentia information, e. g. 'X, the **heir to the Russian throne**', or 'Y, a bitter **enemy of** Z'. As the case labels returned for these, and other, examples were not considered definitional, the differentia slot fill was impoverished.

- In WSJ text it is common to see long sequences of PN's separated by semi-colons. FUNES handling of these is not good, it at present handles them as periods, which means that most of the PN's are missed. If it were to handle them as commas, it would mean that the list would have to be treated as an apposition/conjunction problem, which is not easy to resolve.
- In WSJ text it is common to have an appositive NP inserted into the middle of another NP, e. g. 'Culligan is a Northbrook, Ill., subsidiary of E-II Holdings Co.' This construction was not catered for and was one of the main reasons for PN failure on WSJ text. The 'Northbrook,Ill' unit would be handled correctly, but then the appositive NP (Ill) is taken to be described by the continuation of the original NP ('subsidiary of E-II Holdings Co'), and the main NP is taken to be described by the first half of the interposed appositive unit (Northbrook).
- The Appositive/Conjunction handling was inadequate.

Other factors were identified as leading to failure, but this selection gives some idea of the sorts of problems encountered. We must attempt to separate those which are acceptable errors, from those areas which need refining. Principal among the acceptable errors are the irregular name forms. No system is ever going to obtain 100% performance, and it must be accepted that there will be always be some names that are irregular. As one moves away from business and foreign news, and into areas like arts and film news, the number of strange forms increases. Isource names in particular are prone to occur with no WTD, especially if they are the names of famous films or shows. The demand for computational analysis of this sort of news is not likely to be high, so these problems are not great.

Some problems are directly attributable to poor syntactic or semantic coverage. These would be resolved as the syntactic and semantic coverage is increased. Similarly, additions to the lexicon, particularly KW and compound terms, would lead to improved performance.

The specific areas that we would desire to improve are:

1) handling of apposition and conjunction and their interaction
2) handling of distal descriptions

In addition to these improvements certain extensions of the system will also be discussed:

1) handling of the interposed appositive construction
2) extension of the PN categories handled
3) extension of verbal definitions
4) use of a certainty factor for PN semantic classification

### 11.4.3 Improvements

#### Conjunction and Apposition

Conjunction is perhaps the greatest problem for existing NLP systems, and FUNES is no exception to this. It presents many problems that have not been solved, although many of the simpler cases are handled well. In particular, ellipsis can lead to a loss of information in constructions like 'former president and chief operating officer of New England Electric'. When we have a conjoined NP with an attached PP, as in the previous example, we must decide if the PP also describes the preceding NP. In this example the PP 'of New England

Electric' clearly attaches to both 'former president' and 'chief operating officer'. However, we could imagine 'former senator and chief operating officer of New England Electric'. In this case the PP is only part of the second (conjoined) NP. A decision must be based on the compatibility of the two NP headnouns, for example 'president' and 'chief operating officer' are compatible in that both are corporate posts, but 'senator' is not. Other cases of ellipsis were mentioned above. Simpler cases occur in expressions like 'George and Maggie Smith' or 'Alf Dubbins and his wife Olive'. These could be handled in the same manner as plural KW's.

The use of appositive NP's to describe several following or preceding NP's is also a problem. This was discussed in chapter 8, and the above evaluation has shown that it does occur enough to demand more examination. Often words such as 'both' can give a clue, but the number of the appositive headnoun is the deciding factor. For example in 'Scotia Mcleod Inc and RBC Dominion Securities Inc, both Toronto-based investment dealers,' the fact that 'dealers' is plural should indicate that both the conjuncts must be retrieved and passed to the semantic analyser.

The heuristic to divide an appositive NP from a conjoined NP is too reliant on 'all or nothing' indicators such as letter case and presence of commas. Although these are often used in the manner expected by the heuristic, when they are not it fails completely. For instance, in WSJ text corp PN's often have 'unit' or 'subsidiary' appended to the end, so what is really a PN will have a lower case headnoun. If this occurs after or before another corp PN without such an ending FUNES immediately assumes they are in apposition. Ideally, this heuristic should be combined with higher level understanding of the text, as described in chapter 8. The problems of appositive/conjunction interaction argue strongly for the integration or close-coupling of syntax and semantics. A further factor that needs to be considered here is the semi-colon, which is frequently used to break up long groups of appositive/NP pairs.

We feel that the FUNES approach of handling these constructions within the NP parse is a valid one. An alternative, which is used in GE's TU system [95], is to attempt to separate out PN's and their accompanying appositive descriptions within a pre-processing stage, and then to hand these to the parser as single units. Although this sounds very appealing, and would obviously greatly facilitate the parser's task, how the pre-processing tool works has never been described. As the handling of apposition and conjunction is one of the most complex tasks in language analysis, the pre-processor would probably end up being almost as complex as a parser. What would essentially be happening is two passes through the input, one to handle PN's, the other to handle the remaining text. In essence though, the handling of PN's would probably be very similar to the FUNES approach, regardless of where it was undertaken. This theory has been supported by Mcdonald (pers. comm. ), who, commenting on the similarities between FUNES and his system Sparser's approaches, concluded that this was due to the nature of the task.

### Distal Descriptions

The use of distal descriptions covers a wide variety of constructions. Corp PN's can be described (like personal PN's) in a previous sentence. This could be handled by an extension of the procedure which currently handles personal PN's. In US news one often sees the term 'this ... company' used to refer to a corp PN given in the headline. Therefore corp PN's mentioned in headlines should be kept as potential referents for descriptions within the body of the story. Pronoun references are more complex. They include expressions such as 'their leader' or 'its chairman'. The use of the possessive determiner signals the fact that we have some kind of reference, the type of referent is indicated by the headnoun.

So 'leader' implies the desired referent is a group of people (rebels, petitioners etc), and 'chairman' that the referent is a corp.

A similar situation occurs with definite NP references such as 'the chairman'. The problem here is that when encountered there is no way of telling if 'works_for' information will be coming in a following PP, or if the word 'chairman' refers back to a previously mentioned company. In business news, and US business news in particular, a solution similar to the global origin heuristic would work well, i. e. if a person has a role but no works_for slot, then utilise the previously mentioned company to create one.

### 11.4.4 Extensions

The interposed appositive construction is peculiar to US news, and appears to be used exclusively for giving state superpart information. The most parsimonious way to handle this would be to add the fifty US states and their abbreviations to the lexicon, and to give special consideration to a superpart type appositive containing a state name. This would lead to the appositive being removed after semantic analysis, and prevent the continuing NP being taken as a further appositive describing the state name.

Extension to include the PN categories 'time', 'disease' and 'medicine' should also be undertaken. Observed examples of these categories have been described exclusively through KW's. Appositive descriptions could at present be handled, but the semantic hierarchy needs expanding to include the categories 'disease/illness' and 'medicine/drug'. Addition of the relevant KW's, and expansion of the KW handling procedures would handle cases of KW description.

As described in chapter 8, FUNES does utilise some verbal descriptions in the creation of differentia slots for PN's. The WSJ news was particularly rich in verbal definitions, especially those describing job changes. Some examples are show below:

> Rudolph Agnew,55, was named a nonexecutive director of ...
> Pierre Vinkern, 61, will join the board as a nonexecutive director
> Magna Stores Inc, which operates a chain of speciality retail stores
> Backe Group Inc agreed to acquire Atlantic Publications Inc

Extension to handle such cases would greatly increase the amount of information extracted from the text. This would require an analysis of the sorts of verbs used (name, acquire, supply, merge), and the extension of the current mechanisms for verb definitions.

The last area we wish to consider is the use of a confidence factor which would express level of confidence in a particular genus category. [6] This would enable a more elegant handling of disparate sources of information. For instance, many more morphological heuristics could be utilised, if they contributed one (or more) genus categories together with a confidence in the correctness of those categories. Thus if a word ended in 'ese', it could be returned as an origin PN with a confidence of .9, and a personal PN with a confidence of .1. These factors would be maintained throughout processing, and finally the highest one retained. Similarly, the default personal PN or place PN heuristic could be changed so it carries a certain confidence factor, which might then be decreased or increased by the presence of an appositive. At present, the default heuristic simply returns a genus category, and then the appositive handled returns another category, and if they are not the same both are retained as equally valid (unless we have a human name followed by a corp name, in which case only the corp name is retained). FUNES does have a 'priority rating' for each information source, and will not allow a lower priority information source

[6] This is a mechanism used in the FactFinder system [125] described in the following section.

```
propnoun ← human
         ← cmpny
human ← khuman name
khuman ← rank
       ← FORMER rank
rank ← POST
     ← state POST
     ← STATE President
     ← Soviet President
     ← COUNCIL President
     ← COMPANY President
FORMER ← former, acting
POST ← Foreign Minister etc.
STATE ← US,French, etc.
```

Table 11.4: Example Proper Name Grammar Rules from Katoh et al

(such as a verb selectional restriction) to contradict a higher priority source (such as a KW). However, a graded system of confidence factors would be more sensitive and effective.

## 11.5 A Comparison of the FUNES system and other PN processors

In this section we compare FUNES to the following systems:

- The PN handling of Katoh et al.'s English to Japanese MT system [99]
- The Corp PN extractor of Rau [137].
- The FACTFINDER system of Miller [125].
- The MLNL project of Geller et al. [66, 62]

### 11.5.1 The System of Katoh et al.

This is the most similar piece of work to FUNES in its overall approach, and was clearly motivated by many of the same concerns. We described it briefly in chapter 5. The basic approach is to handle PN's as compound words, composed of the name itself plus its KW. The only category of PN described in detail in the paper is that of personal PN's, although the use of KW's in handling corp and conference names is mentioned. The systems utilises 'local' grammar rules for the processing of PN's, as shown in Table 11.4. It does appear that these rules are rather less general than those used in FUNES. However, they have been derived from a large corpus of news, and presumably reflect the particular terminology of that corpus.

PN's are handled in a separate stage between the morphological and syntactic. This stage utilises a special PN dictionary and rules like the above, to identify PN's and return them as a single compound word to facilitate the syntactic analysis. A candidate PN is detected as a sequence of lower case letters following an initial capital. Unknown PN's are acquired by matching candidate PN's to particular PN grammar rules. The newly acquired PN's are entered into a dictionary of new terms. As described in chapter 5, one of the problems for English to Japanese translation is the use of a different word in Japanese for the same word in English. To overcome this problem various different versions of each PN are acquired (e. g. 'Lee Raymond', 'President Lee Raymond', 'Exxon Corp President Lee Raymond').

The system was evaluated using simple word sequences, rather than in the context of the MT system it was designed for. The evaluation consisted of translating the 1000 most frequent names in the AP news corpus. Before this test was undertaken the local dictionary (which holds PN's) already contained 3,000 entries. Given this, a considerable number of the 1000 most frequent corpus names would have been already known (we are not told how many were). The results showed that 94% of the 1000 names were analysed successfully. Of the failures, 80% were due to title KW's not being in the local dictionary, and 20% were due to name ambiguity (e. g. a name held in the local dictionary as a place occurs as part of a personal name).

While 94% is an impressive result, it must be treated with some reservation in comparison to the FUNES best result of 85% [7] or 77%. This is because all of the FUNES PN's handled were unknown, whereas we are not told how many of Katoh et al's were. In addition, all of FUNES PN's were handled as they occurred in news text, whereas Katoh et al's were not, being removed from the text beforehand. This means that the system never had to differentiate non-PN's from PN's, nor did it have to deal with any of the more general language analysis problems that complicate the analysis of PN's in context.

In summary, we would conclude that the work of Katoh et al. supports the approach taken by FUNES, in that it is one of the few working news processors in the world and it has paid so much attention to the problems that PN's can cause. The approach is similar, but the focus of the work is different, being MT rather than TU. No attempt is made to acquire more detailed information on PN's, nor is appositive information dealt with. This last shortcoming is perhaps the most serious, as appositives are so frequent in news text, that even if one is not utilising the information they contain to help in the processing of PN's, some consideration must still be given to handling them.

### 11.5.2 Rau's Company Name Extractor

As described in chapter 5, [137] describes a system designed to operate with GE's NL Toolset, which extracts company names from news text. It is capable of operating in both mixed case and all upper case input. While the operation of the system itself is very clear, it is not clear how it integrates with the rest of the GE Toolset.

The system basically relies on the presence of company name suffixes such as 'Co' and 'Inc' for company name detection. If the input is in mixed case it reads back from these to the first non-capitalised word, and when this is found exits. This heuristics appears to be sensitive to linking items such as 'de' and 'van', but no mention is made of '&'. If the input is in mixed case the system reads back from the corp suffix and utilises a stop list of words found to commonly precede the start of a corp name.

[7] This figure is taken from performance on the Development Corpus. Although it does not give a reliable indication of performance on free text, it does give an idea of the level of performance that could, in theory, be obtained using the FUNES approach.

The system also detects corp names involving conjunction, utilising constraints on verb agreement and words such as 'all' and 'each'. How effective it is at differentiating these from different corp names linked by conjunction is not shown. The use of verb agreement was considered in the FUNES system but rejected as too risky, as journalists appear to vary in their use of singular and plural verbs with corp names, e. g. 'British Airways has always', 'British Airways have always'.

Consideration is given to the problem of variant forms in a very similar manner to FUNES. Abbreviated forms and shortened forms are generated for all corp names extracted and entered into the lexicon. Certain conditions are placed on variant forms, to prevent erroneous constructions. Problem of ambiguity arising from the same namehead form are also catered for.

The results are in many ways awkward to interpret. From an analysis of 1,510 stories a human extracted 772 corp names, while the program extracted 1187, of which 693 were in common with the human, and 494 were not. Presumably these were all valid names, but this is not stated clearly. The human extracted 53 names that the program missed due to the absence of a suffix. In summary, the program extracted 97.5% of the names extracted by the human **that had suffixes**, and extracted an additional 40% of names.

The above results appear to indicate a very impressive system, as indeed it is. A direct comparison with the figures for FUNES cannot be made as the two programs are performing somewhat different tasks. Rau's program is purely a corp PN extractor, whereas FUNES detects and acquires seven different categories of PN. Rau's program is not an NLP system that acquires corp PN's as it runs, so it does not have to cope with all the complications of NL analysis. Nor does it acquire any differentia information. The program's most marked feature is its total reliance on company name suffixes. Since the system has been constructed for US news this is reasonable, but as we have mentioned before, the habit of suffixing every company with 'Co' or 'Inc', even if it is not formally part of the name, is not a practice followed in English news. It can be expected, therefore, that performance would degrade if tested on English news. From this point of view the reliance poses a restriction on the system's utility, in a similar way to FUNES' reliance on mixed case, which poses a restriction on its utility. In the conclusion to [137] Rau states 'extracting company names without such a suffix, especially with upper-case input is a difficult, if not, impossible task'. We would contend that the use of appositive descriptive phrases goes some way to overcoming this task, as does the use of a large set of corp KW's, both of which are utilised in FUNES.

The absence of use of appositives is a drawback of the system. Rau has stated (pers. comm. ) that this is not necessary due to the vast amounts of news the corp name extractor can be run on. Should a name occur with apposition most of the time, and a suffix only once, it will still be detected due to this single occurrence.

While its handling of ambiguous nameheads is more advanced than FUNES, it would appear to create many problems for itself by automatically creating these variations and adding them to the lexicon when it encounters the full name. FUNES, on the other hand will wait for a confirmatory occurrence in the text, before adding the variant form to the lexicon.

### 11.5.3 Miller's FACTFINDER system

Work on the FACTFINDER system [125] is as yet unpublished. As we have little information about it some of our comments below may be unfair or inaccurate. However, given it is the only system we know of solely aimed at extracting all classes of PN, it is certainly worthy of discussion.

FACTFINDER is an ongoing project aimed at 'automatically extracting important facts from documents and electronic messages in a timely and effective way'. Facts in this terminology seem to correspond directly to names of all sorts — PN's of the sort we have been discussing, dates, numbers, scientific terminology and medical terminology. The aim is to develop an automated tool to scan texts for facts and present them to a human analyst in a meaningful way. To achieve this aim the system relies on a pattern-matching approach, combined with detailed knowledge bases.

The pattern matcher scans text from left to right, using a word lexicon to identify simple words, and a suite of grammar rules to combine the identified words into 'facts'. Each identified string is assinged a confidence factor to indicate the degree to which it fits a fact category. These confidence factors can be combined to aid in ambiguity resolution, e. g. 'Georgia' is held as both a person's name and a state name. If this co-occurs with 'Atlanta' then the state confidence increases, but if it co-occurs with 'Brown' then the person name confidence increases. The word lexicon does not simply rely on KW's to make a match, but includes many PN components to aid in disambiguating word strings. For instance the word 'Bernadino' is defined in the lexicon as a component of a mountain range called 'San Bernadino'. All of the lexical entries appear to have been hand built.

An evaluation of the system is not possible as yet, as it is currently undergoing testing. The number of different 'fact' categories it is hoped to identify appears very ambitious — 24 in the initial test (however it is claimed to be able to identify up to 140 different categories). It is hard to compare to FUNES as it is different in so many ways, and very little material is available. Firstly, it has no connection with an NLP system, it simply scans though text identifying the PN's (or facts). Whether it would be possible to link up to an NLP system is not known. The patterns it uses appear to be similar in some ways to FUNES syntactic patterns presented in chapter 6, but there are several differences. FACTFINDER appears to make great use of a large knowledge base which contains many PN's and PN components, whereas FUNES tries to get by with very little initial PN knowledge. In addition, it is not clear how it handles conjunction or apposition in its patterns. Finally it goes no further than identifying and categorising the PN's/facts, it obtains no differentia information.

Performance statistics will be of great interest when they do finally emerge, as will information on the possibilities of linking up with an NLP system.

Of the news text work that we reviewed in chapter 2, very little goes into detail into its handling of PN's. We discussed PN handling in the systems of Kuhns, Mcdonald and the various MUC-3 participants. What PN handling is attempted goes no further than use of morphology to detect Spanish names (in various of the MUC groups), and the use of corp KW's to classify unknown capitalised words as corp PN's (Kuhns). Mcdonald's system goes along way further than this, but nothing is published as yet describing its PN handling. It is claimed (pers. comm. ) to be similar to FUNES.

Rau [139] does make some references to handling other classes of PN beyond company names. Her approach to handling personal PN's utilises title words, large lists of first names, and detection of consecutive unknowns. Brief mention is also made of the use of 'prefixes' and 'suffixes' in the detection of other names. From the examples given (island, villa, mountain), these would appear to be analagous to the FUNES KW's. As so little information is given, no detailed comparisons can be made. On the surface the approaches appear very similar.

Work on the Core Language Engine [9], a broad coverage syntactic and semantic analyser for English aimed mainly at interface applications, has also briefly considered the problem of PN's. In [32], a simple PN inference component is described. This appears to be aimed purely at personal PN's (and maybe place PN's) — certainly that is the only class it could handle with any success. The inference component can be turned on after tokenisation and will return any sequence of capitalised words as a single PN. If the sentence initial word is unknown it is counted as a potential PN. While this approach will enable detection of personal names in interface contexts, such as 'Does Michael Peterson still work here?', it will not work on PN's that involve prepositions or conjunctions, nor will it correctly extract place or corp PN's where the KW is not capitalised. All PN's discovered by the inference component are given a sortal restriction (semantic category) of 'physical object', which includes animate being and inanimate objects, so no detailed semantic classification is attempted.

In summary, we can say that the above-described work is very similar to FUNES in its motivation and some of its methods, lending strong support to the necessity of specialist PN analysis. The main differences are the level of detail between the above systems and FUNES. The above lay major emphasis on the use of KW's, none extend the analysis of PN's into the semantic stage, nor do any of the NLP-oriented systems make use of appositives. FUNES builds very detailed name-frames on the PN's it encounters, purely because a high level of detail is commonly supplied in the text, whereas the above do little beyond formation of a genus category.

The final comparision we shall make in this section is of a more theoretical nature, and considers the nature of the 'learning' which takes place in FUNES.

### 11.5.4 The MLNL Project of Geller et al

[66, 62] describe a project at the New Jersey Institute of Technology to model the language acquisition of a human child and adult. It was partially described in chapter 2, and the reader may wish to review that description before continuing.

The modelling of the acquisition of older children, through the single-channel approach, uses 'Explanation Patterns' to provide simple definitions for unknown words. These patterns consist of known closed class words, known words acquired through the multi-channel approach, and a single unknown. The explanation patterns lead to the creation (or modification) of a memory structure, represented as a semantic network, in which the unknown word is grounded by the known words. The explanation patterns are not considered to provide a full semantics for the unknown word, but to connect it to words learned through the multi-channel approach. This approach is felt to avoid the symbol-grounding problem described by Harnad [75].

Some of the different types of grounding described in [62] are similar to the acquisition of PN's in FUNES. Top-down grounding acquires an unknown word as a subtype of a known word, e. g. 'a dog (unknown) is an animal (known)'. Top-down grounding with distinguishing attributes acquires an unknown word as a subtype of a known word, plus an item of differentia information, e. g. 'a boy is a young man'. Top-down grounding with an individual is the term used to describe the acquisition of PN's, e. g. 'John is a boy'. However this process appears to work in exactly the same manner as simple Top-down grounding.

The work can be viewed as a simplified, but more formalised, version of the learning occurring in FUNES. Such phrases as:

the town of Isleworth
President Francois Mitterand

Sun Microsystems, the major supplier of workstations in the UK,

are all examples of 'Top-down grounding with an individual', the last two 'with distinguishing attributes'. All of the learning that occurs in FUNES is of sub-types/instances of known categories, thus it is all a form of Top-down grounding. Lacking the imagery component of Geller et al's work, it is open to the accusation of defining meaningless symbols with more meaningless symbols. However, it is questionable how much more meaningful is a pixel image like:

1111

of a pencil, than the simple lexical entry 'pencil'. Moreover, how to ground a concept like 'Sun Microsystems' is not a simple task (see footnote on page 75 for further discussion on this).

Although the motivations for the present work and that of Geller et al. are very different the similarities remain interesting.

The fact that the 'definitions' acquired by FUNES are so detailed in comparison to other PN acquisition work means that it may slot more easily into the paradigm of 'knowledge acquisition from text'. Some sentences in news stories do nothing else but impart descriptive knowledge about the actors they describe. In some ways, therefore, news text can be viewed as a PN lexicon or encyclopedia, and FUNES as a knowledge extraction system. Under this conceptualisation some interesting comparisons can be made between it and other systems that seek to build databases from their natural language analysis. We discuss some of these below.

## 11.6 A Comparison of FUNES and other Knowledge Acquisition Systems

The system described by Alshawi [7] is a toy database capture system, which is used as a framework in which his ideas about language analysis can be displayed and tested. It shows some interesting similarities to FUNES, in its results rather than the way it achieves them. The system processes simple sentences describing machines and museum artifacts into 'database creation statements', which could subsequently be translated into a request in a data manipulation language.

The system uses a memory (or Knowledge base) which represents knowledge of verb meanings through a system of specialisation (isa) and correspondence (has-a) links. The process of deriving DB creation statements from linguistic input is a multi-stage one. First, the sentence is parsed using Boguraev's ATN analyser [19], which produces a 'meaning form' based around the main verb and its cases (similar to the FUNES semantic case-frame).

Clause interpretation operations then apply 'memory' and 'context' mechanisms to these meaning forms. These mechanisms resolve language interpretation problems (such as anaphoric reference and semantic ambiguity resolution), and produce a 'predicate instance entity' in memory which represents the sentence, in terms of specialisation and correspondence entries. This is achieved by mapping the verb to its memory representation, and mapping the NP's which fill its agent and object slots to the specialised agent and object slots held with the verb in memory. From this a Database Creation Statement is built which utilises particular knowledge of the machine or artifact domains to create a statement in the desired form (i. e. utilising the correct database relation and column names).

For example, the verb 'manufacture' in the sentence 'Plexir manufactures P777' is resolved to the verb sense 'manufacturel' in memory. This would produce a DB entity (say E1) which is an instance of the verb 'manufacture'. The case labels delivered by the sentence analyser are specialised by the type of the verb, so 'agent' would be specialised to 'supplier/dbentity', and the entity 'Plexir' is asserted to fill the supplier/dbentity role for E1. Similarly 'P777' is asserted to correspond to the 'part/dbentity' of E1. From these memory assertions the final Database Creation Statement is formed:

(MANUFACTURES/RELATION ((M/MCNUM P777) (M/MNAME Plexir))).

FUNES analyses 'make' sentences in a similar manner, although its analysis is neither as detailed nor as refined, as one would expect since it is working in an unrestricted domain, whereas Alshawi's system operates in an artificial domain, in which the linguistic input is severely restricted and the knowledge base considered complete. In FUNES, when a 'make' sentence occurs with a corp PN as agent, a 'make' triple is entered into the lattr DB, e. g. '(plexir,make,p777)'.

In Alshawi's system, as with FUNES, different syntactic constructions can give rise to the same meaning form and memory representation. So 'P777 is red', 'The color of P777 is red' and 'the red P777' all mean the same thing, and all lead to a predicate entity :

(Corresponds: redl to E2 as machine/color/value to relp/machine/color)

This particular form is only used for the analysis of adjectival phrases and 'property' be phrases (as opposed to 'existence' be phrases). In FUNES, origin information is the only information that is commonly conveyed by adjectives. However we would not see statements like the above used to convey origin information, for in news text such information is not stated so explicitly. We could compare the variety of forms above to the variety of forms that could be used to give role information, e. g. 'George Bush is the President', 'President George Bush' and 'The President, George Bush'. All these statements would lead to 'isa' triples in the lattr DB. The corresponding representation in Alshawi's system would lead to 'specialisation' entries, such as :

(Specialisation: BUSH of president(instance)).

'Be' sentences and 'Have' sentences receive special treatment due to their close association with specialisation and correspondence links in memory. Thus, rather than mapping to corresponding memory representations (e. g. 'manufacture' maps to 'makel') 'be' sentences map directly to specialisation entries, and 'have' sentences to corresponds entries.

The special treatment of 'be' sentences is analogous to FUNES' handling of 'be' sentences (which includes appositives). These lead directly to 'isa' or 'role' entries in the lattr DB. 'Have' sentences do not feature prominently in news text.

A paper by Reimer [142] describes a system (called **wit**) that acquires terminological knowledge describing new data processing products. Interestingly Reimer describes text understanding as the chosen route towards automatic knowledge acquisition. The acquisition of knowledge about data processing products that Reimer's system performs can be compared to the acquisition of knowledge about human actors that FUNES performs. Both rely on a parser and semantic analyser to produce a semantic representation of the text. Both make use of predication within a text (i. e. A is a B), although in the texts analysed by **wit** this is explicitly stated, whereas in FUNES it is usually implicit in the form of appositives. Both use frame/slot formalisms for the representation of the concepts acquired.

**Wit** starts with very little domain-specific knowledge, only the top-level concepts in the concept hierarchy to be acquired are supplied (e. g. the concepts 'printer' and 'computer'). This enables the system to focus on those portions of a text which deal directly with these concepts. A new concept is created whenever a known concept occurs which is modified pre-nominally, e. g. if 'cartridge' is known, then 'ink cartridge' will lead to a new concept. In addition the system has knowledge of text coherence patterns for the type of text it is analysing. These deal with the implicit association of concepts within a text, and enable subsequent description of a concept that does not explicitly state its relation to the earlier concept to be utilised. This type of coherence is shown in the text below:

> 'The Deskwriter from HP is a new ink-jet printer. Ink is deposited from a disposable ink cartridge at a resolution of 300 dpi...

Here we are not explicitly told that the second sentence is describing the ink-jet printer from the first sentence, this must be inferred. Such text coherence patterns are also found in news text, and utilised by FUNES in referring PN's to previous descriptions (see chapter 8, section 8.4.4).

When **wit** has been run over several texts, a KB is produced consisting of concepts represented as frames, described by slots, and linked to other concepts by 'isa' links. A generalisation component runs over this KB to integrate the features of subcomponents into the superordinate component, to produce a generic KB entry for a particular concept.

Although little detail is provided in [142] on exactly how **wit** achieves its task, it does appear to be operating in a similar fashion to FUNES, with one major difference. This is that FUNES acquires knowledge merely as a by-product of its news text processing, whereas **wit** is run purely to produce a KB of terminological knowledge. However, both use intensive text understanding methods to achieve their knowledge extraction. I would argue that this is necessary for detailed knowledge acquisition, although less detailed knowledge may be gained through less intensive methods (cf. Hearst' approach described on page 22 and FUNES knowledge independent approach, section 11.7 below).

Finally, a paper by Castell and Verdejo [33] provides an interesting link between work on knowledge extraction and news text understanding. This describes a partially implemented system (COTEM) that seeks to read news text and build a Knowledge Base concerning the company activity described within the text. Its aim is therefore very similar to the SCISOR system of [140], only it is not at so advanced a stage. The interesting facility it considers that SCISOR does not is the inclusion of the companies it encounters in the news text within the Knowledge Base. This aspect of the work is very similar to FUNES, and the KB entries built by COTEM are also similar.

COTEM is a multi-stage system intended to consist of a parser, semantic analyser, processor and integrator. The first two stages are not implemented — the input at present consists of 'Verb Frames' which are like the Case Frames derived by FUNES, and other such processors ([96, 4, 90]). These frames are classified by the Processor which extracts the important information. This uses a detailed KB which holds a hierarchy of verbs (events) and objects (e. g. people and companies) encountered in news. All knowledge is represented in a frame-slot formalism with an inheritance mechanism. The same KB is used to hold both generic (definitional) items and instances of actual events and companies encountered in the text. The 'Intermediate Units' extracted by the Processor are handed to the Integrator which does some consistency checking and incorporates the items into the KB. Unfortunately the working of these two key stages is poorly described and it is hard to see precisely how they operate.

No consideration is given to the problems of parsing company names, but this is perhaps understandable as the paper focuses on the structure of the KB, the Processor and Integrator. Nor is the relation of the KB and the lexicon made clear. Although the KB is updated with the companies acquired, we are not told if the lexicon is. So the focus of the system in its present state is different from FUNES, which pays much consideration to the problems of parsing company names. It is hard to establish in the examples given if the company names are intended to be acquired from scratch, or if it is only additional information that is acquired and added to existing entries. An example news text in Spanish is shown, describing the acquisition of a French company, Filtrasol S.A., by Hunter Douglas. Additional information is given on Filtrasol, regarding its employees and sales. The relevant extracts are shown below:

> 'Hunter Douglas ha ampliado su grupo de empresas con la compra de dos companias, Flitrasol S.A., de Francia, y Stilsound Holding Limited, del Reino Unido. Filtrasol ocupa a una plantilla de 380 empleados y posee un volument de ventas de unos 4.200 milliones de pesetas anuales... '

This extract is informative in several aspects of foreign news, showing that the structure of company names is very similar, if not identical, in Spanish as it is in English and American. [8] The final KB entry constructed is:

| | |
|---|---|
| C2 INSTANCE | : COMPANY |
| NAME | : Filtrasol, S.A. |
| NATIONALITY | : French |
| HUMAN-FACTOR | : 380 |
| SALES-BY-YEAR | : IMP1 |
| OWNER | : (C1,PUR01) |

C2 is the identifier for this KB entry, IMP1, C1 and PUR01 are identifiers for further KB entries, describing the amount of sales, the owner company (Hunter Douglas) and the purchase instance which lead to Hunter Douglas owning this company. This representation is very like that employed in FUNES. Below we show a FUNES-like version: [9]

```
kbase([filtrasol,s,a]:
        [isa:company,
        origin:france,
        employ:380,
        year_sales:[4200,million,peseta],
        namehead:filtrasol,
        superpart: [hunter,douglas]]
```

FUNES would also acquire an entry for the namehead 'Filtrasol'.

Attempting a deep comparison of the two systems is not possible as the syntax and semantic stage of COTEM is not implemented or described. It would appear that COTEM

[8] It is interesting that orthography does vary considerably across languages as regards PN's, especially as many writers consider capitalisation to be one of the defining features of a PN [5, 115, 136]. Many languages do not utilise the case feature at all, while German capitalises all its nouns. Spanish (and French) do capitalise PN's, but they reserve this capitalisation strictly for PN's, and do not apply it, as does English, to origin PN's.

[9] FUNES could not derive the information on sales and employees at present, but given the input that COTEM is given, it would be simple to acquire such information.

is intended to construct the KB entries entirely from its 'Verb Frames' (which are delivered by the semantic analyser), whereas in FUNES the syntactic and semantic analysis produces both the Case-Frames and the Name-Frames. COTEM fills some slots using expectation rules based on the verb in the Verbal Frame. For instance the verb 'employ' will lead to its theme filling the 'human-factor' slot of the frame representing the verb's agent. This is facilitated by the detailed hierarchy of verbs contained in the KB. FUNES also fills some slots in this way, although, as explained in chapter 6 and 8, little consideration has been given to verbal definitions.

At this stage in the development of both systems all that can be said is that they present interesting similarities in their results. The motivations for each appear to be quite different though.

## 11.7 Towards a Knowledge-Independent PN Extractor

In its present form the FUNES system acquires PN's as a by-product of its processing of text. Therefore, to acquire PN's it needs to perform a full NLP-style analysis. This is due to the motivation behind the work — the need for NLP systems to be able to cope with the large numbers of PN that occur in news text. FUNES was not developed specifically as a PN acquisition system, but to demonstrate how a rudimentary NLP system could acquire PN meanings by using within-text descriptions in the process of fuller text understanding.

However, it would appear that certain areas of the Information Technology community would be interested in a system that could automatically acquire large numbers of PNs along with some kind of definition. [10] However, to utilise a full NLP system to do this would not be cost effective, would be far too slow, and would constantly have the problem of not being able to acquire a PN because it had failed in its parse or semantic analysis. Therefore, it is worthwhile to explore the possibility of building a system that could acquire PNs using a more knowledge-independent approach that would be faster and would not have to perform a full syntactic/semantic analysis of the text. The work of Miller (see above and [125], and Hearst [79]) present approaches to this problem. We have also carried out some (very) initial work on this front, utilising some of the FUNES PN learning techniques.

We initially extracted lines containing a variety of ROLE keywords from a 50,000 word corpus of AP Turkish news. Lines were extracted containing the words 'President', 'Minister' (including 'Prime Minister', 'Foreign Minister' etc.), 'leader of', and 'leader', followed by a capitalised letter. Using ROLE words as possible indicators of PNs avoids the problems of detecting the PNs themselves via initial capitalisation. Within a knowledge independent approach, using initial capitalisation produces problems with false positives from sentence initial words and capitalised words like time units (see [10, 38]).

The first step produced 280 lines of potential matches. In fact, every one of these contained a ROLE PN. The 280 lines were then passed to a program which utilised the FUNES role-KW patterns to automatically locate the PNs, categorise them, and extract role or origin information. This program output 43 personal PNs with role information, 23 of which also had origin information, and 5 location PNs (discovered through morphological analysis). All of the acquired PNs were correct. There were actually 41 unique names in the original grepped lines, but two were spelled differently on different occurrences, and lead to separate entries.

[10] This need was emphasized by the enthusiastic response from printed PN lexica researchers to the FUNES system at the 7th University of Waterloo Conference.

These results are very hopeful for such an approach, which could be extended to all classes of PN. The depth of definition produced is obviously less than those produced by the full FUNES system. Appositives, in particular, could not easily be handled by such a system, nor could distal references or verbal definitions. However, given extremely large text resources, a large number of PN's could feasibly be extracted in this way.

## 11.8 Summary

In this chapter we have described studies aimed at assessing the percentages of PN's in news text, and the percentages of these PN's which receive some sort of description therein. These have shown that almost all obscure PN's receive either an explicit (e. g. an appositive) or implicit (e. g. a self-describing corp PN) description. Therefore a system that starts out with a reasonably-sized lexicon can expect to perform well as regards its handling of unknown PN's. This hypothesis was tested by running FUNES on 200 unseen UK and US news stories. Consideration of only those PN's which were analysed showed that in the UK news (the area on which FUNES was developed) nearly 80% of PN's were correctly identified, and of these, nearly 90% had greater than half the differentia information present in the text correctly extracted. For a system that has been developed from scratch in a previously un-researched area, these figures are considered both impressive and illustrative of the efficacy of the approach to handling PN's that has been presented throughout this thesis. We have presented a description of other TU work that has considered PN's, and attempted some comparisons to FUNES. This comparison has been hampered by the different kinds of evaluation (or in some cases the lack of evaluation) used on these systems. Overall FUNES has been found to stand up well to comparable systems, and to offer a much broader range of coverage, in terms of PN categories handled, than other evaluated work. We have also utilised the evaluative studies to suggest areas in FUNES that are weak, and possible improvements in these areas. Because the analysis of PN's takes place within the context of general NLP processing, improvements to lexical, syntactic, and semantic units would lead to an improved performance on PN handling. Particular PN areas that would benefit from further work are:

1. the handling of conjunction, especially the interaction with apposition and the problem of ellipsis.

2. the handling of distal descriptions, e. g. anaphoric and definite NP reference.

3. the use of confidence factors for each contributing factor to a PN definition

Finally, we described preliminary work on the extraction of PN's from large corpora utilising a knowledge-independent approach. Although producing much less detailed definitions this approach would be faster and unhampered by problems of lexical, syntactic and semantic deficiency. It would therefore be more appropriate for the task of automatic PN extraction for printed lexica construction. However, unlike FUNES itself, such an approach contributes little to the work on robust text understanding.

# Chapter 12

# Conclusion

## 12.1 Summary

In this thesis we have discussed, and presented solutions to, the lexical bottleneck, with our main emphasis being on Proper Names. These are worthy of in-depth research for a variety of reasons:

1. They are poorly represented in dictionaries, and therefore represent a great source of unknown lexical data.
2. Their own internal structure can be highly complex.
3. They can occur in highly complex syntactic and semantic contexts.
4. There is little linguistic work describing the syntactic or semantic structure of other classes of PN beyond the personal name.
5. They are important in successful information extraction and topic classification.

These were the five points stressed in the introduction as composing the 'PN Problem', for which this thesis proposes solutions. We believe that all five have been met.

Concerning point 1 — poor lexical representation — we have demonstrated that poor coverage in dictionaries or in system lexicons need not be a problem, as PN's are very often described in the text in which they occur. This means that we can obtain a description of the unknown PN through a syntactic and semantic analysis of the accompanying Key Word or Within-Text Description (WTD). Even if there is no description, conventional word acquisition techniques can provide a reliable semantic category, as verb and preposition selectional restrictions are much tighter concerning PN classes than common noun classes. The WTD's describing a PN are frequently a complex part of the sentence. It is vital therefore that the processing system has a knowledge of the common syntactic contexts in which PN's occur, and the semantics of these contexts and the PN's themselves.

This brings us on to points 2 and 3. The formal model introduced in chapter 6 goes a long way towards addressing these problems. This model describes the most common classes of PN and their linguistic contexts in great detail, and includes a formal account of their syntax and semantics. It has been implemented in the news text understanding system FUNES, enabling PN's and their accompanying descriptions to be easily analysed. This involves detection and analysis of apposition NP's, conjoined NP's, attached PP's, and preceding and following Key Words. These processes can be used to produce lexical and Knowledge Base entries for those PN's which were unknown prior to processing.

We have also presented solutions to the problems of variant forms and name/noun ambiguity. Our system can refer all variant forms by applying knowledge of the heuristics controlling PN foreshortening, and utilising lexical knowledge of the attachment priorities of Key Words. Ambiguity can be resolved by a comparison of the semantic categories of the ambiguous word and an accompanying Key Word.

The model of PN's also contributes a linguistic description of many PN classes that have been relatively ignored in the literature, thus answering point 4. We have described both the syntax and semantics of these names, showing that very often no further descriptive material is needed to construct a definition, as they have a meaning of their own, in just the same way as normal words. This is particularly true of corp PN's, where often the name is formed entirely from non-name constituents and is completely self-describing (e. g. 'Ministry of Defence'). Object names and place names present an interesting problem in that their name may have a clear meaning (e. g. 'Abbey Wood', 'the Golden Rocket'), but this meaning is inappropriate, and does not describe the referent of the name (e. g. 'Abbey Wood' is a suburban town and not a wood, 'the Golden Rocket' is a train and not a rocket at all). Our work on PN foreshortening expands Carrol's [30] work on namehead formation.

All the above work contributes to a greatly improved handling of PN's in text understanding. This will permit improved performance in fact extraction and topic classification. We examine this final problem in the next section.

## 12.2 Applications

The above approach to the analysis of PN's means that they can be reliably detected in text, and reliably placed into a semantic category. This will aid greatly in the process of fact extraction and topic classification, an area of huge potential utility to businesses and governments (a potential utility fast becoming realised in [11, 93, 50]). Any system utilising the approach described here will be able to reliably detect locations of events, and important companies, people and pressure groups mentioned in the text under analysis. These are frequently among the most important items in a piece of text. The accurate analysis of each PN, and its descriptive material, will also mean that these sections of the sentence can be 'separated off' from the main, event-oriented, part of the sentence. For example, the following sentence is highly complex and discursive, but the effective analysis of each PN and its description permits the final Case-Frames for the sentence to be greatly simplified as shown:

> 'Mr Nicholas Biwott, Kenya's Industry Minister and an aide to Daneil arap Moi, was a prime suspect in the murder of Mr Robert Ouko, then Foreign Minister, a British detective said yesterday.'

```
past
    be
    [theme(biwott)]
    [theme(suspect),property([prime])]
    [during(murder),of_theme(ouko)]
pres
    be
    [theme(ouko)]
    [theme([foreign,minister])]
pres
```

```
be
[theme(biwott)]
[[theme(aide),works_for(moi)],[theme(minister),field(industry),
origin(kenya)]]
```

The lexical and Knowledge Base entries derived by FUNES from this example carry complete information on each actor. In the above example, origin information on all personal PN's is derived, as well as role information on the two individuals who are described in more detail.

The detailed analysis of differentia information that is possible with our approach means that all corp and personal PN's that are described in a text will have this information extracted and entered into their KB entries. It can then be used to aid in topic classification of a story. For example from the input:

> 'Alleghany Corp said it completed the acquisition of Sacramento Savings & Loan Association for $150 million ...New York-based Alleghany is an insurance and financial services concern.'

the area of operation (field) of Alleghany can be acquired. Subsequent mention of this company can provide a reliable indication that the story is dealing with financial news. A similar process occurs for people. If we know that someone works for a particular company, and know that company's field, or the person's field, we can gain an indication of the topic of the item. With places, knowledge of the superpart of a town can immediately give us the country in which the event described occurred.

Thus the detailed analysis of PN's contained in this thesis should improve these two important areas of NLP.

The methods outlined for dealing with unknown words, both common and proper, at a pre-processing and syntactic level will also facilitate the parsing of unedited text, as they enable a system to process such text successfully in the presence of a large number of lexical gaps.

The preliminary work on knowledge-independent PN extraction is regarded as very optimistic for automatic PN compilation from large text corpora. Such work could be of great assistance in the production of printed PN lexica and encyclopedias, such as [2, 163, 162].

## 12.3 Problems and Questions Un-Answered

In the last chapter we reviewed many of the areas we feel are in need of more investigation. Here we discuss some more theoretical considerations.

The first of these is the question of lexical update. Some preliminary thoughts on this subject are contained in [43]. We have pointed out many times that the wealth of within text descriptions means that the poor lexical coverage of PN's in dictionaries is not the problem it otherwise might be. If this is so, why then should we be concerned about updating our lexicon and KB with the acquired PN's. In theory, we need not bother at all, since a system equipped as is FUNES, can handle texts in which all the PN's are unknown. However, we must also consider practical issues such as speed and efficiency of processing. If we have lexical entries for all the words and compounds in a sentence we will be able to process it many times quicker than if we do not. This is the over-riding reason for lexical update. Additionally, knowledge of origin and corp PN's can help in the analysis of those NP's in which they occur as noun complements to a personal PN.

We must also consider at which point (if any) we stop updating. Given some of the statistics mentioned in this thesis, we might eventually end up with a lexicon and KB of several million PN's. While this might be desirable for printed PN lexica, in an NLP application we do not wish to search through such a vastness to find every word.

It is to this end that some sort of human monitoring of the lexicon seems necessary. For instance, it makes sense to retain firstnames, as this is a category of reasonable size (certainly much smaller than surnames), and one of great use in the detection of personal PN's. It also makes sense to retain famous, and therefore frequently occurring, people as compound entries in the lexicon, for the speed and efficiency reasons mentioned above. However, we do not wish to retain every individual we encounter, or else our lexicon will eventually hit the billions. What we have here is yet another version of the store/compute tradeoff. Whereas it is more efficient to store entries for people like 'George Bush', it will be more efficient to compute entries like 'Azare Mclintock' as we encounter them.

A second issue is the reliance on letter-case in the work as it stands. As this work is partly a description of the PN in English, and the capitalisation of the PN in English is one of its defining features, this seems excusable. However, in considering computational applications it may restrict the applicability of the work as some texts are still transmitted all in upper case. There are several reasons why we do not consider this a great problem. Firstly, the number of newswires that do transmit in all upper case is in the minority. Secondly, although we have no statistics to support this, the majority of electronic text we have encountered has been mixed case (in fact all of it except the MUC-3 corpus). So, a system reliant on mixed case will not be overly restricted.

Moreover, there are other strategies that can be used in dealing with PN's when the text is all in single case. If one has a large lexicon (at least 10,000 roots) then a commonly-used strategy is to assume any unknown that cannot be morphologically connected with a known word is a PN. The syntactic patterns described in this thesis could be utilised to support this approach, where instead of a potential PN being marked as a string of capitalised words, it would be marked as a string of unknown words. The Key Words and Within-text Descriptions could then be utilised in the same fashion as they are in FUNES. While this would not cope with PN's composed of totally known words it would cope with those containing unknown Proper Nouns. For corp and legis PN's composed of totally known words, the syntactic patterns described in Chapters 6 and 9 could again be used, in that a corp Key Word followed by a 'for/on/of' PP is a likely candidate as a corp PN.

The final point we wish to consider is one mentioned in Chapter 11 — the question of a knowledge-dependent vs. a knowledge-independent approach to PN acquisition and text understanding. As we mentioned in Chapter 11, FUNES relies heavily on syntactic and semantic analysis in its acquisition of PN's. However, this is because it was produced to examine the issue of PN handling within a text understanding system — it is not something that we are claiming is strictly necessary. Our point is, that if the syntactic and semantic facilities are there already, then they can be utilised to handle PN's. A system that is able to utilise such facilities will also be able to produce much better 'definitions' for PN's, than one that does not have such facilities (such as [79] or the knowledge-independent FUNES described in Chapter 11).

However, it is notable that some systems undertaking analysis of real text for the purposes of fact extraction are moving towards a partial parsing, or even skimming approach, [174, 95]. Might this undermine the validity of our work, based as it is on a fairly thorough syntactic and semantic analysis of the text? The answer is a clear no. This is because we have been at pains to produce a suite of methods, each applicable at a different stage of processing. This multiple strategy approach has been identified as extremely important in the analysis of news text (R. Kuhns, personal communication). The pre-processing stage identifies PN's, and gives them a syntactic category. The syntactic stage utilises accompanying KW's to produce a semantic category, and the semantic stage analyses appositives and PP's to produce differentia information. It would be possible to shift some KW analysis into the pre-processing stage, should it be wished to bracket PN's off earlier. D. Lewis (personal communication) described a to-be implemented PN detector that would utilise two passes through the input, one prior to syntactic analysis and one after. Such an approach would be very amenable to our strategy, being able to utilise pre-processing methods in the first pass, and syntactic methods in the second pass.

We would stress though, that the analysis of the more complex PN descriptions, such as conjunction and apposition inter-mingling, demands a deeper analysis than skimming can provide. However, the loss of the in-depth information provided in such constructions does not mean that a parse/skim should fail, or that a fact extraction system will not work. It simply means that the level of information will be less detailed. If we have a sentence like

> 'Rudolph Agnew, 55 years old and former chairman of Consolidated Gold Fields PLC, was named a non-executive director of Hanson PLC, the British industrial conglomerate.'

then the amount of information extracted can vary, and yet the sentence still be 'understood'. For instance, if apposition analysis failed we could just derive 'Rudolph Agnew was named a non-executive director of Hanson PLC'. If apposition analysis partially succeeded, we might also derive that Rudolph Agnew is 55, and Hanson is a British industrial conglomerate. What is important is that our methods permit that derivation of all types of information. The analysis of personal PN defaults would permit the classification of 'Rudolph Agnew' as a personal PN. The use of corp PN suffixes would permit the classification of 'Hanson' as a corp PN. Such minimal information could be extracted by the sparsest of parsers. Our description of verbal definitions would permit the derivation of the semantic information conveyed by the 'named' VP in the main sentence. This is 'medium-level' information, which would be extracted by most simple parsers, yet requiring deeper analysis than the simple use of a corp suffix. Our description of the nature of appositives, and the problems of interaction with conjunction, would permit the analysis of the conjoined NP's in apposition to 'Rudolph Agnew' and their attachment to him. The same applies to the apposition NP describing 'Hanson PLC'. The derivation of this information would only come from a deeper analysis.

In summary we feel that the thesis has made an important contribution to an under-researched area. We hope that it will, at the very least, serve to demonstrate the sort of problems Proper Names can produce in the text understanding process, and indicate possible approaches to overcoming them. As more and more real text is analysed, the importance of such work can only increase.

# Appendix A

# Other papers describing the FUNES system

[1 ] S. Coates-Stephens. *Expectation based word learning.* Technical Report TCU/ CS/ 1990/7, City University Dept Of Computer Science, 1990.

[2 ] S. Coates-Stephens. *A review of word learning and implementation of an inference-based word learner.* Master's thesis, City University, 1990.

[3 ] S. Coates-Stephens. *Automatic acquisition of proper noun meanings.* In Z. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems - 6th International Symposium, ISMIS '91.* Springer Verlag, 1991.

[4 ] S. Coates-Stephens. *Automatic lexical acquisition using within text descriptions of proper nouns.* In *Proceedings of the 7th Conference of the UW Centre for the New OED and Text Research: Using Corpora*, 1991.

[5 ] S. Coates-Stephens. *Lexical acquisition of proper nouns as a by-product of text processing.* In David Powers, editor, *IJCAI 1991 Workshop on Natural Language Learning*, 1991.

[6 ] S. Coates-Stephens. *The analysis and acquisition of proper names for the understanding of free text. Computers and the Humanities*, 26(5–6), 1993.

# Appendix B

# The Structure of the Lexicon

Words are stored alphabetically in the following format (pos=part of speech):

```
Word : def([pos1(Word,Info1),pos2(Word,Info2),...posn(Word,Infon)],
           [[2wordcompounds] ...],
           [[3wordcompounds] ...] ).
```

for example :

```
break: def([verb(break,trans,state_change,[[_]],[[object],[body_part],[abstract]]),
           verb(break,intrans,state_change,[[object]],[]),
           noun(break,[event],n)],
           [[break,out], [break,down]])
```

The nature of the Information held with a word differs for each part of speech, as does the number of arguments. Nouns are held in the lexicon as triples :

```
noun(Noun,Sem,Gender),
e. g. noun(food,[edible],n)
```

Verbs are held as 5-tuples :

```
verb(Verb,Trans,Sem_cat,Subject_Sel_Res,Object_Sel_Res)
e. g. verb(gather,comp,mbuild,[[human],[corp]],[]),
      verb(gather,trans,gather,[[human],[corp]],[[human],[object],[abstract]])
```

Sem stands for semantics. The most important element here is the HEAD, which gives the Semantic Category of a word. For nouns this includes 'human', 'abstract' or 'event', for verbs 'go' (verbs of movement), 'poss' (verbs of transfer of possession) and 'mbuild' (verbs of conceptualising). TRANS indicates transitivity, and can be trans (itive), intrans (itive) or comp (indicating the verb takes a sentential complement). Subject and Object Sel_Res are the selectional restrictions operating on a verb's subject and object. These are composed of lists of semantic categories. The empty list [] is used for non-existent restrictions, e. g. object restrictions for an intransitive verb; and an uninstantiated list [[_]] is used for those cases where no useful restrictions exist, the possibilities for subject or object simply being too great to list.

Compound words are indexed under their first component word, and checked for in the lexical look-up stage. When a word is found in the lexicon that has a list of compounds for which it is the first word, the following words in the input are compared to the possible compounds. The lexical entry for the compound is held separately. An example here is shown below :

```
gold:
def( [adj(golden, [made_of(gold)] )],
[[golden,handshake]] ).
[golden,handshake]:
def( [noun([golden,handshake],[abstract],n)] )
```

The compound system deals with compound nouns, phrasal verbs, and awkward collocations such as 'some of', and 'in order to'.

The other parts of speech covered in the FUNES lexicon are shown below, together with examples :

```
adjectives, cold: def([adj(cold,[temperature(low)])])
adverbs, quickly: def([adv(quickly,[with(speed)])])
determiners, the: def([det(the,[def])])
degree words, very: def([deg(very)])
pronouns, he: def([prn(he,[[animate],3rd_pers,sing,m])])
prepositions, with: def([prep(with,[with])])
modal auxiliaries, must: def([modal(must)])
have auxiliaries, have: def([have(have)])
progressive auxiliaries, be: def([prog(be)])
passive auxiliaries, pass(be)
```

# Appendix C

# The Structure of the Knowledge-Base

In its present state the Knowledge Base simply consists of a hierarchy for the semantic categories for nouns listed in the lexicon. This hierarchy is shown in figure C.1.

A facility is provided for moving through this hierarchy (with the has_attr predicate) to check for semantic consistency when applying verb selectional restrictions, and for other purposes. Various other sub-hierarchies are contained here, and are used for rapid checking of particular types of information, or for more precise information than is held in the fairly broad semantic categories of the lexicon. For example all the different Key Words used for describing Proper Names are held as sub-types of the type 'kw_descrip', and a more precise type of [object], that of 'artw' (artwork) is defined to include plays, films, novels etc.

A **view** predicate is also provided, which enables certain semantic categories to be viewed as members of other categories, even though they are not sub-types in the strict sense. For example, 'product' words such as 'petroleum' and 'fuel' are not corporation key words and yet in some contexts (when they are capitalised and preceded by other capitalised noun complements) they can function as such, indicating the presence of a corp PN, e. g. British Nuclear Fuels, British Petroleum. The **view** predicate enables them to be viewed as corp words in these situations while not being strict sub-types.

(PN's are also held in the KB, as described in chapters 7-10.)

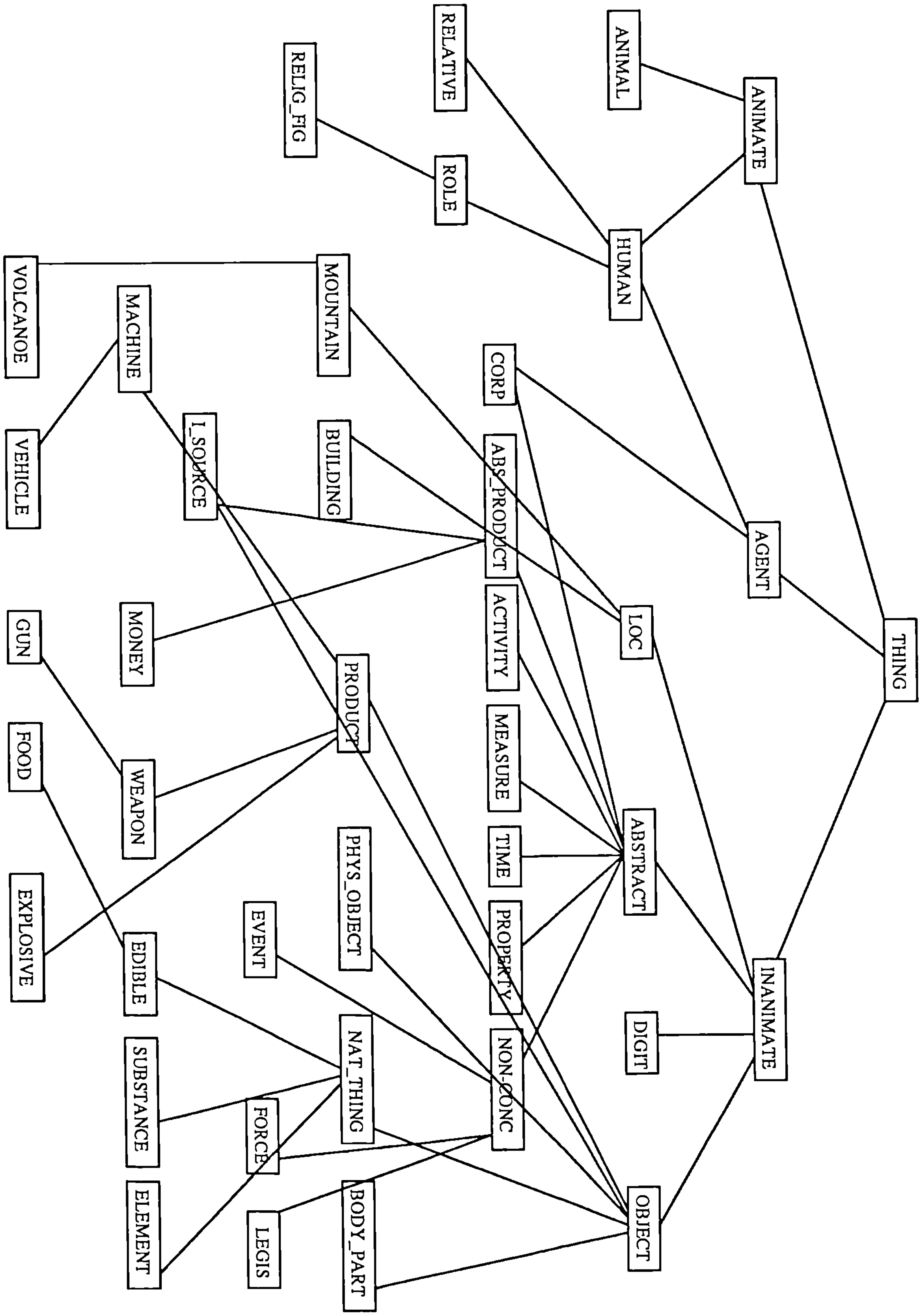


Figure C.1: The FUNES Semantic Hierarchy

# Appendix D

# Lexical Ambiguity Resolution

In this appendix we present the heuristics used for disambiguating noun/verb/adjective ambiguous words. The same heuristics are also used for classifying unknown words as to their part of speech. After presenting these heuristics we discuss the problem of closed class word ambiguity.

## D.1 Open Class Word Ambiguity Resolution

The heuristics used are applied in the order shown. If heuristic 1) should not apply or fail then 2) is applied and so on. The following abbreviations are used:

PW = Preceding Word
PPW = Second preceding word
NW = Next Word

```
1) Unk is 1st word in sentence:

- if unk has apostrophe s, then return as noun
- else if it ends in ‘ly’, then return as adv
- else if NW is ampersand, then return as corp PN
- else if ends in ‘ese’, return as adj
- else if ends in ‘can(s)’, ‘ian(s)’ and root is known loc
  , then return as loc PN
- else if ends in ‘shire’, then return as loc PN
- else if ends in ‘er(s)/or(s)’ and has known verb root
  then return as role noun
- else return as unknown noun.

2) Unk is hyphenated (see chap 4)

3) Unk is capitalised, return as PN

4) Check Morphology (see chap 4)

5) PW is comma or linkword
if unk ends in ‘ed/ing’  and (NW = det or linkword)
return as verb
```

```
else noun

6) PW is negative,
PW= ‘no’, return as noun
PPW= verb aux or noun, return as verb (noun possibility for cases like
‘for a novel not nominated)’
        else as noun
(could also be adj, e.g. ‘it was not risky’, but uncommon).

7) PW is adj, and adj not 1st word in sent
If adj is digit return noun (can have ‘5 died’ but uncommon)
PPW = adj-type verb, return as verb (‘appeared angry Unk’)
PPW = deg, if preceded by verb return unk as verb (‘seemed very angry Unk’)
        else as noun
   PW is adj and is 1st word in sent
NW is adj, return as adj
NW is verb return as noun
        else return as noun

8) PW is degree word
Unk is last word return as noun
else as adj

9) PW is det (not ‘that’)
NW is adj, return unk as adj
else as noun

10) PW is rel prn
Unk is last word, return as verb
NW is adv/verb aux (but not to/from) , return as noun
NW is prep/det/adj/prn return as verb
NW is init_cap return as verb
NW is noun (can be ambig) and unk ends in ed/ing return as verb
else return as noun

11) PW is verb aux
return as verb

12) PW is adv (very difficult)
NW is det/deg/adj/digit/prep
     /unk ends in ‘ed/ing’
then return as verb
else as adj

13) PW is prep
Unk is last word, or NW is prep
     return as noun
Unk ends in ‘ing’ return as verb
NW is rel prn return as noun
NW is adj return as adj
NW is noun/init_cap return as noun

14) if PW is personal PN/unk is 's, return as noun

15) if PW is N/Prn
PW is 's, if NW =adj return as adj
  else return as noun
Unk ends in 'ing' and NW is prep
  return as verb
NW is deg/adj/prn/neg/through/digit/det (not 'that')
  return as verb
Unk ends in 'ed'
  return as verb
Unk is last word return as noun
NW is prep/rel/verb
        OR NW is noun and Ambig Word is tenseless
  return as noun

16) PW is verb
Adj-type verb, return as adj
NW = adj, return as adj
PW is verb ending in 'ing', and Unk ends in 'ing'
return as verb
else return as noun

17) Return as noun
```

## D.2 Closed Class Ambiguity Resolution

1) Adj/Prep ambig (for 'early', 'next', 'late', 'later' etc). The main heuristic used is that if the next word is a noun of semantic type 'time', or a verb or another preposition then the ambiguous word is a preposition, else it is an adjective. A prior check is also made that if there is a previous word it is not a determiner or an apostrophed noun, if so the word is returned as adjective.

```
For 'Past' : if PW is det then AMB is adj else Prep
  (the past week vs drove past the town)
For others : PW is det/'s noun then AMB is adj
    Else if NW = time noun/verb(can be ambig)/prep
                 (e.g. next year, who later reported, early on).
    Then AMB=prep
    Else adj.
```

2) Aux/Verb Ambig: If the next word is a negative or an adverb it is skipped. For 'have' and 'be' a check is also made that the tense of the auxiliary and the following verb agree. Such checks will decide, for example, that in 'they were murders', 'were' is a main verb and 'murders' a noun, whereas in 'they were murdered', 'were' is an auxiliary and 'murdered' the main verb.

```
NW is adv/neg skip
NW is Amb verb: Aux is be, agree_tense(Aux,Verb) , AMB=aux
else AMB=verb
           : Aux is have, Verb=be, AMB=aux
OR agree_tense(Aux,Verb) AMB=aux
Else AMB=verb
     : Aux is do, AMB= aux
NW is verb : AMB =aux
NW is det/deg/adj/digit/noun/prn/prep/init_cap AMB=verb
NW ends in ing, AMB=prog(Aux)
Aux is be and NW is abb/init_cap AMB=verb
Else NW ends in ed/en AMB=pass(Aux)
Else AMB=verb
Aux is have, NW is compound, AMB=verb
Else NW ends in ed/en AMB=aux
Else AMB=verb
Aux is do AMB = aux.
```

```
3) Deg Ambig
For 'Own': is V/Deg ambig, if PW is poss determiner (my, her etc) AMB=deg
Else = verb
For 'some', 'many', 'more', 'most' , are Prn/Deg ambig:
NW is aux/verb/prep (not of)/rel prn OR AMB is last word
Then AMB=prn
Else AMB=deg
For 'about' and 'around', are Prep/Deg ambig:
NW=time noun AMB=prep
Else if NW = number AMB= deg
Else AMB= prep
For 'half' and 'quarter' : N/Deg ambig
NW is aux/verb AMB=noun
Else AMB=deg
```

(Although really always nouns, it is much easier to analyse them as degree words in expressions like 'half the members voted for ... '. The collocations 'half of' and 'quarter of' are recognised as such and just treated as 'half' and 'quarter').

```
4) Prep/Modal Ambig
if NW is a verb
then word is a modal
else if NW is ambiguous
if one possible case is an 'ing' verb
and word is 'of' or 'for'
then word is modal
else word is modal
else if NW is a verb auxiliary
      then word is a modal
else if PW is a sentential complement verb
      and NW is not a noun group word nor init\_cap
      and word is 'to' or 'for'
      then word is a modal
```

```
else word is prep

5) Prep/Det Ambig (for 'this')
NW = noun AMB=prep
NW = aux, AMB= pronoun
Else AMB=det

6) Prn/Det Ambig (for her, each etc)
For her, AMB is last word OR NW = prep/det AMB=prn
 Else AMB=prn
For each/some, NW=noun/adj/digit AMB=det
 Else AMB=prn
For another, AMB is last word AMB = prn
Else NW = noun/adj/digit/unknown  AMB=det
Else AMB=prn

7) Modal/Noun (for may/might)
AMB is last word or init_cap AMB=noun
NW=aux/verb AMB=modal else Noun
```

8) Adjective Ambiguity: Little adjectival ambiguity is recognised. Any word that can be an adjective or a noun is simply held as a noun. In this way it will still be processed whether it actually occurs as a noun or an adjective, since in the former case it will be analysed as the head noun (or a noun comp), and in the latter it will be processed as a noun comp. If such words were held as adjectives occurrence in a text as a noun would cause the parser to fail as it would not be able to find a head noun (e. g. if 'republican' were returned as an adjective, then the NP 'a staunch republican' would not parse). Such a strategy enables FUNES to side-step the issue of noun/adjective ambiguity. The question of verb/adjective ambiguity (i. e. many words are both past tense or progressive verbs and adjectives, such as 'boring') is avoided by allowing past and progressive form verbs to act as adjectives within the NP parse.

# Appendix E

# The FUNES Grammar

The grammar specified here outlines the strings accepted by FUNES. It does not describe the restrictions on these strings, in terms of agreement, attachment, or semantic acceptability. Lacking these restrictions (which are implemented directly into the parser), it could generate parses that would not actually be accepted by FUNES, due to, for instance, an incorrect attachment. Its use is simply to give an idea of the weak generative capacity of the system.

| | |
|---|---|
| () | represents optional constituents |
| * | represents arbitrary repetition of that constituent |
| (m) | means that constituent can occur in any position in that rule |
| / | means or |

S → S1 S1
S → S1
S1 → (LW) (PP)* NP VP
LW → and/or/although ...

NP → Prenoun NG (AMP) (Postnoun)
NP → (NEG) (DEG_P) Pronoun (Postnoun)
Prenoun → (NEG) (DEG_P) (DET_P) (DEG_P) (ADJ_P)
AMP → & PN
Postnoun → comma Appos comma
Postnoun → PP*
Postnoun → Rel_clause
Postnoun → and NP

DET_P → Det*
DET_P → Number of (Det)
Det → the/a/some ...
Number → [0-9]*/ adj(X,[number])

NEG → no/not
DEG_P → Deg DEG_P
DEG_P → Deg
DEG → most/some/even ...

ADJ_P → Adj1

```
ADJ_P → (Adj1) Measure
ADJ_P → Adj1 and ADJ_P
ADJ_P → Adj1 comma and ADJ_P
ADJ_P → Adj1 comma ADJ_P
ADJ_P → Dig_P
Adj1 → Adj*/Adv*/Origin_noun*/Verb1*
Origin_noun → noun(_,_,_,[human,origin,_],_)
Verb1 → verb(_,_,ing,_,_)
        verb(_,_,fin(_,_,past),_,_)
Adj → small/foreign/violent ...
Adv → really/extremely/ ...
Measure → Measure_noun
          {Condition: NW=due/directly/News_dir }
Measure_noun → mile/square-_/metre/kilometre
New_dir → north/south/east/west/north-east/north-west/
          south-east/south-west
Dig_P → Digit
Dig_P → (US/C) $ Digit
Digit → [0-9]

NG → (Noun)* Noun
NG → Noun
NG → (Noun)* (Det) Noun
Noun → president/country ...
Noun → [0-9]*
Noun → Roman_N
Roman_N → X/V/I/ Roman_N
Roman_N → X/V/I/


PP → Prep (Prep) NP
PP → today/tomorrow/yesterday (morning/afternoon/evening)
PP → early/earlier/later/later (NP)
PP → Month
Month → January/Jan/February/Feb/March/Mar/April/Apr/May/June/
        Jun/July/Jul/August/Aug/September/Sept/October/Oct/
        November/Nov/December/Dec
Prep → from/for/of ...

Rel_Clause → prep Rel (PP) NP VP
Rel_Clause → Rel/Prog_Verb/Sent_mark (PP) VP
Rel_Clause → Rel/that (PP) NP VP
Rel → who/which/where ...
Prog_Verb → verb(_,_,ing,_,_)
Sent_mark → modal(to)/modal(for)/that


Appos → (adv) NP
```

```
VP → PreVerb Verb_P PostVerb
PreVerb → (Adv(m)) (NEG(m)) (modal) (have) (pass) (prog) (do)
Verb_P → Verb (Adv) (NEG)
PostVerb → (PP)* Comp_verb
PostVerb → (PP)* Obj_P
PostVerb → (PP)* Adjv_P
Comp_verb → whether to VP
Comp_verb → VP
{ Condition : NW= to/for/of/verb(_,_,ing,_,_) }
Comp_verb → whether/that NP VP
Comp_verb → []
{ Condition : main verb was passive and not bitrans }
Comp_verb → NP Comp_verb
{ Condition : NW= that }
Comp_verb → NP VP
Obj_P → (NP) (Adv_P) (NP) (PP*)
Adjv_P → (NEG) (DEG_P) adj/verb(_,_,ing,_,_)/
verb(_,_,fin(_,_,past),_,_)
Adv_P → adv/deg/comma however comma
Verb → attack/say/warn ...
```

# Appendix F

# Parsing of Noun-Phrases

The process of parsing a NP is illustrated in the flow chart shown in figure F.1. This is simplified in many ways, but gives a high-level picture of the process of parsing an NP.

The NP is the most complex constituent in the FUNES grammar, reflecting the importance of nouns (agents, locations, companies etc) in news text. Below we briefly discuss some of the main constituents and how they are handled.

## F.1 Pre-Noun Constituents

### F.1.1 Determiners

The grammar for the determiner is relatively straightforward. The only complication is the inclusion of constructions like 'One of the ...'. These are also allowed to take the place of a determiner, to parse constructions such as 'One of Peru's ...'. The test_det predicate implements the grammar for determiners. It is passed the list of unparsed words by the np predicate and inspects it to see if the head item is 'det(Det,Sem_type)'. If so it returns this item in the two variables 'Det' (to hold the actual determiner) and 'Sem' to hold the semantic type (def, indef, poss etc.) If not it just exits with these variables uninstantiated.

### F.1.2 Negatives and Degree Words

Negatives are words such as 'no', and 'not'. Degree words are basically a subset of adverbs. An arbitrary number of these can occur in any one NP. As can be seen from the grammar rules contained in Appendix E they can occur both before and after a determiner. Thus we could have 'EVEN the most hardened criminal ...', or 'the MOST expensive properties ... '. Many awkward constructions are handled by being processed as degree words, e. g. 'most of', 'some of', 'less than', 'more and more', 'at least', and 'up to'. The grammar rules for degrees and negatives are implemented by the predicates test_deg and test_neg. These both work in a similar fashion to test_det. Test_neg will set a global NEG flag for the current syntactic level if it does detect a negative. This is subsequently checked by the semantic analyser when it constructs the case frame for that level. If it is set, the case-frame is prefixed by 'neg'. Test_deg will call itself recursively in case there is more than one degree word.

### F.1.3 Adjectives

The Adjective Group within a NP can be very complex, as the grammar describing it shows. Like all the grammar rules in FUNES, those for ADJ_P have been derived through

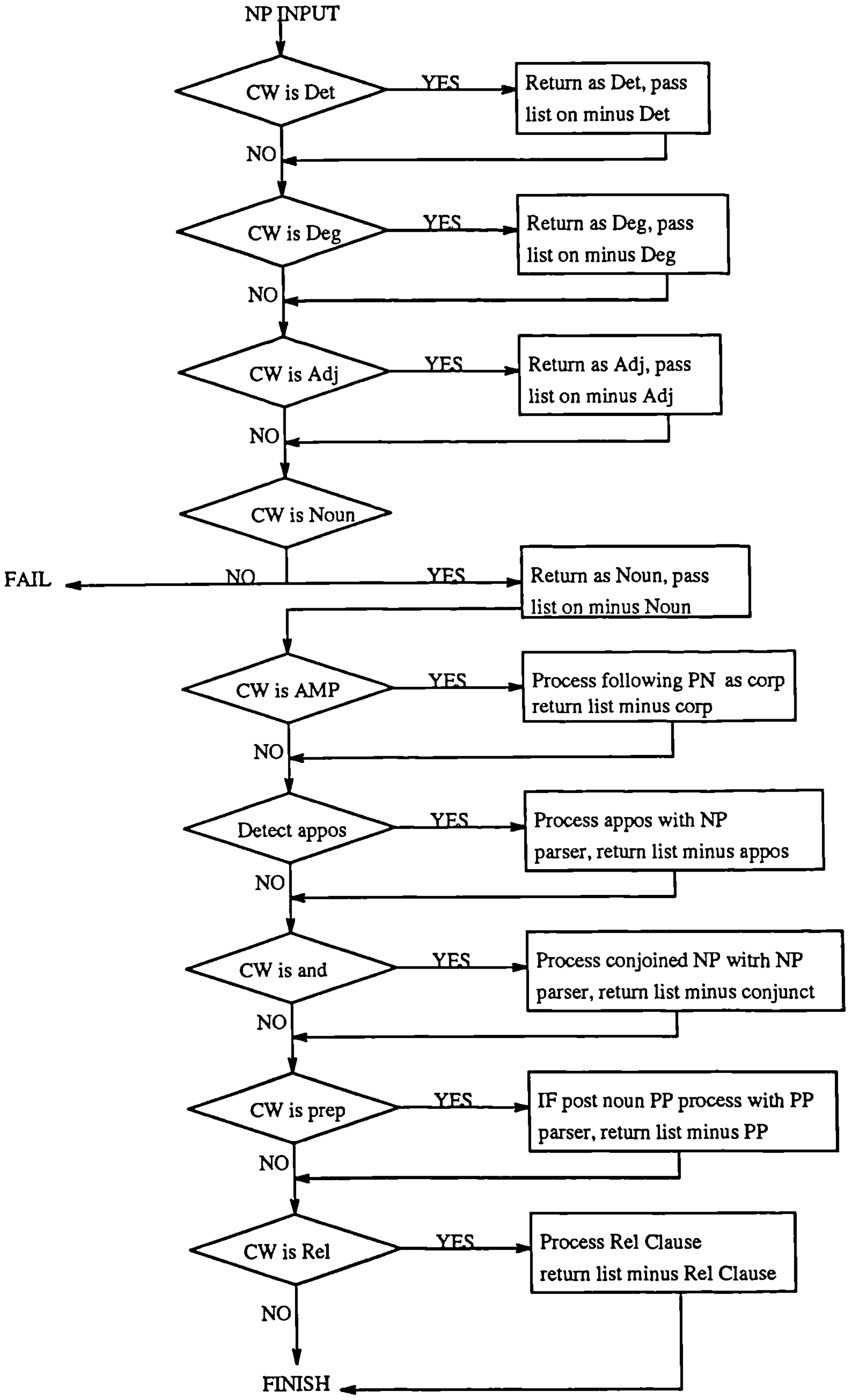

NP INPUT
CW is Det
YES
Return as Det, pass list on minus Det
NO
CW is Deg
YES
Return as Deg, pass list on minus Deg
NO
CW is Adj
YES
Return as Adj, pass list on minus Adj
NO
CW is Noun
FAIL
NO
YES
Return as Noun, pass list on minus Noun
CW is AMP
YES
Process following PN as corp return list minus corp
NO
Detect appos
YES
Process appos with NP parser, return list minus appos
NO
CW is and
YES
Process conjoined NP witrh NP parser, return list minus conjunct
NO
CW is prep
YES
IF post noun PP process with PP parser, return list minus PP
NO
CW is Rel
YES
Process Rel Clause return list minus Rel Clause
NO
FINISH

Figure F.1: Noun Phrase Parsing

the processing of many news stories, and their complexity reflects the fact that what many journalists consider perfectly acceptable grammar diverges considerably from that found in simple grammar books. As can be seen from the rules, adverbs and past tense or progressive verbs are allowed to function as adjectives, as well as words actually held as adjectives in the lexicon. As examples, we could have 'some forces remained EXTREMELY loyal to the President', 'the use of DEPLETED uranium bombs', and 'a LEADING human rights lawyer'. This facilitates the problem of lexical ambiguity resolution and permits a sparser lexicon. Words held as origin nouns (such as Kenyan, American, etc. ) can also function as adjectives, provided they are followed by another adjective or noun (if not they will be the head noun).

The Measure phrase is to handle directional expressions such as '200 miles south of ...'. Here the number and measure noun are treated as adjectives and the directional word as the head noun. As discussed in previous chapters such phrases are important in the definition of place PN's.

Finally, digits are also analysed as adjectives (unless they are the head noun). A special procedure processes money phrases like '$200' or 'US$35'. This removes the 'US' or the 'C' and the dollar sign and inserts the word 'dollar' after the digit.

The adjective grammar rules are implemented in the predicate test_adj. This is a recursive predicate like test_deg, as an arbitrary number of adjectives can occur in a NP.

### F.1.4 The Noun Group

The main problem in processing a noun group is the determination of its end when two NP's run together. Various heuristics are used in deciding when to split a noun group. The main context in which this problem occurs in news text is in an initial PP. Such phrases invariably give location or temporal information, e. g. 'In July George Bush announced ... ', 'In Croatia troops were ordered ...'. So if we are in an initial PP and have a time or loc noun, the noun group is ended. The temporal indicators 'yesterday' and 'today' can also run into a previous noun group, and so must be checked for (as must the collocations 'yesterday morning', 'yesterday evening' etc.) Apostrophe s, and dates also need special consideration. The former case was discussed in chapters 9 and 10 with reference to the ambiguity it can cause when occurring with a PN.

The problem of adjoining noun groups following a bitransitive verb, so beloved of linguistic discussions, has not been looked at, due to its infrequency of occurrence in the news text examined.

The noun group grammar rules are implemented in the test_noun predicate. This has two definitions, one for handling the head noun, the other recursive predicate for handling noun complements. This checks the look-ahead buffer (the list of unparsed words) to see if the NW is indeed a noun before recurring. If the NW is not a noun it means we are at the head noun, and the recursive predicate fails, and we move to the second predicate that analyses the head noun. The nouns parsed are returned as a list, together with their case, semantic category, number and gender, and a flag indicating whether they are known or not. This information is all used to fill the NP register.

### F.1.5 Pronouns

FUNES utilises a history list mechanism for the location of pronoun referents. This holds all the nouns from the present and previous sentence. The first noun that matches the pronoun on number and gender is selected. Certain grammatical constraints on reference are alse employed. Nouns from the same sentence which occur on the same syntactic level as the pronoun (unless the pronoun is reflexive) are disbarred as possible referents. In addition no attempt is made to deal with more complex referents such as events.

One problem never considered in the large body of work on pronoun reference (see [84] for an excellent summary) is how to locate a referent if some of the previous words are unknown. In this case much of the commonly used information for making a reference decision, such as gender, number, and semantic category, is not available. In addition when one has to deal with apposition, one is often faced with the problem of two potential referents that are in fact the same thing. To cope with the problem of unknowns for which gender and number information may not be present, several passes through the history list are made. If no exact match is found the nearest unknown is taken as the referent.

It has been noted that in news text the vast majority of anaphoric references are to humans. Given this the process of locating referents could be made more efficient (at some small cost in accuracy) if all non-human/role nouns (except perhaps corp nouns) were excluded from consideration. This step has not been taken in FUNES though. One reason for this is the use of the history list at other stages in processing, where it is extremely useful in locating the preceding NP.

The grammar rule for pronouns is implemented in a separate np predicate from the noun np predicate. If this predicate should fail we backtrack and move to the second np predicate for parsing det-adj-noun type NP's. The pronoun np predicate returns the noun that has been located as the referent if it was found, else it just returns the pronoun itself.

When the np predicate exits, having parsed an NP, the appropriate NP register is filled, using the variables returned by test_det, test_adj and test_noun.

## F.2 Post Noun Constituents

When the NP register has been filled, the NP parser continues and checks for the occurrence of post noun constituents — appositives, conjunctions, PP's, and relative clauses. The handling of appositives and conjunctions was described in detail in chapter 8, and will not be dealt with again here. Instead we expand the description of PP's and relative clauses provided in chapter 3.

### F.2.1 Post-Noun PP's

As stressed throughout this thesis, FUNES was primarily designed to investigate PN problems. Given the time constraints on its construction many issues in NLP system design have been given only superficial treatment, the main desire being to get a system up and running. The guiding criteria have been aimed at producing a system capable of analysing a large number of realistic texts, at the cost of complete correctness in the analysis, in order to permit the examination of a wide range of PN constructions. The handling of PP attachment is one of the areas that has not been explored in depth. FUNES simply uses a few general attachment heuristics and leaves it at that. These heuristics make use of semantic information to decide if the PP will form a valid semantic construct if attached to the NP. If a wrong attachment is made it can not be undone. Surprisingly perhaps, given the attention usually given to PP attachment, these simply heuristics have proved very successful.

The heuristics FUNES uses are based on the type of preposition, and the letter case and semantic class of the nouns concerned. A PP is attached to the preceding noun in the following cases:

i) If the preceding noun is the subject noun or was in a string of PP's the first of which was attached to the subject noun
ii) If the preposition is 'of', 'as' or 'between'
iii) If it is 'for' and followed by a PN
iv) If it is 'to' or 'from' and the verb is not of sem_cat MOVE, POSS or SET_FREE
v) if it is 'on' and the previous noun is a corp or abstract PN.

These heuristics are used to actually decide whether to parse a PP from with the NP parser. The test_prep_rel predicate handles the parsing of post noun PP's. It is handed the list of words returned by test_and (the conjunction handling procedure), and checks if the NW is a preposition. If it is, it then uses the above heuristics to decide if it is a PP that should be attached to the preceding noun. If it is such a PP the pp predicate is called, if it is not then the predicate just exits (as it does if the NW is not a preposition).

As might be concluded from an examination, some of the above heuristics are aimed very much at dealing with corp and legis PN's (e. g. 'Committee on X', 'Forum for the Restoration of X'). The vast majority of 'of PPs' attach to the noun, those which do not are mostly clearly indicated by a particular verb ('accuse of', 'die of' etc). The choice of noun attachment for 'between' is purely due to the nature of the news text examined, where the large majority of cases appear to attach to the noun rather than the verb, as in 'fighting between ...' or 'talks between ...'. The attachment of 'as PP's' to the preceding noun facilitates the processing of the role information that they very often convey. This is an example of how much of the processing in FUNES is structured to facilitate the handling of PN's.

The success of these simple heuristics lends support to a feeling emerging in Computational Linguistics that many issues that have been covered in detail in the NLP arena, purely because they were covered in detail in linguistics, are not very common or important in the analysis of real text. A corollary to this, is that many phenomena that are common and important in real text, such as compound nouns, apposition, PN's, dates and figures, have been relatively neglected in linguistics. Such a feeling has been echoed by Tomita and Tsujii in [164, 165] and stated very clearly by Jacobs in [92]. It has also been surfaced in the area of NL interfaces [25, 53] and is evident in the current focus on large text corpora as a source for the study of language, rather than linguistic introspection.

### F.2.2 Relative Clauses

The final constituent checked for within the NP parse is the relative clause (RC). This can basically be split into three types:

1. Subject missing, e. g. 'The Iranian oil tanker Avaj2, which broke down in the Channel ... '
2. Object missing, e. g. 'The Committee, which President Bush described as ... '
3. PP missing, e. g. 'for control of the hill, from which the army could attack Manerplaw'.

Type 1 cases are by far the most common in the news examined during this thesis. As described in chapter 3 the RC is usually flagged by the occurrence of a relative pronoun or sentential complement marker. These types can easily be detected using the look-ahead buffer. However reduced RC's can not, as their detection requires an arbitrary far lookahead, as the past tense verb could be the main verb or the relative clause verb. This is shown in the example below:

> '... who said the rocket fired from an ambulance parked near his house in the north-eastern town of Dera Bugti had caused extensive damage. '

Reduced RC's are the source of many famous (notorious) garden path sentences, e. g. 'The horse raced past the barn fell'. Inability to process past tense reduced relatives was one of the main sources for parse failure in the FUNES evaluation tests.

The processing of an RC is illustrated in the flow chart in figure F.2.

The test_rel predicate implements the grammar rules for RC's. It receives the list output by the test_prep_rel predicate which processed any post noun PP's. Firstly it checks the look-ahead buffer for a preposition and a relative pronoun. If found the NP and VP parsers are called to parse the 3rd type of RC above. If not found, the buffer is checked for a relative pronoun, or a progressive verb with no progressive auxiliary or a sentential complement marker (in which case the main verb, if already processed, must not take a sentential complement). If found the VP parser is called to process a VP relative clause. As subject missing and object missing clauses can be preceded by the same relative pronoun, it may well be that the VP parser has been called when in fact there is an NP next, as we really have an object missing clause. If this is the case, the VP parser will fail and we will move on to test for an object missing clause.

To do this the buffer is checked for a relative pronoun or the word 'that' (where again the main verb must not be one that takes a 'that' complement), and the NP and VP parsers called. A flag is added to the VP call to indicate that the object will be missing. When the NP or VP calls have finished, and the relevant phrases have been parsed the missing constituent is located (see below) and the semantic analyser called to analyse the RC. When test_rel exits it returns the list of words it received minus the RC, and nothing else. The whole of the RC has now been dealt with, and only exists now as its case frame. When the main sentence is analysed semantically it will be as if it never contained an RC.

After processing the RC, the missing subject, object or PP must be located. In many cases this is the noun immediately preceding the RC marker. Depending on the nature of the RC this will be inserted as subject or object or PP. However if the immediately preceding constituent was a PP then the missing constituent may be the noun within this PP, or it maybe the main noun.

The location of the moved constituent is achieved by examining the registers of analysed phrases. The constituent immediately preceding the RC is first located, and if it is not a PP it is automatically selected as the referent. If however it is a PP, FUNES uses animacy constraints to check if this phrase can be the correct referent. It compares the sem_cat of the relative pronoun (either animate or inanimate) to the sem_cat of the preceding noun, using the has_attr predicate. If they match, the preceding phrase is selected, if not the previously parsed phrase is located, and that is checked for animacy consistency. The first one that matches is chosen. If none match the immediately preceding phrase is selected as a default. When the RC occurs after a progressive verb or a sentential marker there is no animacy setting for comparison. Left as a variable this will simply match the first constituent it is matched against (i. e. the immediately preceding one). This is not a perfect solution, it would be better to utilise the selectional restrictions of the verb, and ensure that the sem_cat of any candidate nouns matched these. This would be relatively simple to implement.

When any RC's have been parsed, the final action carried out by the NP parser is to check for the presence of any parsed appositives, and if any are found, to attach them and call the semantic analyser. When this is done the NP parse is complete.

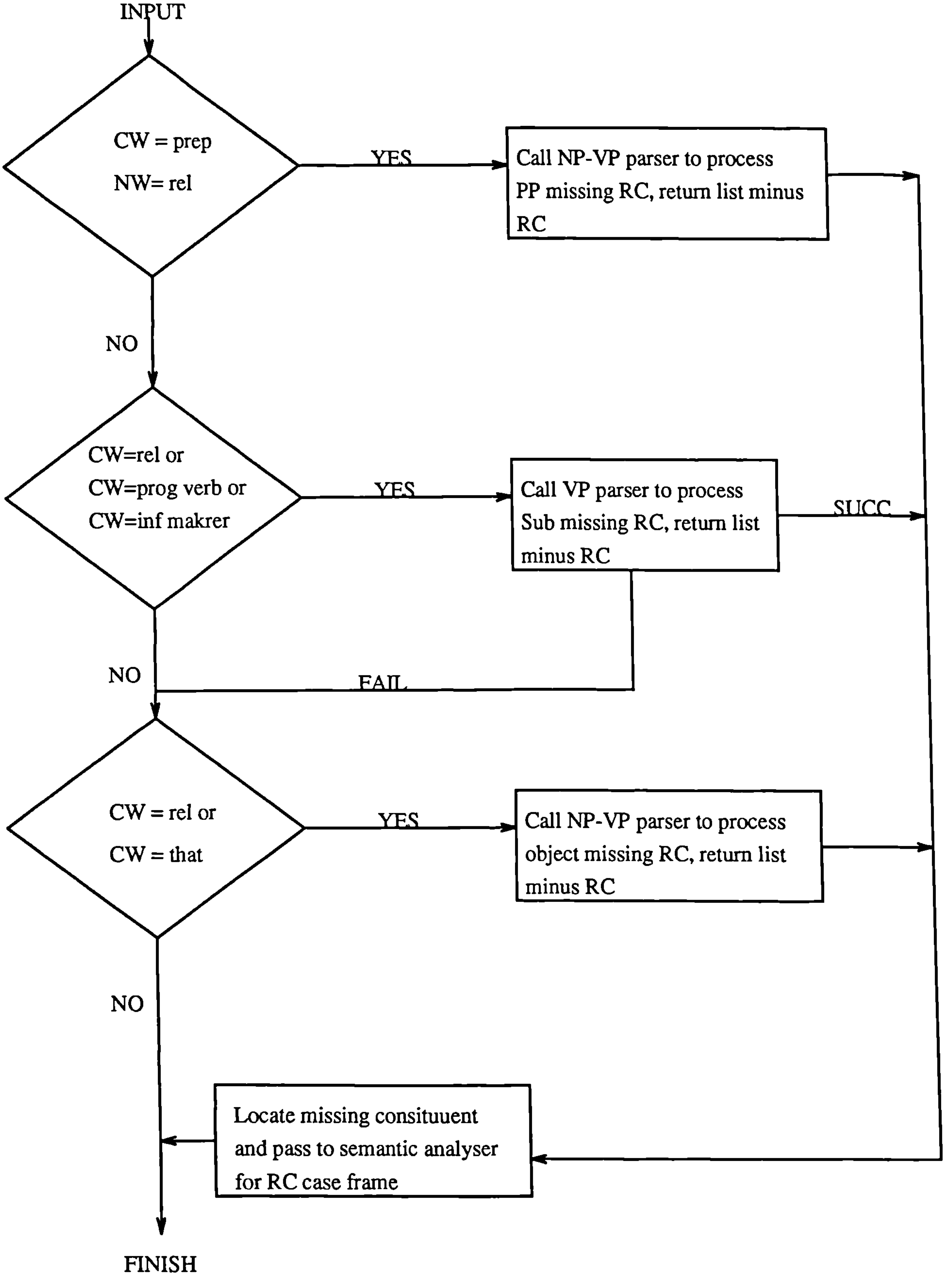

INPUT
CW = prep
NW= rel
YES
Call NP-VP parser to process PP missing RC, return list minus RC
NO
CW=rel or
CW=prog verb or
CW=inf makrer
YES
Call VP parser to process Sub missing RC, return list minus RC
SUCC
NO
FAIL
CW = rel or
CW = that
YES
Call NP-VP parser to process object missing RC, return list minus RC
NO
Locate missing consituuent and pass to semantic analyser for RC case frame
FINISH

Figure F.2: Relative Clause Parsing

# Appendix G

# Parsing of Verb Phrases

VP's are parsed by the vp predicate. Its action is similar in some ways to the np predicate, i. e. it moves through the list of words handed to it searching for a verb. On the way to the verb it processes pre-verb complements which indicate tense and mood. Once the verb has been parsed, post-verbal complements are handled, and, as with the NP, these frequently involve recursive calls to the NP and VP parser(s). We begin by examining the verb and its preceding complements.

## G.1 Pre-Verb and Verb Constituents

The Pre-verb structure is comprised of the verb auxiliaries — modals, have, be and do. Any of these can be followed by a negative or an adverb. The modal auxiliary (might, may, would etc. ) conveys mood, and nothing else. Such information is not used in FUNES. The 'have' auxiliary conveys tense and number information. FUNES only recognises three 'tenses', — present, past and progressive — and the last two of these are indicated by the ending on the main verb, so the tense information from this auxiliary is not returned. The reason for this sparse treatment of tense is that FUNES has no need to deal with it more adequately, as its processing only extends to the delivery of a semantic form. Were the processing to be extended to a higher level, where the overall relationship of a story's constituent sentences had to be considered, more detailed tense handling would be necessary. The information required for this is present, it is simply not fully utilised. Number information is returned as it may not be indicated by the main verb. The 'be' auxiliary can be a progressive or a passive (or both). Passive information is returned as it is only conveyed by the auxiliary, progressive information is not as it is also conveyed by the main verb ending. Number information is returned if not already known. The auxiliary 'do' appears to be somewhat less related to the others, in that it can not occur with them. It returns number and tense information if it is not already known.

The grammar for all the above is very simple and all are handled by very similar predicates — test_modal, test_have etc. These all receive the list of unparsed words, test if the first word in this is the appropriate one ('modal(might)', 'prog(be)' etc. ), and if so remove it and return the new list and a variable to hold the auxiliary, and if not simply exiting and returning the list unchanged. If the item in the list contains tense or number information and this is required it is also returned. Each of these predicates also call test_adv and test_neg which parse any adverb or negative constituents.

The main verb is parsed by the test_verb predicate in a similar fashion to the above. This returns tense and number information if not already known. The TRANS setting contained with the verb is also returned, but may later be overwritten depending on the presence or absence of an object. As with auxiliaries the main verb can be followed by an adverb or negative.

## G.2 Post-Verb Constituents

The flow chart in figure G.1 shows a simplified version of the processing of post-verb constituents. In this section we shall expand on the description of embedded sentences provided in chapter 3.

After parsing the main verb, if its TRANS setting is comp, or if the NW is 'that' or 'to', the comp_verb predicate is called. This first checks the look-ahead buffer for the presence of embedded sentence markers such as 'to' or 'that'. If found the level is incremented and the VP parser, or NP and VP parsers, are called to parse the embedded sentence. Upon completion of the parse, the moved constituent, if there is one, is located and the semantic analyser called to derive the case frame of the embedded sentence. Then the main verb register is filled with the main verb, and accompanying information, and the vp predicate exits. Below we show some examples of sentences with clearly marked embedded sentences:

i) 'Zaire is reported **to** have closed its two border posts with Uganda'.
ii) 'Saudi Arabia is considering **sending** its entire surplus stock of wheat
to the USSR'.
iii) 'President Mobutu Sese Seko announced **that** he would ... '
iv) 'Saudi economists said the nation had not yet decided **whether to** donate...'

If there is no marker, and the verb is COMP we are faced with the problem of how to determine if we have an embedded sentence or a simple object. We could have:

i) They wanted the car
ii) They wanted the car to start
iii) They saw the plane
iv) They saw the plane explode
v) The organisers told the crowd that they should disperse

The NP parser is called to parse the NP that directly follows the verb. Having parsed this NP (hence NP1) the next word is examined, and if it is 'that' comp_verb is called to parse the embedded sentence. This would be the situation in example v) above. If not, the vp predicate is called. The success of failure of this predicate will indicate whether NP1 was the subject of an embedded sentence (as in 'they saw the plane explode') or just an object (as in 'they saw the plane'). If the latter is the case the call to vp will fail. If it does fail, backtracking is prevented and the parse continued as if NP1 were a simple object. As NP1 was entered into the main sentence object-NP-register, it will be automatically recovered for semantic analysis.

Before the semantic analysis of an embedded sentence can occur the missing subject must be located if the embedded constituent was a simple VP, rather than an NP + VP. The missing subject can be either the main sentence subject or object. If there was no object in the main sentence then it will be the subject, as in 'Michael decided to leave'. If there is an object in the main sentence then this is the subject of the embedded sentence, as in 'Michael asked Peter to leave', i. e. it is Peter who will be doing the leaving. The missing constituent is located by calling up the appropriate NP register, and copying the contents into the subject NP register at Level+1.

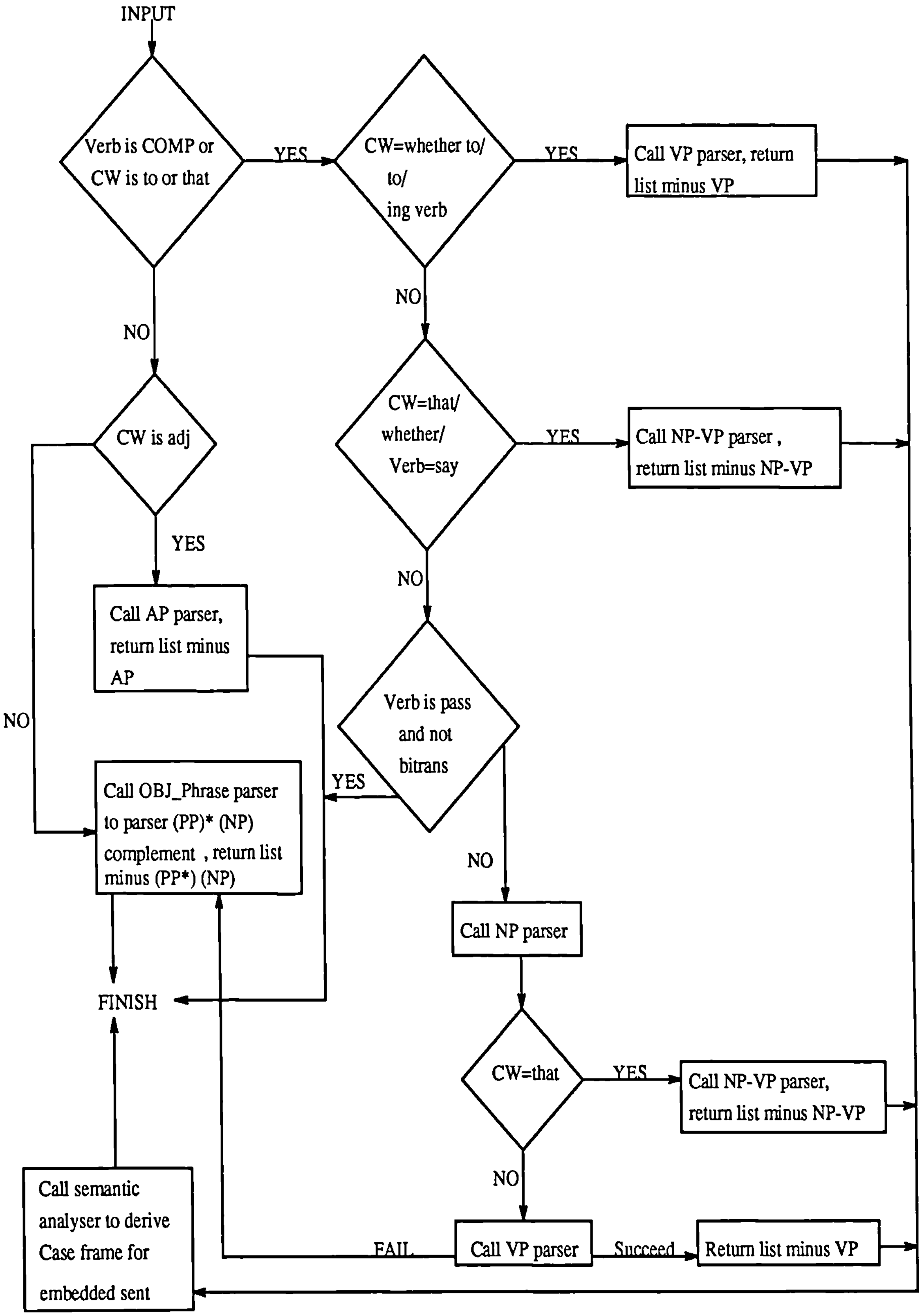

INPUT
Verb is COMP or CW is to or that
YES
CW=whether to/ to/ ing verb
YES
Call VP parser, return list minus VP
NO
NO
CW is adj
CW=that/ whether/ Verb=say
YES
Call NP-VP parser, return list minus NP-VP
YES
NO
Call AP parser, return list minus AP
NO
Verb is pass and not bitrans
YES
Call OBJ_Phrase parser to parser (PP)* (NP) complement , return list minus (PP*) (NP)
NO
Call NP parser
FINISH
CW=that
YES
Call NP-VP parser, return list minus NP-VP
NO
Call semantic analyser to derive Case frame for embedded sent
FAIL
Call VP parser
Succeed
Return list minus VP

Figure G.1: Post-Verb Phrase Parsing

The handling of simple NP, AP, and PP post-verb complements was described in chapter 3, and will not be expanded on here. Having completed this account of VP parsing, we next turn to look at more general cases of conjunction.

## G.3 Conjoined Constituents

As previously mentioned the emphasis in the design of FUNES has been the NP, and its internal structure, especially with reference to PN's. The relationship of NP's and VP's within the overall sentence structure has not received as much study. This is reflected in the treatment of conjunction. Conjunction of NP's is a problem that has received much work, as described in preceding chapters, due to its bearing on the PN problem. However conjunction of NP's and other constituents, and conjunction of sentences has not been looked at in depth.

The only parsable conjunction beyond NP + NP is sentence + sentence. FUNES can not parse conjunction of different constituents, such as Sentence + VP. Various types of sentence conjunction are permitted

i) although/while/when/because/if/so/unless S1 S1
ii) S1 although/while/but/when/because/if/and/or/plus/so/then/unless S1

When such link words occur at the end of a sentence the processing of conjunction is quite straightforward. Detection of the link word leads to the setting of a global CONJ flag, which is checked for upon completion of the sentence parse. If on, the sentence parser is called to parse the conjoined sentence. If the conjoined element is in fact a VP it can not be parsed. [1]. If the word 'and' occurs after a NP, FUNES assumes the conjunct is another NP and will call the NP parser to process it. If this is not the case the parse will terminate after the parse of the conjoined NP.

When the link word occurs at the start of a sentence the situation is a little more complicated as it may result in problems determining the end of a noun group. As before, such an occurrence causes the setting of the global CONJ flag, which is checked for upon completion of the first conjoined sentence. However when we have an example like the following

> 'Although the arms were a great boost to the government forces unrest continued throughout the area'.

FUNES is not able to break the noun group in the appropriate place, and the second conjunct will not be parsed. A way round this would be, if the CONJ flag were on to force an arbitrary break in the noun group, to at least permit some sort of parse of the second conjunct. However to perform correct splitting of such adjoining noun groups is still a major problem and is beyond the scope of this thesis.

## G.4 Prepositional Phrases

This topic was described in appendix F, but there the focus of discussion was on the question of attachment. Here we consider the nature of the PP itself. This is a relatively

[1] It would be a trivial matter to enable FUNES to handle such conjunctions. It has been a constant temptation to continue expanding the syntactic processing of the system. This has had to be resisted as syntactic processing is not the focus of the work, yet it is so open ended that it would be all too easy to devote far too much time to it. We have insisted that once a sufficient level had been reached to permit the analysis of a large number of stories, the syntactic processor would not be further extended

simple, although extremely common constituent. It is also notable for the fact that it can occur in a large number of places within a sentence. Fortunately its occurrence is usually explicitly marked by the occurrence of a preposition. The PP is typically composed of a preposition (or two) plus an NP. Thus for the most part it is processed in the same way as an NP. However the types of PP encountered in real text are somewhat more varied than this simple construction.

In particular, the words 'today', 'tomorrow' and 'yesterday' are awkward, in that they are single words which function as a whole temporal PP, meaning on or at that particular time. They are held in the lexicon as prepositions. The pp predicate checks for them, and if found, returns the preposition 'on' before calling the NP parser with a slightly modified word list. The item 'prep(today,[at_time])' is removed and the item 'noun( today, word, known, [time], [sing,n])' added. Similar considerations apply to the words 'somewhere', 'elsewhere', 'early', 'late', 'earlier' and 'later'.

Finally, in US news in particular, the abbreviated forms of months are often used without prepositions as time PP's, as in 'an announcement is due Nov 21'. Therefore a PP is also allowed to start with a month name.

As any number of PP's can occur together, upon exiting the call to the np predicate, the list returned by this predicate is checked to see if it starts with a preposition, if so a recursive call to the pp predicate is made.

# Appendix H

# Specifications for the Derivation of Differentia Case Labels

This appendix specifies the conditions under which each of the differentia slots described in chapter 6 are formed. For the most part these slots correspond to the cases returned by the pre or post-nominal complement analysers. They are almost a subset of these cases, but for the following differentia slots:

supertype
role

which are not members of the case set employed in FUNES.

For each slot we describe the situations which lead to its entry in the lattr DB, and the values for its first and third arguments. The following abbreviations are used:

PPnoun : the head noun of a PP
Prevnoun : the head noun of the preceding PP, or the head noun of the NP to which the PP is attached if there are no preceding PP's.
Sem cat : the semantic category of a noun or verb. We use the abbreviated expression 'Noun=Sem cat' to mean that the noun has that particular semantic category
Det : determiner, can be [def] (definite) or [indef] (indefinite)
Prep : preposition

## H.1 The Role Case

A Role triple is entered in the lattr DB in the following situations:

1. a role KW occurs in a NG followed by one or more init_cap nouns. In this case the head noun of the NG forms the first argument of the lattr triple, and the role KW the third argument, e. g. Chancellor Helmut Kohl → (kohl, role, chancellor). A proviso is that the role KW must be the rightmost KW in the NG.

2. a role/relative/assoc KW occurs in an appositive NP and the main NP is composed of capitalised nouns (or vice-versa), e. g. Helmut Kohl, the German chancellor, → (kohl, role, chancellor)

3. an Acting_as case is returned by a PP module. Here the theme of the clause is used as the first argument of the triple, (if there is no theme the preceding noun or subject

are used) and the case structure argument forms the third argument, e. g. Helmut Kohl has been elected as chancellor → (kohl, role, chancellor).

4. a personal PN occurs with no descriptive KW, and a previous descriptive KW has occurred with no accompanying PN. Here the previous KW provides the third argument of the triple. For example:

   > 'The German Chancellor arrived in Paris for crisis talks on the state of European monetary union. Helmut Kohl will meet the French Finance Minister, ...'

   → (kohl, role, chancellor).

An Acting_as case is returned by the PP modules:

If prep = as.

## H.2 The Isa/Supertype Case

(These two terms are used interchangeably throughout the thesis and here). An Isa triple is entered into the lattr DB in the following situations:

1. a loc/object/event/isource KW occurs as headnoun in a NG proceeded by one or more capitalised nouns. In this case the capitalised nouns, and possibly the KW, form the first argument in the isa triple, the KW the third argument.
2. a loc/object/corp/legis/isource KW occurs in a NG followed by one or more capitalised nouns. In this case the situation is the same as for role words above.
3. a loc/object/corp/legis/isource/event KW occurs in an appositive NP and the main NP is composed of capitaised nouns (or vice-versa).
4. a 'Name' case is returned by a PP module

A Name case is returned :

If prep = of and
PPnoun is init_cap and
there is no indefinite determiner and
Prevnoun is one of :
enclave, state, village, region, town, city, province, island, isle, republic.

When a Name case is returned the PPnoun forms the first argument of the isa triple, and the preceding noun the third argument.

## H.3 The Works_for Case

A Works_for triple is entered in the lattr DB in the following situations:

1. a works_for case structure is returned by a PP module. Here the first element of the triple is the head noun of the NP to which the PP is attached, or, if this part of a descriptive appositive, the first argument is formed from the PN which the appositive describes. The third element the case structure argument, e. g. 'Sir Allen Sheppard, the chairman of Grand Metropolitan,' → (sheppard, works_for, [grand,metropolitan]).

2. a works_for case structure is returned by the Ncomp analyser. Here the first element of the triple is the head noun of the NP in question, or, if this NP is describing a PN in an appositive relationship, then the first argument is the PN itself. The third argument is the case structure argument, e. g. 'Mecca Leisure chairman Michael Guthrie' → (guthrie, chairman, [mecca,leisure]).

3. a VP containing 'replace/take over from' is analysed. If the theme has a works_for triple entered in the lattr DB this is withdrawn and transferred to the Agent.

A works_for triple is returned by the PP modules in the following situation:

```
If prep = in and
        PPnoun = corp OR
        VP attached PP where PPnoun is init_cap and Vtype=mtrans and Det
        present (e.g. 'writing in The Times').
If prep = on/at and
        NP attached PP where PPnoun = corp
If prep = under and
        Prevnoun = role and PPnoun is init_cap
If prep = of and
        Prevnoun = role/human and
                PPnoun = corp/isource
                OR PPnoun is init_cap and Prevnoun selects for corp/isource
                OR PPnoun is abb
If prep=to and
        Prevnoun= role and PPnoun = human
If prep = for and
        NP attached PP and PPnoun = corp
        OR Default
        OR VP attached PP and Prevnoun = role
If Prep=with and
        Prevnoun = role OR PPnoun case is abb
        OR VP attached PP and PPnoun = corp
```

```
A Works_for triple is returned by the Ncomp analyser:
If the head noun is a personal PN
        AND the last but one Ncomp is corp/isource or can be Viewed as corp
        or has no semantic cat (i.e. is unknown).
OR the head noun is a role noun
        AND the last Ncomp is corp/isource or can be Viewed as corp
        or has not semantic cat.
```

## H.4 The Run_by Case

A Run_by triple is entered when the PP modules deliver a run_by case, e. g. 'Dallhold is the family holding company of Australian financier Allan Bond' → (company, run_by, bond). This triple will be picked up in name-frame compilation, and added to the supertype slot in the Dallhold name-frame, to give:

(dallhold, isa, [company,run_by([allan,bond])]).

A Run_by case is returned if:

If prep = of and Prevnoun = corp/isource and PPnoun = human

## H.5 The Boss_of Case

A Boss_of triple is entered when a VP is analysed containing a verb of sem_cat 'control', e. g. 'Mr Simmons owns Valhi Inc,' → (simmons, boss_of, [valhi,inc]).

```
If V sem_cat = control
then if Corp returned from Object analysis enter triple (Agent,boss_of,Corp)
        else enter triple (Agent,boss_of,Theme).
```

## H.6 The Related_to Case

A Related_to triple is entered if the PP modules return a related_to case, e. g. 'Sadie Lawrence, aged 84, mother of Ivan Lawrence, ' → (mother, related_to, lawrence(2)). This information is added to the role slot for 'sadie lawrence' in name-frame compilation to give:

(lawrence, role, [mother, related_to([ivan,lawrence])])

The related_to case is returned

If prep = of and Prevnoun = relative

## H.7 The Origin Case

An Origin triple is entered in the following situations:

1. An origin case is returned by the PP modules. Here the arguments are selected as we described in the Works_for triple. This will hence be referred to as 'the standard fashion'.

2. An Origin case is returned by the Adj analyser. Again the arguments are selected in the standard fashion.

3. If a PN has no origin triple by the name-frame compilation stage, then one is formed from the origin terms found in the origin DB.

An Origin case is returned by the PP modules in the following situations:

```
If prep = of and
        PPnoun = loc and not Prevnoun = loc
        OR PPnoun is init_cap/abb and Prevnoun is university/treaty
        OR Prevnoun = role/human and PPnoun is loc name
                                 OR PPnoun is init_cap and Prevnoun selects
                                 for loc
                                 OR PPnoun is init_cap and Prevnoun is unknown
                                 and no det and Prevnoun has role entry in
                                 lattr DB
        OR Prevnoun = corp/isource and PPnoun is loc name
```

```
                                       OR default if PPnoun is init_cap
If prep = from and
        PP is sentence initial and not PPnoun = time
        OR PPnoun = loc

An Origin case is returned by the Adj analyser :
If any of the adjectives are in the origin DB
        AND not headnoun = loc
        AND adjective is not directional type.

A word is entered into the Origin DB
If - it has sem_cat [human,origin,Loc]
   - it has sem_cat [from(loc)]
   - it is unknown and init_cap and ends in 'ish/ese'
   - it is unknown and ends in -can(s)
     or -ian(s) and the root is a known place PN
   - it is hyphenated and word-two is 'based' or 'born'
   - it is unknown and init_cap and a compound beginning with 'New',
     'North', 'South', 'East' or 'West', and it is not the head noun.
   - it is unknown and init_cap and preceded by 'north', 'south', 'east' or
     'west' and is not the head noun.
   - it is unknown,init_cap, and occurs as a Ncomp to a place KW, which
     is followed by 'of'. (e.g. the Croatian port of Split).
   - it is the subject of the sentence and a place PN
   - it is returned from a PP module as origin(Loc)
```

This is essentially saying if the word is a known origin PN, or revealed as origin by morphology or accompanying KW's, or is a place PN and the subject, or is returned as an origin PN by a PP module, then enter into the origin DB. These conditions have been derived through long testing, and found to give the best results. Places giving potential origins can be given in other ways (e. g. in 'to' or 'at' PP's), but these may not be reliable at giving potential origins for those actors not explicitly given an origin.

## H.8 The Assoc Case

An Assoc triple is entered in the standard fashion if the PP modules returns a Co_Agent Case. This is returned:

```
if prep = of and
        Prevnoun = assoc
        OR Prevnoun = human/role and PPnoun = human name
If prep = with and
        NP attached PP and PPnoun is init_cap (tested after works_for case)
        OR Vtype=mtrans and PPnoun= corp/human
        OR PPnoun=human/relative/role/loc
        OR PPnoun case is init_cap
If prep = between and PPnoun = human
```

## H.9 The Superpart Case

A superpart triple is entered in the following situations:

i) A PP module returns the case at_loc or superpart
ii) The Adj analysis returns the case 'superpart'
iii) The appositive analysis returns the case 'superpart'

An At_loc case is returned:

```
if prep= in and
        VP-attached PP and PPnoun = loc
                       OR default if PPnoun is init_cap
If Prep = on/at and
        NP-attached PP and PPnoun = loc
                       OR default
        OR VP-attached and PPnoun is inanimate and not (abstract or event)
                          OR PPnoun is init_cap
If Prep = close to
        and PPnoun = loc OR init_cap
If Prep = over
        and PPnoun = loc OR init_cap
If Prep = within
        and PPnoun = loc or init_cap
If Prep =  around/inside/outside/near/through
        is default case.

A Superpart case is returned:
if Prep = of and
        PPnoun is PN and Prevnoun = loc
                                   and det is indefinite
                                   OR Prevnoun not one that indicates 'Name' case
        OR PPnoun is PN and Prevnoun = subpart
        OR Prevnoun = corp/isource
                    and PPnoun is corp OR Prevnoun is subsidiary/division/arm

The decision on whether the Superpart triple is describing the main NP
or the directly preceding PP is difficult. We have used the following
heuristic:
       if Prevnoun is unk loc use Prevnoun
       else if Main noun is unk loc use Mainnoun
       else if Prevnoun is known loc use Prevnoun
       else if Main noun is known loc use Main noun
       else if Prevnoun is known corp use Prevnoun and a 'based_in' triple
       else use Prevnoun

A Superpart case is returned by the Adj analyser if:
       any of the adjectives are in the Origin DB
       and they are not directional
       and the headnoun is loc
```

```
A Superpart case is returned by the Appositive analyser if:
       both the main NP and the appositive NP are init_cap
       and (both are unknown OR one is a known loc)
       and the appositive NP has no determiner
```

## H.10 The Direction Case

A Direction case is formed in two situations:

i) by the PP modules
ii) by the appositive analyser

Direction cases never actually form triples on their own, always being combined in a Location triple with an 'of' triple' or a 'distance' triple.

A Direction case is returned by the PP modules:
If Prep = to and PPnoun is directional (e. g. 'to the south of')

A Direction case is returned by the appositive analyser when the headnoun of the appositive is directional (e. g. '30 miles south of ... )

### H.10.1 The Distance Case

A Distance case is formed if:

a distance case is returned by the appositive analyer.
This is the case if:
the headnoun of the appositive NP is a measure noun.
Distance case-frames are formed thus:
[distance(measure(Measure_type)]
They may have a number component added if the measure noun was preceded by a digit (e. g. '30 miles from ...), in which case the final form is:
[distance(measure(Measure_type),number(Digit)], e. g.
[distance(measure(mile),number(30)]

As stated above distance case-frames combine with direction case-frames to form Location triples.

### H.10.2 The Location Case

A Location triple is formed from the combination of distance, direction and of case-frames, in the following situations:

1. a distance case-frame is returned by the appositive analyser. This may be combined with a direction case-frame returned by the PP modules.

2. a direction case-frame is returned by the PP modules. The first argument of the Location triple is either the main NP (in an appositive case) or the Subject (when the direction case is returned from a VP-attached PP), and the third argument is the case-frame formed from the distance and direction case-frames.

## H.11 The Composed_of Case

A Composed_of triple is formed when the PP module or the Ncomp analyser returns a composed_of case. The PP modules return a composed_of case

```
If Prep = of
        and Prevnoun = object and PPnoun = substance
        OR Prevnoun = corp/isource and PPnoun is plural
                      and PPnoun = corp/human/role/object/loc

The Ncomp analyser returns a composed_of case
If the corp PN analyser has activated
        and one of the Ncomps = role

The Ncomp analyser activates
If i) headnoun is corp/isource or can be viewed as corp
   ii) or headnoun is role noun and last Ncomp fulfills i) or has no sem cat
   iii) or headnoun is personal PN and penultimate Ncomp fulfills i) or has
        no sem cat
```

If the PP modules have returned the composed_of case then the 1st argument of the triple is the entire PP corp PN (which may not yet have been totally formed), and the third argument the PPnoun. If it was the Ncomp corp PN analyser which returned the case then the 1st argument is the corp PN returned by the Ncomp corp PN analyser.

## H.12 The Product Case

A Product triple is entered when the PP module returns a Product case or when the NG KW analyser enters a product triple directly:

```
The PP modules returns a product case
If Prep = of
        and Prevnoun= author/director/producer and Det = def
        OR Prevnoun= maker/producer/seller/manufacturer/distributor
                     and Det = indef

The NG KW analyser returns a product triple
If headnoun is object KW
        and Known Corp in NG
        OR KW is artwork type
        OR NG contains serial number
```

If the PP module returns the case then the main NP forms the first argument, as such a construction invariably follows the pattern 'X, a maker of Z,', and the PP head noun as the third argument. The NG KW analyser uses the known corp or unknown init_cap Ncomp as the first argument, and the KW as the third argument.

## H.13 The Field Case

A Field triple is entered when the PP module returns a Field case, or when the Ncomp or Adj analysers return a Field case.

The PP modules return a Field case

```
If Prep = of
        and PPnoun is init_cap and Prevnoun = legis (origin already checked for)
        OR Prevnoun = role and PPnoun = abstract/event
        OR Prevnoun = corp/legis and PPnoun = abstract/abs_product/event
Prep = to
        and Prevnoun = role and PPnoun not (= human)
Prep = for
        and NP-attached PP and Prevnoun = corp
                           OR PPnoun = abstract/event
                           OR Prevnoun = role
Prep = on/at
        and NP-attached PP and Prevnoun = corp/abstract/event
```

```
The Adj/Ncomp analyser returns a Field case
If headnoun = role and Ncomp = abstract and the Corp PN analyser failed
If Corp PN analyser fired and Ncomp is abs_product/food/substance/object
```

In all three situations the first argument of the triple depends on the sem cat of the preceding PPnoun (for PP module) or headnoun (for Adj/Ncomp). If it was corp then the field PP is part of the corp PN and the whole PN will form the first argument, if it was role then the standard procedure is followed.

## H.14 The Made_by case

A Made_by triple is entered directly from the KW analysis procedures in the syntactic stage, when a 'Corp PN Object PN' pattern is detected. This occurs when:

1. an object KW is headnoun, preceded by at least one unknown capitalised word, and a known corp PN or an all upper case word or a corp KW preceded by at least one capitalised word.
2. a known corp PN occurs followed by an unknown capitalised word.

## H.15 Agent, Theme, Instrument and Amount

Unlike the above heuristics these return cases for subject and object NP's, based upon the semantic category and transitivity of the verb, and the semantic category of the head noun, as shown below:

```
If Verb sem cat is 'be'
   then if noun sem cat is 'measure' or 'number'
           then CASE=amount
           else CASE=theme
   else if TRANS=intrans
           then if there are PP's
                   then if noun cat is 'human' or 'corp'
                           then CASE=agent
                           else CASE=theme
                   else CASE=theme
   else if TRANS = adj
```

```
        then CASE=theme
else if TRANS = trans
        then if noun has_attr animate or loc
                then CASE=agent
                else CASE=instrument
```

# Appendix I

# Example Inputs at Each Level of Processing in FUNES

In this appendix we show two sentences at each stage of their 'development' as they pass through the FUNES system. We have utilised two sentences, as we could not find a single sentence which exhibited all the different facilities of the system.

## I.1 Pre-Processing

Sentence 1, which we track through the pre-processing stage is shown below:

> "Nancy B", who suffers from the Guillain-Barre syndrome, had said her life had become meaningless (Our Foreign Staff writes).

This is input to FUNES from a file in exactly this form. The output from the tokeniser is :

```
[(nancy,init_cap,_52),(b,init_cap,_194),(who,word,_290),(suffers,word,_423),
(from,word,_600),(the,word,_731),(guillain,init_cap,hyphen),(barre,init_cap,
_1064),(syndrome,word,_1240),comma,(had,word,_1455),(said,word,_1576),(her,
word,_1707),(life,word,_1828),(had,word,_1959),(become,word,_2107),(meaning
less,word,_2298),(our,init_cap,o_brak),(foreign,init_cap,_2657),(staff,init
_cap,_2845),(writes,word,_3021)]
```

(Terms composed of an undersore and a number, e.g. _194, are simply Prolog's way of representing an uninstantiated variable). The output consists of a list of triples. The first element in a triple is the input word in lower case. The second element gives information on the case of the word as it occurred in the input. 'Init_cap' means that its initial letter was capitalised (i. e. it may be a PN); 'word' means it occurred as a normal word; and 'abb' means that its first two letters were capitalised, a short-hand way of checking for acronyms. The third element holds hyphenation or bracketing information. The hyphenated word 'Guillain-Barre' has been processed as two 'words', with the hyphen information carried with the first word. The bracketed words '(Our Foreign Staff writes)' are tokenised in exactly the same way as the rest of the input, the fact that they are within brackets is shown by the o_brak accompanying 'our'. Normally the closing bracket is shown by a c_brak setting with the word following the closing bracket. However when the bracket is the last thing in a sentence there is nothing for this information to reside in, so a global flag 'end_brak' is set, and is checked for in subsequent processing.

After tokenisation each word is looked up in the lexicon, as described in previous chapters. The output from the lexical look-up stage is shown below:

```
[[unknown(nancy,init_cap)],[unknown(b,init_cap)],[rel(who,animate)],
[verb(suffer,trans,fin(pers3,sg,pres),suffers,suffer)],[modal(from),prep(
from,[from])],[det(the,[def])],[unknown([guillain,barre],init_cap)],[noun(
syndrome,word,known,[abstract],[sing,n])],[comma],[verb(have,comp,fin
(_4072,_4073,past),had,have),have(have,fin(_4072,_4073,past))],[verb(say,
comp,fin(_4340,_4341, past),said,mtrans)],[prn(her,[[animate],3rd_pers,
sing,f]),det(her,[poss])],[unknown(life,word)],[verb(have,comp,fin(_4660,
_4661,past),had,have),have(have,fin(_4660,_4661,past))],[verb(become,comp,
inf,become,be)],[unknown(meaningless,word)]],
```

This shows us the various different forms each part of speech occurs in. It can be seen that 'from' can be both a modal and a preposition, 'have' can be both a main verb and a verb auxiliary, and 'her' can be both a determiner and a pronoun. These are just some of the case of lexical ambiguity that have to be resolved before parsing can begin. These ambiguities are resolved using the LAR heuristics described in D, to produce the final form of the sentence which is handed to the parser:

```
[noun(nancy,init_cap,unknown,_,_),noun(b,init_cap,unknown,_,_),rel(who,
animate),verb(suffer,trans,fin(pers3,sg,pres),suffers,suffer),prep(from,
[from]),det(the,[def]),noun([guillain,barre],init_cap,unknown,_,_),noun(
syndrome,word,known,[abstract],[sing,n]),comma,have(have,fin(_4072,_4073,
past)),verb(say,comp,fin(_4340,_4341,past),said,mtrans),det(her,[poss])],
noun(life,word,unknown,_,_),have(have,fin(_4660,_4661,past)),verb(become,
comp,inf,become,be),noun(meaningless,word,unknown,_,_)]
```

We can see that all the unknowns have been classified as nouns. These are all correct except for the adjective 'meaningless'. This is not a serious problem as a parse will still be returned, the verb 'become' can take a noun object as well as an adjective one. It might be that a morphological heuristic could handle such cases more accurately. The ambiguous words have also been correctly resolved. The 'from' in 'suffers from' is returned as a preposition as the next word is not a verb or an auxiliary, nor is the preceding word a comp or adj type verb. 'Have' is returned as a verb auxiliary in both cases because it is followed by an unambiguous verb. 'Her' is returned as a det because it is not followed by a preposition or a determiner nor is it the last word in the sentence.

## I.2 Syntactic Analysis

To illustrate the parsing process in more depth, we utilise the following sentence:

> 'An Indian court opened hearings in a criminal suit arising from the lethal gas leak which blanketed Bhopal in 1984 in the world's worst industrial disaster'.

Due to the relative complexity of this example an in-depth examination of how it is processed could become highly tedious, and would thus lose much of its expositry power. So we will consider the sentence to end at 'lethal gas leak'. Figure I.1 shows the state of each of the registers upon completion of the parse.

The sentence will be delivered from the pre-processing stage in the following form :

INPUT SENTENCE :

**An Indian court opened hearings in a criminal suit arising from the lethal gas leak which blanketed Bhopal in the world's worst industrial disaster.**

LEVEL ZERO REGISTERS

| SUB_NP | VERB | OBJECT NP | PP1 |
|---|---|---|---|
| NOUN: court | VERB : open | NOUN: hearing | NOUN: suit |
| CASE: word | TRANS: trans | CASE: word | CASE: word |
| SEM_CAT: corp | TENSE: past | SEM_CAT: event | SEM_CAT: abstract |
| ADJ : | FORM: opened | ADJ: | ADJ: criminal |
| NOUN_C: indian | SEM_CAT: open | NOUN_C: | NOUN_C: |
| DET: indef | SUB_SR: Agent | DET: | DET: indef |
|  | OBJ_SR: [object,loc,event] |  | PREP: [in, in] |

LEVEL ONE REGISTERS

| SUB_NP | VERB | PP1 |
|---|---|---|
| NOUN: suit | VERB: arise | NOUN:leak |
| CASE: word | TRANS: intrans | CASE: word |
| SEM_CAT: abstract | TENSE: prog | SEM_CAT: abstract |
| ADJ: criminal | FORM: arising | ADJ: [lethal,gas] |
| NOUN_C: | SEM_CAT: cause | NOUN_C: |
| DET: indef | SUB_SR: [abstract] | DET: def |
|  | OBJ_SR: [] | PREP: [from,from] |

LEVEL TWO REGISTERS

| SUB_NP | VERB | OBJECT NP | PP 1 |
|---|---|---|---|
| NOUN:leak | VERB: blanket | NOUN: bhopal | NOUN:1984 |
| CASE: word | TRANS: trans | CASE: init_cap | CASE: word |
| SEM_CAT: abstract | TENSE: | SEM_CAT: | SEM_CAT: digit |
| ADJ: [lethal,gas] | FORM: blanketed | ADJ: | ADJ: |
| NOUN_C: | SEM_CAT: | NOUN_C: | NOUN_C: |
| DET: def | SUB_SR: | DET: | DET: |
|  | OBJ_SR: |  |  |

| PP 2 | NOUN_PP |
|---|---|
| NOUN: disaster | NOUN: world |
| CASE: word | CASE: word |
| SEM_CAT: event | SEM_CAT: loc |
| ADJ: [worst, industrial] | ADJ: |
| NOUN_C: | NOUN_C: |
| DET: def | DET: |
| PREP: [during,at_time] | PREP: [of,of] |
|  | QUAL: disaster |

Figure I.1: State of FUNES registers during parsing

```
[det(an,[indef]),noun(indian,init_cap,unknown,[human,origin,india],[sing,b]),
noun(court,word,known,[corp,comm,l],[sing,n]),verb(open,trans,fin(_4484,
_4485,past),opened,body_move),noun(hearing,word,known,[event],[plur,n]),
prep(in,[at_time]),det(a,[indef]),noun(criminal,word,known,[human],[sing,b]),
noun(suit,word,known,[abstract],[sing,n]),verb(arise,intrans,ing,arising,
cause),prep( from,[from]),det(the,[def]),noun(lethal,word,unknown,_7597,
_7598),noun(gas,word,known,[substance],[sing,n]),noun(leak,word,known,
[abstract],[sing,n]),rel(which,inan),verb(blanketed, _7923,_7924,_7925,
_7926),noun(bhopal,init_cap,unknown,_8110,[sing,_8121]),prep(in,[at_time]),
digit(1984),prep(in,[at_time]),det(the,[def]),noun(world,word,known,[loc,
comm,loc],[sing,n]),adj(worst,unknown),adj(industrial,known),noun(disaster,
word,known,[event],[sing,n])]
```

This list is handed to the s_parse predicate, which parses a sentence as comprising (LW) (PP*) NP VP. Each of these constituents is parsed by calling the appropriate predicate and handing it the list of unparsed words. The lw predicate checks for a link word such as 'although' or 'but', at the start of the sentence. As there is not one here it exits having done nothing. Next, the pp predicate is called. This examines the list of unparsed words (which functions as the look-ahead buffer) to see if the next word is a preposition. As it is not, it simply exits, and the np predicate is called.

This predicate implements the grammar for a NP. The first np predicate attempts to process a pronoun. As there is no pronoun present this fails and we move to the second np predicate. This in turn calls all the constituent predicates which parse the NP — test_neg, test_deg, test_det, test_adj, and test_noun. Test_neg and test_deg fire and exit having done nothing. Test_det, however, processes the determiner 'an' and returns the list minus this constituent, and a variable to hold the word 'an' and another to hold its semantics 'indef'. Test_adj, receiving the list minus the word 'an' processes the origin noun 'indian' as an adjective, and returns the list minus this word, and a variable to hold it. The test_noun predicate, receiving the list minus 'an indian', processes the noun 'court', and returns the list minus this and a variable holding the noun, its syntax, case, and semantic category. This terminates the analysis of the noun and its preceding constituents. The subject noun register at Level zero is thus filled as shown in figure I.1, utilising the values returned by test_det, test_adj and test_noun.

The parser then proceeds to check for post noun constituents. Each of the relevant predicates will be called (appos, test_and, test_prep_rel, and test_rel), but all will simply exit having done nothing. This is because there are no commas to indicate a potential appositive, no 'and' to indicate a conjoined NP, no preposition to indicate a post noun PP, and no relative pronoun to indicate a relative clause.

At this point in the parse all that has been done is to process the subject NP, and fill the Level zero subject NP register as shown in figure I.1. Following the grammar rule for a declarative sentence we then proceed with a call to the VP parser. This first checks for verb auxiliaries but finds none, and so moves on to process the main verb 'opened'. This predicate returns the features TRANS, TENSE and SEM_CAT (derived in the preprocessing stage) which will later be placed in the Level zero verb register. The TRANS variable for this verb is then inspected. As it is simply 'trans' we proceed to check for a simple object NP, calling the test_obj predicate. This first calls the pp predicate to check for a post verb PP. None is found so this simply exits. Next the np predicate is called, which runs through all the predicates for the pre-noun constituents as described above. However the only NP constituent found is the noun 'hearings', so the object noun register is filled with this as shown in figure I.1. When test_prep_rel is called to check for post noun

PP's it finds a preposition. However 'in' is not a preposition that indicates attachment to the preceding noun, so the PP parser is not called from within the np predicate. Instead the NP parser exits, having simply filled the object register with the word 'hearings'.

Proceeding with the analysis of the post-verb constituents the pp predicate is called. (As 'open' is not a bitrans verb we do not check for a second NP). The preposition 'in' indicates a PP, and the NP parser is called from within the post-verb PP parser. This parses 'criminal suit' and fills the PP register as shown. When checking for an RC, the occurrence of the verb 'arising' with no progressive auxiliary indicates a subject missing relative clause. The VP parser is therefore called from within the NP parser, with its Level increased by one as we are now processing an embedded constituent . The verb 'arising' is parsed. Its TRANS setting is intrans, so the first thing checked for after it is a PP. The preposition 'from' is detected and again the NP parse is called to parse 'the lethal gas leak'. This is parsed according to the rule NP → Det Adj Noun Noun, and a Level one PP register filled accordingly.

Upon completion of this NP parse, the np predicate exits and, as there are no more PP's to parse (recalling we have curtailed our sentence at this point), the pp predicate exits. This completes the VP 'arising from the lethal gas leak'. The verb 'arising' is entered into the verb register at Level one as shown and the vp predicate exits. Processing then continues within the test_rel predicate which called it, locating the missing subject 'a criminal suit'. This is copied into the Level one subject register, and the semantic analyser called to analyse the reconstructed relative clause 'a criminal suit arising from the lethal gas leak'. Upon completion the test_rel predicate is exited, and thus the np predicate which called it.

This completes the PP 'in a criminal suit', and as there are no more PP's the pp predicate exits, completing the VP 'opened hearings in a criminal suit'. The verb 'opened' is entered into the Level zero verb register, and the semantic analyser called to analyse the main sentence 'An Indian court opened hearings in a criminal suit'.

The semantic analysis of this sentence is illustrated below.

## I.3 Semantic Analysis

The various segments of this sentence are handed to the semantic analyser in the form of NP, PP, and verb registers. The state of the registers returned by the parser are shown in figure I.1.

The first segment of this sentence to be handed to the semantic analyser is the final Relative clause 'which blanketed Bhopal ... ', together with the missing subject 'lethal gas leak'. The syntactic Level two registers which hold the relevant items are copied into the semantic registers, and then erased. The semantic registers used are 'subnoun', 'unobjnoun', 'mainverb' and three prepnoun registers. There contents are exactly the same as the Level two registers shown in figure I.1. Firstly the unobjnoun register is examined, and it is found to contain 'Bhopal' , an unknown PN. The SR's of 'blanketed' are not known, as it is an unknown verb, so no sem_cat can be derived. Nevertheless 'Bhopal' is still transferred to the objnoun register, where it will be subsequently analysed.

The check on the consistency between the sem_cats of the subject and object — 'leak' and 'Bhopal' — with the verb's SR's reveals no mismatch, as there are no SR's for the unknown verb. (As discussed in chapter 4, these could be acquired using the semantic categories of the accompanying subject and object). We then check to see if there are any PP's attached to 'leak', by examining the noun PP registers for any PP's with their QUAL slot filled by 'leak'. There are none, so we proceed to derive the case of 'leak'. The

TRANS slot of 'blanket' was set to 'trans' as it occurred with an object, so the CASE for 'leak', which has sem_cat [abstract], is set to INSTRUMENT. The adjectives/noun_comps which accompany 'leak' are then examined. The words 'lethal' and 'gas' are both held in the ADJ slot, as they are not init_cap, nor of sem_cat 'role' or 'corp'. As they are neither digits, durations or locs, they are simply returned as 'property'. So the case-frame for 'lethal gas leak' is [instrument(leak),property(lethal,gas)].

The objnoun register is processed next. This contains the single PN 'Bhopal'. The sem_cat being still unknown, and the TRANS of the verb 'trans', the case for this is THEME.

Finally the prepnoun registers are processed. The first of these contains the digit '1984' and the preposition 'in'. The 'in module' returns the case 'time(1984)', using the heuristics described in appendix H. There are no adjectives/noun_comps to analyse. There are also no PP's attached to this noun.

The next prepnoun register contains the PP 'in the worst industrial disaster'. The 'in module' returns the case 'during(disaster)'. 'Worst' and 'industrial' are simply returned as properties. Checking the noun PP register we find the PP 'of the world' attached to disaster. This is analysed by the 'of' module, which returns the case 'origin(world)'. So the final case frame for this PP is [during(disaster),property(worst,industrial),origin(world)].

The component case-frames are joined together to form the final CF for this reconstructed relative clause thus :

```
past
    blanketed
    [instrument(leak),property([lethal,gas])]
    [theme(bhopal)]
    [time(1984)]
    [during(disaster),property([worst,industrial]),origin(world)]
```

The CF's for the other two constituents are constructed in a similar manner. As stated above, the lack of lexical knowledge of the phrasal verb 'arise from' impoverishes the accuracy of the CF for the RC 'arising from the lethal gas leak'. This should ideally be

```
cause
[instrument(leak),property([lethal,gas])]
[theme(suit),property([criminal])]
```

where the CF would properly reflect the underlying semantics of the phrase. Lacking this knowledge it really only reflects the surface syntactic form:

```
pres
    arise
    [theme(suit),property([criminal])]
    [from(leak),property([lethal,gas])]
```

Similarly the CF for the main sentence does not reflect the fact that the hearings are part of the criminal suit, such a realisation is dependent on fairly deep knowledge of the nature of the law suits and hearings. As such the global 'in' is used. The word 'indian' has been correctly returned as 'origin' by the adjective analyser. The adjective 'criminal' is returned as 'property':

```
past
    open
    [agent(court),origin(india)]
    [theme(hearing)]
    [in(suit),property([criminal])]
```

After completion of each CF, the semantic registers are erased, and any global flags set during the analysis are removed. If the CF is that of an embedded sentence it is entered into a COMP register, from which it will be withdrawn when the main sentence in which it is embedded is analysed, and slotted into the CF for this sentence. The CF's for relative clauses are not actually slotted into the main sentence CF. As it is usually quite clear from the way the CF's are output that the CF for the relative clauses complement the CF for the main sentence, they are just output alongside. It would be a trivial matter to incorporate them within the main sentence CF.

# Appendix J

# Personal PN's and Role KW's

## J.1 Example Personal PN's

In this appendix, we show some example inputs (and extracts from inputs) containing a variety of personal PN's. After each input, we show the lexical and KB entries which FUNES derived for them. For the first three examples we also show the semantic form delivered for the sentences. In all of the cases we only show the lexical and KB entries acquired for the personal PN's in the input sentences.

1) A grand jury indicted Egyptian-born El-Sayyid Al-Nossair. He is accused of assassinating the anti-Arab Israeli, Rabbi Meir Kahan, in a Manhattan hotel in June.

Below we show the Case-Frame produced for these sentences:

```
past
    indict
    [agent(jury),property([grand])]
    [theme(al-nossair),origin(egypt)]
pres
    accuse
    [agent(unknown)]
    [theme(al-nossair)]
    ing
      assassinate
      [agent(al-nossair)]
      [theme(israeli),property([anti(arab)])]
      [at_loc(hotel),at_loc(manhattan)]
      [at_time(june)]
pres
    be
    [theme(israeli),property([anti(arab)])]
    [theme(kahan),role([rabbi])]
```

```
KB Entries:
kbase([al,nossair]):
[firstname:[el,sayyid],fullname:[[el,sayyid],[al,nossair]],origin:[egypt]].

kbase(kahan):
```

```
[role:[rabbi],firstname:meir,fullname:[meir,kahan],origin:[israel],
property:[anti(arab)]].

Lexical Entries:
[el,sayyid]:
def([noun([el,sayyid],[human,name],m)]).

[al,nossair]:
def([noun([al,nossair],[human,name],b)]).

meir:
def([noun(meir,[human,name],m)]).

kahan:
def([noun(kahan,[human,name],b)]).

[[el,sayyid],[al,nossair]]:
def([noun([[el,sayyid],[al,nossair]],[human,name],m)]).

[meir,kahan]:
def([noun([meir,kahan],[human,name],m)]).
```

Here we can see that FUNES is able to acquire actual country of origin info from the terms 'Egyptian-born' and 'Israeli'. The genders for both names are acquired successfully: 'Al-Nossair' from the use of the pronoun 'he' in the second sentence, and 'Kahan' from the gender of the KW 'rabbi'.

2) A Greek newspaper editor, Serafeim Fyntanides, was arrested yesterday.

Case-Frame:

```
past
    arrest
    [agent(unknown)]
    [theme([newspaper,editor]),origin(greece)]
    [time(yesterday)]
pres
    be
    [theme([newspaper,editor]),origin(greece)]
    [theme(fyntanides)]
```

```
KB Entry:
kbase(fyntanides):
[role:[newspaper,editor],firstname:serafeim,fullname:[serafeim,fyntanides],
origin:[greece]].

Lexical Entries:
serafeim:
def([noun(serafeim,[human,name],_34)]).

fyntanides:
```

```
def([noun(fyntanides,[human,name],b)]).
```

In this input (only the start of the story is shown), the gender of 'Fyntanides' can not be derived, as their is no occasion for pronoun references, nor are newspaper editors of any definite sex.

3) A former border policeman,Ismael Abdala, tried to shoot ex-President Raul Alfonsin of Argentina at an opposition rally in San Nicloas.

Case-Frame:

```
past
    try
    [agent(policeman),property([former,border])]
    inf
       shoot
       [agent(policeman),property([former,border])]
       [theme(alfonsin),role([ex(president)]),from(argentina)]
       [at_event(rally),property([opposition])]
       [at_loc([san,nicloas])]
pres
    be
    [theme(policeman),property([former,border])]
    [theme(abdala)]
```

```
KB entry:
kbase(abdala):
[role:[policeman,property([[former,border]])],firstname:ismael,fullname:
[ismael,abdala],origin:[argentina,[san,nicloas]]].

kbase(alfonsin):
[role:[ex(president)],firstname:raul,fullname:[raul,alfonsin],origin:
[argentina]].

Lexical Entries:
ismael:
def([noun(ismael,[human,name],m)]).

abdala:
def([noun(abdala,[human,name],b)]).

raul:
def([noun(raul,[human,name],_90)]).

alfonsin:
def([noun(alfonsin,[human,name],b)]).

[ismael,abdala]:
def([noun([ismael,abdala],[human,name],m)]).

[raul,alfonsin]:
```

```
def([noun([raul,alfonsin],[human,name],_91)]).
```

Here we can see that the origin for Alfonsin is correct, as it is explicitly given. That for Abdala is not explicitly given, so FUNES uses the global heuristic, and correctly guesses he is from Argentina, and also from San Nicloas.

4) The President of the Ivory Coast, Felix Houphouet-Boigny, said that the Liberian rebel leader Charles Taylor and the head of the interim government in Monrovia, Amos Sawyer,were reconciled at a recent summit.

```
KB entries:
kbase([houphouet,boigny]):
[firstname:felix,fullname:[felix,[houphouet,boigny]],role:[president],origin:
[ivory,coast]].

kbase(taylor):
[role:[leader],firstname:charles,fullname:[charles,taylor],property:[rebel],
origin:[liberia]].

kbase(sawyer):
[works_for:[government,property([[interim]])],firstname:amos,fullname:
[amos,sawyer],role:[head],origin:[liberia]].

Lexical Entries:
felix:
def([noun(felix,[human,name],_34)]).

[houphouet,boigny]:
def([noun([houphouet,boigny],[human,name],b)]).

charles:
def([noun(charles,[human,name],_98)]).

taylor:
def([noun(taylor,[human,name],b)]).

amos:
def([noun(amos,[human,name],_160)]).

sawyer:
def([noun(sawyer,[human,name],b)]).

[felix,[houphouet,boigny]]:
def([noun([felix,[houphouet,boigny]],[human,name],_320)]).

[charles,taylor]:
def([noun([charles,taylor],[human,name],_360)]).

[amos,sawyer]:
def([noun([amos,sawyer],[human,name],_400)]).
```

This is a very complex sentence, despite which FUNES has derived almost complete information on all three personal PN's. NP's that are 17 words long seem rarely covered in most grammatical sources, and yet in real text they are not uncommon.

5) Abdel-Rahim Ahmed, leader of the Iraqi-backed Arab Liberation Front and a member of the PLO's Executive Committee, died of cancer yesterday.

```
KB entry:
kbase(ahmed):
[works_for:[executive,committee],works_for:[arab,liberation,front],firstname:
[abdel,rahim],fullname:[[abdel,rahim],ahmed],role:[leader],property:[dead]].

Lexical Entries:
[abdel,rahim]:
def([noun([abdel,rahim],[human,name],_36)]).

ahmed:
def([noun(ahmed,[human,name],b)]).

[[abdel,rahim],ahmed]:
def([noun([[abdel,rahim],ahmed],[human,name],_160)]).
```

Again, almost complete information has been derived from a very complex example. A reasonable default for the gender of any Muslim type name in news text would be male (as indeed it would for the role word 'leader', however much we may wish it otherwise). The only slight problem here is that Ahmed has not been directly tied to the PLO. However, as the subsequent part of the story associates the Arab Liberation Front and the PLO, the KB entry for Arab Liberation Front includes that it is a sub-part of the PLO. An inference component that could create a work_for link from this information would be a viable consideration.

6) American Express Co. named Joan Edelman Spero to the post of senior vice president and treasurer. Ms. Spero, 45 years old, had been senior vice president for international corporate affairs and retains those duties.

```
KB entry:
kbase(spero):
[role:[[vice,president],property([senior])],firstname:joan,fullname:
[joan,edelman,spero],origin:[america],age:[45]]

Lexical Entries:
joan:
def([noun(joan,[human,name],f)]).

spero:
def([noun(spero,[human,name],b)])].

[joan,edelman,spero]:
def([noun([joan,edelman,spero],[human,name],f)]).
```

This is an example from the WSJ corpus, and the acquired information is far from perfect.

The construction 'name X to the post of Y' is not handled in FUNES, so all of the first sentence's definitional info is lost. Work to remedy this would be perfectly feasible.

7) Ohio Mattress Co. said Malcolm Candlish, president and chief operating officer, was given the additional position of chief executive officer nine months earlier than had been expected. As chief executive, Mr. Candlish, 54 years old, succeeds Ernest M. Wuliger, 68.

```
KB entries:
kbase(wuliger):
[role:[chief,executive],firstname:ernest,fullname:[ernest,m,wuliger],
age:[68]]

kbase(candlish):
[role:[president],role:[chief,operating,officer],firstname:malcolm:,
fullname:[malcolm,candlish],age:[54]]

Lexical Entries:
[ernest,m,wuliger]:
def([noun([ernest,m,wuliger],[human,name],_2002)]).

[malcolm,candlish]:
def([noun([malcolm,candlish],[human,name],m)]).

ernest:
def([noun(ernest,[human,name],_2003)]).

wuliger:
def([noun(wuliger,[human,name],b)]).

malcolm:
def([noun(malcolm,[human,name],m)]).

candlish:
def([noun(candlish,[human,name],b)]).
```

This is actually a later result, from a WSJ corpus example which, in the evaluation, fared very badly. The compound noun 'chief operating officer' was handed to the parser as 'noun verb noun', and the name 'candlish' was returned as an adjective, due to the 'ish' ending. Thus absolutely no info was gained on Mr Candlish. In the revisions after the test run, 'chief operating officer' was added as a compound noun, and the 'ish → adj' heuristic only run after the personal PN heuristics. Performance was thus improved, but is still far from perfect. Firstly there are no works_for slots, this is because we are not explicitly told that the men work for Ohio Matress Co. This shows the need for some kind of global heuristic for works_for info, similar to the global origin heuristic. The final NP 'nine months earlier than had been expected' is not parsed correctly, as it runs into the role NP that precedes it. This means that this role info is lost. Finally, the topicalisation of 'As chief executive' prevents the working of the 'succeeds' process, so this info is not correctly acquired.

## J.2 Role KW's

Below we list all the role words currently in the FUNES lexicon. In addition to those shown all unknown words ending in 'ist', or those ending in 'er/or' and which have a known verb root, will be classified as role words.

Those entries which are not preceded by def are the latter halves of ambiguous entries, e. g. chair can be both an item of furniture and a role word.

The second argument in the Semantics slot indicates whether this role word is preceded by a place PN complement or a corp PN complement. 'Loc' indicates the former, 'corp' the latter. 'B' indicates it can be both. This argument performs the same role for a following PP, i. e. indicating whether it is a place or a corp.

```
def([noun(accountant,[role,corp],b)]).
def([noun(activist,[role,b],b)]).
def([noun(advisor,[role,corp],b)]).
def([noun(agent,[role,corp],b)]).
def([noun(aide,[role,b],b)]).
def([noun(ally,[role,loc],b)]).
def([noun(ambassador,[role,loc],m)]).
def([noun(analyst,[role,b],b)]).
def([noun(arbitrator,[role,b],b)]).
def([noun(archbishop,[role,b],m)]).
def([noun(architect,[role,b],b)]).
def([noun(assassin,[role,b],m)]).
def([noun(attache,[role,b],b)]).
def([noun(attorney,[role,corp],b)]).
def([noun(author,[role,corp],b)]).
def([noun(ayatollah,[role,b],m)]).
def([noun(baron,[role,loc],m)]).
def([noun(bishop,[role,b],m)]).
def([noun(boss,[role,corp],b)]).
def([noun(brigadier,[role,b],m)]).
def([noun(broker,[role,corp],b)]).
def([noun(campaigner,[role,b],b)]).
def([noun(candidate,[role,b],b)]).
def([noun(captain,[role,loc],m)]).
def([noun(captain,[role,loc],m)]).  % def for 'capt'
noun(chair,[role,corp],b)]).
def([noun(chairman,[role,corp],b)]).
def([noun(chancellor,[role,b],b)]).
def([noun(chief,[role,corp],b)],
def([noun([chief,minister],[role,corp],b)]).
def([noun([chief,executive],[role,corp],b)]).
def([noun([chief,executive,officer],[role,corp],b)]).
def([noun([chief,operating,officer],[role,corp],b)]).
def([noun(civilian,[role,loc],b)]).
def([noun(cleric,[role,loc],m)]).
def([noun(clerk,[role,b],b)]).
def([noun(colonel,[role,loc],m)]).
def([noun(colonel,[role,loc],m)]).  % def for 'col'
def([noun(comedian,[role,loc],b)]).
def([noun(commander,[role,corp],m)]).
def([noun(commissioner,[role,corp],b)]).
def([noun(communist,[role,b],b)]).
def([noun(congressman,[role,b],m)]).
def([noun(conservative,[role,loc],b)]).
def([noun(consultant,[role,b],b)]).
def([noun(controller,[role,b],b)]).
noun(convict,[role,b],b)]).
def([noun([co,ordinator],[role,b],b)]).
def([noun(coroner,[role,b],b)]).
def([noun(councillor,[role,corp],b)]).
def([noun(counsel,[role,b],b)]).
noun(count,[role,loc],m)]).
def([noun(countess,[role,loc],f)]).
def([noun(counterpart,[role,b],b)]).
def([noun(creditor,[role,b],b)]).
def([noun(critic,[role,b],b)]).
def([noun(customer,[role,b],b)]).
def([noun(deacon,[role,b],b)]).
def([noun(dealer,[role,corp],b),
def([noun(defendant,[role,b],b)]).
def([noun(democrat,[role,b],b)]).
def([noun(deputy,[role,b],b)]).
def([noun(detective,[role,loc],b)]).
def([noun(diplomat,[role,loc],b)]).
def([noun(director,[role,corp],b)],
def([noun([director,general],[role,corp],m)]).
def([noun(dissident,[role,loc],b)]).
def([noun(doctor,[role,corp],b)]).
def([noun(doctor,[role,corp],b)]).  % def for 'dr'
def([noun(duchess,[role,loc],f)]).
def([noun(duke,[role,loc],m)]).
def([noun(earl,[role,loc],m)]).
def([noun(economist,[role,b],b)]).
def([noun(editor,[role,corp],b)]).
def([noun(employee,[role,corp],b)]).
def([noun(engineer,[role,b],b)]).
def([noun(entrepeneur,[role,loc],b)]).
def([noun(envoy,[role,loc],b)]).
def([noun(executive,[role,corp],b)],
def([noun([executive,vice,president],[role,corp],b)]).
def([noun(expert,[role,corp],b)]).
def([noun([foreign,minister],[role,loc],b)]).
def([noun([foreign,secretary],[role,loc],b)]).
def([noun(founder,[role,corp],b)]).
def([noun(general,[role,corp],m)]).
def([noun(general,[role,corp],m)]).  % def for 'gen'
def([noun(governor,[role,loc],b)]).
```

```
 noun(head,[role,corp],n)]).
def([noun(heir,[role,loc],b)]).
def([noun(hijacker,[role,b],b)]).
def([noun([home,secretary],[role,loc],b)]).
def([noun([interior,minister],[role,loc],b)]).
def([noun(investor,[role,b],b)]).
def([noun(journalist,[role,b],b)]).
 noun(judge,[role,corp],b)]).
def([noun(justice,[role,b],m),
def([noun(king,[role,loc],m)]).
def([noun(lawyer,[role,corp],b)]).
def([noun(leader,[role,b],b)]).
def([noun(lieutenant,[role,corp],b)],
def([noun([lieutenant,general],[role,corp],m)]).
def([noun([lieutenant,colonel],[role,corp],m)]).
def([noun(lieutenant,[role,corp],b)]).   % def for ‘lt’
def([noun([lieutenant,colonel],[role,corp],m)]).  % def for ‘lt col’
def([noun(magistrate,[role,loc],b)]).
def([noun(major,[role,corp],m)],
def([noun([major,general],[role,corp],m)]).
def([noun(manager,[role,corp],b)]).
def([noun([managing,director],[role,corp],b)]).
def([noun(marine,[role,b],m)]).
def([noun(master,[role,b],m)]).
def([noun(mayor,[role,loc],b)]).
def([noun(member,[role,corp],b)]).
def([noun(militant,[role,b],b)]).
def([noun(miner,[role,b],m)]).
def([noun(minister,[role,loc],b)]).
def([noun([member,of,parliament],[role,loc],b)]).
def([noun([newspaper,editor],[role,corp],b)]).
def([noun(nominee,[role,b],b)]).
def([noun(offender,[role,b],b)]).
def([noun(officer,[role,corp],b)]).
def([noun(official,[role,corp],b)]).
def([noun(opponent,[role,b],b)]).
def([noun(partner,[role,b],b)]).
def([noun(pastor,[role,corp],m)]).
def([noun(patient,[role,b],b)]).
def([noun(pilot,[role,b],b)]).
def([noun(policeman,[role,corp],m)]).
def([noun([police,officer],[role,corp],b)]).
def([noun(politician,[role,b],b)]).
def([noun(president,[role,loc],b)]).
def([noun(priest,[role,b],m)]).
def([noun(primate,[role,b],m)]).
def([noun([prime,minister],[role,loc],b)]).
def([noun(prince,[role,loc],m)]).
def([noun(princess,[role,loc],f)]).
def([noun(principal,[role,corp],b)]).
def([noun(professor,[role,b],b)]).
def([noun(professor,[role,b],b)]).  % def for ‘prof’
def([noun(prosecutor,[role,b],b)]).
def([noun(publisher,[role,i_source],b)]).
def([noun(qc,[role,b],b)]).
def([noun(queen,[role,loc],f)]).
def([noun(rabbi,[role,b],m)]).
def([noun(rebel,[role,b],b)]).
def([noun(reporter,[role,b],b)]).
def([noun(representative,[role,b],b)]).
def([noun(republican,[role,b],b)],
noun(rival,[role,b],b)]).
def([noun(scientist,[role,b],b)]).
def([noun([second,in,command],[role,b],b)]).
def([noun(secretary,[role,corp],b)],
def([noun([secretary,general],[role,corp],m)]).
def([noun(senator,[role,loc],b)]).
def([noun(senior,[role,b],b)]).
def([noun(sergeant,[role,loc],b)]).
def([noun(shareholder,[role,corp],b)]).
def([noun(sheik,[role,loc],m)]).
def([noun(singer,[role,b],b)]).
def([noun(socialist,[role,b],b)]).
def([noun(solicitor,[role,b],b)]).
def([noun(speaker,[role,corp],b)]).
def([noun([special,envoy],[role,loc],b)]).
def([noun(spokesman,[role,corp],m)]).
def([noun(spokesperson,[role,corp],b)]).
def([noun(spokeswoman,[role,corp],f)]).
noun(star,[role,n],b)]).
def([noun(stockbroker,[role,corp],b)]).
def([noun(student,[role,b],b)]).
def([noun(successor,[role,b],b)]).
def([noun(supporter,[role,b],b)]).
def([noun(surgeon,[role,b],b)]).
def([noun(surveyor,[role,b],b)]).
 noun(suspect,[role,corp],b)]).
def([noun(technician,[role,b],b)]).
def([noun(teenager,[role,loc],b)]).
 def([noun(terrorist,[role,corp],b)]).
def([noun(tory,[role,loc],b)],
def([noun(treasurer,[role,corp],n)]).
def([noun([under,secretary],[role,corp],b)]).
def([noun(undersecretary,[role,corp],b)]).
def([noun(underwriter,[role,corp],b)]).
def([noun(unionist,[role,b],n)]).
def([noun([vice,president],[role,b],b)]).
def([noun([vice,secretary],[role,corp],b)]).
def([noun(victim,[role,b],b)]).
def([noun(warlord,[role,loc],m)]).
noun(witness,[role,b],b)]).
def([noun(worker,[role,b],b)]).
```

# Appendix K

# Corp PN's and KW's

## K.1 Example Corp PN's

In this section we present several examples of corp PN's in sentences taken from the FUNES test and development corpora together with the lexical and KB entries derived for them by FUNES. (We only show the corp entries, not the entries produced for all the PN's).

1) Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.

```
KB Entry:
kbase([elsevier,n,v]):
[isa:[group,property([[publishing]])],origin:[holland],staff:
[chairman(vinken)]].

Lexical Entry:
[elsevier,n,v]:
def([noun([elsevier,n,v],[corp,name],n)]).
```

A straightforward example. 'N.V' is a common European corp suffix. Here we can see that the corp 'Elsevier' is defined both by a suffix and an appositive.

2) Pacific First Financial Corp. said shareholders approved its acquisition by Royal Trustco Ltd. of Toronto for $27 a share, or $212 million. The thrift holding company said it expects to obtain regulatory approval and complete the transaction by year-end.

```
Lexical Entries:
[pacific,first,financial,corp]:
def([noun([pacific,first,financial,corp],[corp,name],n)]).

[royal,trustco,ltd]:
def([noun([royal,trustco,ltd],[corp,name],n)]).
```

Again, a straightforward example, showing the heavy use made in US news of corp suffixes. Here we have an example of a distal definition which is not yet handled by FUNES for corps. 'The thrift holding company' is a description of 'Pacific First Financial Corp'. The information conveyed by 'approved its acquisition' can not be derived by FUNES. There are no KB entries for these corps as no differentia info has been acquired. The only real info available is the origin of 'Royal Trustco Ltd', but as Toronto is unknown this can not be acquired. Toronto could be a human as well as a place, so can not be assigned as origin.

3) J. P. Bolduc, vice chairman of W. R. Grace & Co., which holds a 83.4% interest in this energy-services company, was elected a director. He succeeds Terrence D. Daniels, formerly a W.R. Grace vice chairman, who resigned.

```
KB Entries:
kbase([w,r,grace,&,co]):
[staff:chairman(bolduc)].

kbase([w,r,grace]):
[staff:chairman(daniels)].

Lexical Entries:
[w,r,grace,&,co]:
def([noun([w,r,grace,&,co],[corp,name],n)]).

[w,r,grace]:
def([noun([w,r,grace],[corp,name],n)]).
```

This is a more complex example. The ampersand makes it clear that we have a company and not an person. The second occurrence of the corp is a variant form, connected to the first occurrence with a heuristic described in chapter 6. In FUNES this connection has not been made as noun complement corp PN's are not checked against the Genus DB, but just entered straight into it. This is an oversight which could be easily remedied.

4) Texas Instruments Japan Ltd., a unit of Texas Instruments Inc., said it opened a plant in South Korea to manufacture control devices.

```
KB Entries:
kbase([texas,instrument,japan,ltd]:
[isa:[unit],superpart:[texas,instrument,incorporated],origin:[japan],
field:[instrument]].

kbase([texas,instrument,incorporated]):
[field:[instrument]].

Lexical Entries:
[texas,instrument,japan,ltd]:
def([noun([texas,instrument,japan,ltd],[corp,name],n)]).

[texas,instrument,incorporated],[corp,name],n):
def([noun([texas,instrument,incorporated],[corp,name],n)]).
```

5) McDermott International Inc said its Babcock & Wilcox unit completed the sale of its Bailey Controls Operations to Finmeccanica S p A for $295 million. Finmeccanica is an Italian state-owned holding company with interests in the mechanical engineering industry.

```
KB Entries:
kbase(finmeccanica):
[origin:italy,isa:[company,property([[(state,owned)],[holding]])],namehead:
[finmeccanica,spa]].

kbase([babcock,&,Wilcox]):
[isa:unit].

Lexical Entries:
[babcock,&,Wilcox]:
def([noun([babcock,&,Wilcox],[corp,name],n)].

finmeccanica:
def([noun(finmeccanica,[corp,name],n)].

[finmeccanica,spa]:
def([noun([finmeccanica,spa],[corp,name],n)].

[mcdermott,international,inc]:
def([noun([mcdermott,international,inc],[corp,name],n)]).

[bailey,control,operation]:
def([noun([bailey,control,operation],[corp,name],n)].
```

All four of these corps have been successfully acquired. The only failure is the inability to use knowledge of the possessive pronoun 'its', to tie Babcock and Wilcox and Bailey Controls to McDermott International. The phrase 'with interests in' is also not catered for.

6) American Telephone & Telegraph Co said it signed five-year telecommunications contracts totaling $100 million with Kemper Financial Co and PaineWebber Inc. The contracts follow filings AT&T made last Friday with the Federal Communications Commission under a special arrangement called Tariff 12.

```
KB Entries:
kbase([american,telephone,&,telegraph,co]):
[origin:[america]].

kbase([kemper,financial,co]):
[origin:[america]]

kbase([painewebber,incorporated]):
[origin:[america]].

[federal,communication,commission]:
[origin:[america]].

Lexical Entries:
[american,telephone,&,telegraph,co]:
```

```
def(noun([american,telephone,&,telegraph,co],[corp,name],n)]).

at&t:
def([noun(at&t,[corp,name],n)]).

[kemper,financial,co],[corp,name]:
def([noun([kemper,financial,co],[corp,name],n)]).

[painewebber,incorporated],[corp,name],n):
def([noun([painewebber,incorporated],[corp,name],n)]).

[federal,communication,commission],[corp,name],n):
def([noun([federal,communication,commission],[corp,name],n)]).
```

Again, all the corps are successfully acquired. There is no differentia info to be acquired, except origin, which is done via the global origin heuristic. The only problem is the failure to match AT&T to its full form. This is because the acronym form constructed for 'American Telephone ...' will include the Co, as it was capitalised, and thus 'AT&TC' will not match 'AT&T'. This could be solved by constructing two acronym forms, one with and the other without the corp KW.

7) Sir Allen Sheppard, chairman of Grand Metropolitan, is joining the board of Michael Guthrie's new leisure company, Brightreasons II.

```
KB Entries:
kbase([brightreasons,ii]):
[isa:[company,property([[new,leisure]]),run_by(guthrie)]].

kbase([grand,metropolitan]):
[origin:[britain],staff:[chairman(sheppard)]].

Lexical Entries:
[brightreasons,ii]:
def([noun([brightreasons,ii],[corp,name],n)]).

[grand,metropolitan]:
def([noun([grand,metropolitan],[corp,name],n)]).
```

The phrase 'joining the board of X' is not covered in the semantic stage, so the fact that Sheppard now works for Brightreasons II is not derived. As we are now back in UK news, the absence of corp suffixes becomes immediately apparent.

8) Price Waterhouse, administrator of MCC, has laid off 36 staff at Macdonald, the paperback publisher it is trying to sell.

```
KB Entries:
kbase([price,waterhouse]):
[superpart:[mcc],isa:[administrator]].

kbase(macdonald):
```

```
[role:[publisher,property([[paperback]])]].

Lexical Entries:
mcc:
def([noun(mcc,[corp,name],n)]).

[price,waterhouse]:
def([noun([price,waterhouse],[corp,name],n)]).

macdonald:
def([noun(macdonald,[human,name],n)]).
```

This input illustrates the problems caused by ambiguous KW's. 'Publisher' is held as a role KW, and so 'Macdonald' is classified as a personal PN. The 'superpart' slot for 'Price Waterhouse' is generated due to the PP module finding the pattern 'Corp of Corp' which is usually indicative of the 'superpart' relationship.

9) The European Commission has approved the purchase of Cinzano, the Italian drinks firm, by Grand Metropolitan's subsidiary International Distillers and Vintners.

```
KB Entries:
kbase(cinzano):
[isa:[firm,property([[drink]])],origin:[italy]].

kbase([international,distillers,and,vintners]):
[isa:[subsidiary],superpart:[grand,metropolitan]].

Lexical Entries:
cinzano:
def([noun(cinzano,[corp,name],n)]).

[international,distillers,and,vintners]:
def([noun([international,distillers,and,vintners],[corp,name],n)]).
```

When this story was processed, both 'European Commission' and 'Grand Metropolitan' were both known, so are not acquired.

## K.2 Corp KW's

In this section are listed all the corp KW's in the FUNES lexicon. The 'comm' argument in the semantic information indicates they are common nouns, rather than PN's. This is useful with corps which occurs in complex NG's, as if the corp is a PN it indicates the following word is a product (i. e. object PN), while if it is a KW it indicates the following word is a PN (a corp PN). The third argument, which can be 'n', 'l', or 'b', is used in deciding the category of a preceding capitalised unknown. 'N' indicates it will be a Proper Noun, (e. g. Burton Group, Miller Bros). 'L' indicates it will be an origin/place PN, (e. g. the Serbian army, Durham County Council). 'B' is for words which have no definite preference. It may seem that many of the words classified as 'n' can also take origin PN's.

This is true, but has not found to occur often, and as the majority of origins are known, would not cause a problem anyway.

```
def([noun(accountants,[corp,comm,n],n)]).
def([noun(acquisition,[corp,comm,n],n)]).
def([noun([action,group],[corp,comm,b],n)]).
def([noun(administration,[corp,comm,b],n)]).
def([noun(administrator,[corp,comm,n],n)]).
def([noun(ag,[corp,comm,n],n)]).
def([noun(agency,[corp,comm,n],n)]).
def([noun([air,force],[corp,comm,l],n)]).
def([noun(airline,[corp,comm,b],n)]).
def([noun(airways,[corp,comm,b],n)]).
def([noun(alliance,[corp,comm,b],n)]).
def([noun(amalgamated,[corp,comm,n],n)]).
def([noun([appeal,court],[corp,comm,l],n)]).
def([noun(appliances,[corp,comm,n],n)]).
def([noun([armed,forces],[corp,comm,l],n)]).
def([noun(army,[corp,comm,l],n)]).
def([noun(assembly,[corp,comm,b],n)]).
def([noun(associated,[corp,comm,n],n)]).
def([noun(associated,[corp,comm,n],n)]).  % def for ‘assoc’
def([noun(association,[corp,comm,n],n)]).
def([noun(auditor,[corp,comm,n],n)]).
def([noun(authority,[corp,comm,b],n),
 noun(bank,[corp,comm,n],n)]).
def([noun(battalion,[corp,comm,n],n)]).
def([noun(bloc,[corp,comm,n],n)]).
def([noun(board,[corp,comm,b],n)]).
def([noun(body,[corp,comm,b],n)]).
def([noun(branch,[corp,comm,b],n)]).
def([noun(brigade,[corp,comm,b],n)]).
def([noun(bros,[corp,comm,n],n)]).
def([noun(brotherhood,[corp,comm,n],n)]).
def([noun([building,society],[corp,comm,n],n)]).
def([noun(business,[corp,comm,n],n)]).
def([noun(campaign,[corp,comm,n],n)]).
def([noun(caste,[corp,comm,n],n)]).
def([noun(centre,[corp,comm,b],n),
def([noun(center,[corp,comm,b],n),
noun(chain,[corp,comm,b],n)]).
def([noun(chamber,[corp,comm,b],n),
def([noun(charity,[corp,comm,n],n)]).
def([noun(church,[corp,comm,b],n),
def([noun([city,council],[corp,comm,l],n)]).
def([noun(club,[corp,comm,n],n)]).
def([noun(co,[corp,comm,n],n)]).
def([noun(coalition,[corp,comm,n],n)]).
def([noun(collective,[corp,comm,b],n)]).
noun(combine,[corp,comm,b],n)]).
```

```
 noun(command,[corp,comm,b],n)] ).
def([noun(commission,[corp,comm,b],n)]).
def([noun(committee,[corp,comm,n],n)]).
def([noun(commonwealth,[corp,comm,b],n)]).
def([noun(community,[corp,comm,b],n)]).
def([noun(company,[corp,comm,n],n)]).
def([noun(concern,[corp,comm,b],n),
def([noun(confederation,[corp,comm,b],n)]).
def([noun(conglomerate,[corp,comm,n],n)]).
def([noun(congress,[corp,comm,b],n)]).
def([noun(consortium,[corp,comm,b],n)]).
def([noun([co,operative],[corp,comm,b],n)]).
def([noun(corp,[corp,comm,n],n)]).
def([noun(corporation,[corp,comm,n],n)]).
def([noun(council,[corp,comm,l],n)]).
def([noun(court,[corp,comm,l],n)],
def([noun([court,of,appeal],[corp,comm,l],n)]).
def([noun([crown,court],[corp,comm,l],n)]).
noun(dealer,[corp,comm,b],n)]).
def([noun(delegation,[corp,comm,loc],n)]).
def([noun(democrats,[corp,comm,b],n)]).
def([noun(department,[corp,comm,b],n)],
def([noun([department,store],[corp,comm,n],n)]).
def([noun(detachment,[corp,comm,n],n)]).
def([noun(distributor,[corp,comm,n],n)]).
def([noun(division,[corp,comm,b],n)]).
def([noun(electric,[corp,comm,n],n)]).
def([noun(embassy,[corp,comm,l],n)]).
def([noun([estate,agency],[corp,comm,n],n)]).
def([noun([estate,agents],[corp,comm,n],n)]).
def([noun(estates,[corp,comm,n],n)]).
def([noun([executive,committee],[corp,comm,b],n)]).
def([noun([executive,council],[corp,comm,b],n)]).
def([noun(faction,[corp,comm,n],n)]).
def([noun(federation,[corp,comm,b],n)]).
def([noun(firm,[corp,comm,l],n)]).
 noun(force,[corp,comm,b],n)]).
def([noun(forum,[corp,comm,b],n)]).
def([noun(foundation,[corp,comm,n],n)]).
def([noun(front,[corp,comm,n],n),
def([noun(fund,[corp,comm,n],n),
def([noun(giant,[corp,comm,b],n)]).
def([noun(government,[corp,comm,l],n)]).
def([noun(group,[corp,comm,n],n)]).
def([noun(holdings,[corp,comm,n],n)]).
def([noun(inc,[corp,comm,n],n)]).
def([noun(incorporated,[corp,comm,n],n)]).
def([noun(industries,[corp,comm,n],n)]).
def([noun(infantry,[corp,comm,b],n)]).
def([noun([inquest,jury],[corp,comm,l],n)]).
def([noun(institute,[corp,comm,b],n)]).
def([noun(institution,[corp,comm,b],n)]).
def([noun([interior,ministery],[corp,comm,l],n)]).
def([noun(international,[corp,comm,n],n),
def([noun(junta,[corp,comm,b],n)]).
def([noun(jury,[corp,comm,b],n)]).
def([noun(league,[corp,comm,n],n)]).
def([noun(legislature,[corp,comm,b],n)]).
def([noun(limited,[corp,comm,n],n)]).
def([noun(ltd,[corp,comm,n],n)]).
def([noun(maker,[corp,comm,b],n)]).
def([noun(management,[corp,comm,b],n)]).
def([noun(manufacturer,[corp,comm,b],n)]).
def([noun(marines,[corp,comm,l],n)]).
noun(market,[corp,comm,n],n)]).
def([noun(merchant,[corp,comm,b],n)],
def([noun([merchant,bank],[corp,comm,b],n)]).
def([noun(military,[corp,comm,l],n)]).
def([noun(militia,[corp,comm,b],n)]).
def([noun(mine,[corp,comm,n],n)]).
def([noun(ministry,[corp,comm,b],n)]).
def([noun(movement,[corp,comm,b],n)]).
def([noun(nv,[corp,comm,n],n)]). % two entries as can occur as a single word
def([noun(nv,[corp,comm,n],n)]). % or two words, e.\ g.\ N. V. , or NV
def([noun(nation,[corp,comm,l],n)]).
def([noun(navy,[corp,comm,l],n)]).
def([noun(network,[corp,comm,n],n)]).
def([noun([news,agency],[corp,comm,n],n)]).
def([noun(office,[corp,comm,b],n),
def([noun(operation,[corp,comm,n],n)]).
def([noun(opposition,[corp,comm,b],n)]).
def([noun(organisation,[corp,comm,n],n)]).
def([noun(panel,[corp,comm,b],n)]).
def([noun(parliament,[corp,comm,l],n)]).
def([noun(partnership,[corp,comm,b],n)]).
def([noun(party,[corp,comm,n],n)]).
 noun(patrol,[corp,comm,b],n)]).
def([noun(plc,[corp,comm,n],n)]).
def([noun(police,[corp,comm,l],n)],
 noun(press,[corp,comm,l],n)],
def([noun(property,[corp,comm,n],n)]).
def([noun(public,[corp,comm,b],n)],
def([noun(refiner,[corp,comm,b],n)]).
def([noun(regiment,[corp,comm,b],n)]).
def([noun(resource,[corp,comm,n],n),
def([noun(spa,[corp,comm,n],n)]). % two entries as for NV. Can be 'S. P. A.'
def([noun(spa,[corp,comm,n],n)]). % or simnply 'spa/Spa'.
def([noun([security,forces],[corp,comm,b],n)]).
def([noun(securities,[corp,comm,n],n)]).
def([noun(senate,[corp,comm,b],n)]).
def([noun(shop,[corp,comm,n],n)]).
def([noun(society,[corp,comm,n],n)]).
def([noun(squad,[corp,comm,n],n)]).
def([noun(squadron,[corp,comm,b],n)]).
def([noun(staff,[corp,comm,b],n),
 noun(state,[corp,comm,b],n)],
 def([noun(steelworks,[corp,comm,b],n)]).
def([noun(subsidiary,[corp,comm,b],n)]).
def([noun(supplier,[corp,comm,b],n)]).
def([noun([supreme,court],[corp,comm,l],n)]).
def([noun(syndicate,[corp,comm,b],n)]).
def([noun(system,[corp,comm,n],n)]).
def([noun(team,[corp,comm,b],n)]).
def([noun(thinktank,[corp,comm,b],n)]).
def([noun([town,council],[corp,comm,l],n)]).
def([noun([trade,union],[corp,comm,l],n)]).
def([noun(treasury,[corp,comm,l],n)]).
 def([noun(tribunal,[corp,comm,n],n)]).
  noun(trust,[corp,comm,n],n)]).
def([noun([tv,station],[corp,comm,b],n)]).
def([noun(union,[corp,comm,n],n)]).
def([noun(unit,[corp,comm,b],n)]).
def([noun(united,[corp,comm,n],n)]).
def([noun(utilities,[corp,comm,n],n)]).
def([noun(watchdog,[corp,comm,b],n)]).
def([noun(wing,[corp,comm,l],n)]).
```

# Appendix L

# Place PN's and KW's

## L.1 Example Place PN's

In this appendix we present several examples of place PN's in sentences taken from the FUNES test and development corpora, together with the lexical and KB entries derived for them by FUNES. As usual only the place PN entries are shown.

1) Togo's reformist prime minister, Joseph Kokou Koffigoh, captured in a coup by rebellious troops, was still at the helm yesterday.

```
Lexical Entry:
togo:
def([noun(togo,[loc,name],n)]).
```

This shows the use of a role KW in the derivation of a place genus for a following 'of PP'. The pattern 'Prime Minister of X' clearly indicates that X is a place PN.

2) More than one hundred thousand Bulgarians packed central Sofia yesterday.

```
Lexical Entries:
sofia:
def([noun(sofia,[loc,name],n)]).

bulgarian:
def([noun(bulgarian,[human,origin,bulgaria],b)]).
```

This shows the use of a preceding KW 'central' in categorising a following PN as a place PN. Also the use of morphology, where unknown PN's ending in -'ians' are categories as Origin PN's.

3) More black factional violence erupted to the south of Johannesburg when a mob set fire to a beer hall and bus in Sebokeng township.

```
KB Entry:
kbase(sebokeng):
[isa:[township]].

Lexical Entries:
sebokeng:
```

```
def([noun(sebokeng,[loc,name],n)]).

johannesburg:
def([noun(johannesburg,[loc,name],n)]).
```

This shows the use of 'south' as a KW, in the pattern 'to the south of X'. 'Sebokeng' is also defined as a place PN, due to the following KW 'township'.

4) The crash of the transport plane ended nearly 11 years of military rule in Pakistan.

```
Lexical Entry:
pakistan:
def([noun(pakistan,[loc,name],n)]).
```

A simple example showing the use of the preposition 'in' in classifying a following unknown as a place PN (at time of processing all the world's countries were not in the FUNES lexicon so Pakistan was unknown).

5) Thirty seven people were killed when a plane crashed in a heavy rainstorm on the Thai resort island of Koh Samui.

```
KB Entry:
kbase([koh,samui]):
[isa:[island,property([[resort]])],superpart:[thailand]].

Lexical Entry:
[koh,samui]:
def([noun([koh,samui],[loc,name],n)]).
```

This shows the use of the pattern 'place_kw of X' to classify X as a place PN. When this story was originally processed 'Thai' was unknown to FUNES. It was still returned as the 'superpart' of 'Koh Samui' due to its occurrence before a place KW which is followed by 'of'. However, as it is not possible to derive the correct place form of 'thai' is was not entered as 'thailand'. The KB entry above was derived from subsequent processing, when 'Thai' was known.

6) Attackers seeking weapons raided a military base in Russia's breakaway Chechen republic and fought an all-night gunbattle with national guardsmen, Interfax news agency said, killing or wounding 20 people.

```
KB Entry:
kbase(chechen):
[isa:[republic],superpart:[russia]].

Lexical Entry:
chechen:
def([noun(chechen,[loc,name],n)]).
```

Another example showing the use of a following KW. The presence of the following 'of Russia' also provides superpart info.

7) Fierce fighting is being waged between Rangoon troops and Burmese rebels for a strategic mountain-top from which the army could attack Manerplaw, headquarters of Burma's

pro-democracy movement. More than 7,000 soldiers are attacking Htee Pawi Kyo hill, about 15 miles from Manerplaw, defended by about 3,000 fighters from one of the largest rebel groups, the Karen National Union (KNU).

```
KB Entries:
kbase(manerplaw):
[isa:[headquarters],origin:[burma]].

kbase([htee,pawi,kyo]):
[isa:[hill],location:[distance(measure(mile),number([15])),
from_loc(manerplaw)],superpart:[burma]].

Lexical Entries:
manerplaw:
def([noun(manerplaw,[loc,name],n)]).

[htee,pawi,kyo]:
def([noun([htee,pawi,kyo],[loc,name],n)]).
```

This is quite a complex input, from which most of the info has been derived. 'Rangoon' is not returned as it could be a corp or a place, 'Burmese' was already known. The only problem with the entry for 'manerplaw' is we have not derived who or what it is the headquarters of. The pattern 'loc of corp' was not held to contribute 'interesting' differentia information, and so no lattr entry was created. As 'hill' allows a namehead to be formed, only the PN component 'Htee Pawi Kyo' is entered into the lexicon.

8) The Arce Battalion Command has reported that about 50 peasants of various ages have been kidnapped by terrorists of the Farabundo Martii National Liberation Front (FMLN) in San Miguel department. The mass kidnapping took place in San Luis de la Reina. Meanwhile the Atonal Battalion reported that one extremist was killed during a clash yesterday afternoon near La Esperanza farm, Santa Elena jurisdiction, Usulutan department.

```
KB Entries:
kbase([san,miguel]):
[isa:[department]].

kbase([santa,elena]):
[isa:[jurisdiction],superpart:[usulutan]].

kbase([la,esperanza]):
[isa:[farm],superpart:[santa,elena]].

kbase(usulutan):
[isa:[department]].

Lexical Entries:
usulutan:
def([noun(usulutan,[loc,name],n)]).

[san,luis,de,la,reina]:
```

```
def([noun([san,luis,de,la,reina],[loc,name],n)]).

[san,miguel]:
def([noun([san,miguel],[loc,name],n)]).

[santa,elena]:
def([noun([santa,elena],[loc,name],n)]).

[la,esperanza]:
def([noun([la,esperanza],[loc,name],n)]).
```

This is an example taken from the MUC-3 development corpus (converted to mixed case). All the place PN's are acquired correctly. Inclusion of certain foreign words as place kw's (such as 'san') would be a possible addition to the FUNES lexicon. But there would be a problem if the system were then applies to South-East Asian news, where 'San' is also a personal name. In ther MUC-3 corpus, most of the locations are clearly flagged. Given its military purpose, and consequent need for clarity, this is understandable.

9) Sion Aubrey Roberts, 20, of Plas Tudur, Llangefni, and David Gareth Davies, 32, of Gwalchmai Uchaf, Gwalchmai, were remanded in custody by Holyhead magistrates, Anglesey, accused of conspiring to cause explosions.

```
KB Entries:
kbase([plas,tudur]):
[superpart:[llangefni]].

kbase([gwalchmai,uchaf]):
[superpart:[gwalchmai]].

Lexical Entries:
gwalchmai:
def([noun(gwalchmai,[loc,name],n)]).

llangefni:
def([noun(llangefni,[loc,name],n)]).

[plas,tudur]:
def([noun([plas,tudur],[loc,name],n)]).

[gwalchmai,uchaf]:
def([noun([gwalchmai,uchaf],[loc,name],n)]).

holyhead:
def([noun(holyhead,[loc,name],n)]).
```

This example shows the common pattern for personal PN's in court news items of 'Personal PN, Age, Origin'. The first 17 words of this sentence comprise a single NP (in certain grammars), showing the very high level of complexity of news stories. Without a set of PN patterns such as those presented in this thesis, the analysis of such a sentence would be virtually impossible. The appositive 'Holyhead magistrates, Anglesey,' shows an awkward example, where the appositive NP is actually describing the noun complement of the preceding NP, and not the head noun. FUNES can not handle such cases. It does however, utilise the role KW 'magistrates' to provide a place PN genus for the preceding word 'Holyhead'.

10) The multilevel railcars, scheduled for delivery in 1990, will be made by Thrall Manufacturing Co., a Chicago Heights, Ill., division of closely held Duchossois Industries Inc., Elmhurst, Ill.

This example, from the WSJ corpus, shows several counter-examples to the rules derived for UK news. Firstly we have the problem of the 'intervening' appositive, where an appositive NP intrudes into a NP that continues beyond it, so we have the NP 'a Chicago Heights division' with the appositive NP 'Ill' stuck in the middle. This leads to many errors in the semantic analysis of this appositive. Secondly the corp KW 'division' follows a place PN and not a corp PN. Thirdly, we have a corp PN used in the place PN appositive pattern — 'Duchossois Industries Inc, Elmhurst, Ill'. As 'Duchossois Ind Inc' will have a corp PN semantics, it will not activate the place PN type appositive, and so will end up asserting that the corp 'isa, Elmhurst'. The place-type appositive could easily be extended to cater for a 'Corp PN, Place PN, ' pattern if required. The entries acquired are therefore very misguided:

```
KB Entries:
kbase([duchossois,industries,inc]):
[isa:[elmhurst],property:([[closely,held]])]].

kbase(elmhurst):
[superpart:[ill]].

kbase([thrall,manufacturing,co]:
[isa:[heights]].

kbase(heights):
[superpart:[ill],property:([[chicago]])].

kbase(ill):
[isa:[division]].

Lexical Entries:
[thrall,manufacturing,co]:
def([noun([thrall,manufacturing,co],[corp,name],n)]).

heights:
def([noun(heights,[loc,name],n)]).

[duchossois,industries,inc]:
def([noun([duchossois,industries,inc],[corp,name],n)]).

elmhurst:
def([noun(elmhurst,[loc,name],n)]).
```

## L.2 Place KW's

Here we list all the place KW's used in the FUNES system. The nature of the semantic info accompanying these is more complex than for people or corps. This is because it aids in the namehead formation decision. The third argument carries this info. 'Loc' means that the KW optionally attaches to the PNcon it follows (where the KW precedes then it is always optional). 'Unit' means that it must attach, i. e. that the place PN containing this KW can not form a namehead. 'Building' covers all buildings where a namehead can be formed (those where one can not be formed are held as 'unit'). This argument is itself a term, in which the arguments give the relationship of the PNcon to the whole PN. 'Loc' means that the PNcon describes the location of the whole PN (e. g. Gatwick airport). This is often the same as 'name', which means the PNcon carries the name of the whole PN (e. g. Larry's Bar, H Block). 'Owner' means the PNcon names the owner, either a corp or a person, again this may be the same as 'name' (as in Larry's Bar), or it may be different (Olivetti factory).

Some of these arguments are not applicable as the KW will not occur with a preceding KW. In these cases the 3rd argument is just held as 'loc'.

```
def([noun(airport,[loc,comm,building(loc/name)],n)]).
def([noun(area,[loc,comm,loc],n)]).
def([noun(archive,[loc,comm,unit],n)]).
def([noun(avenue,[loc,comm,unit],n)]).
def([noun(back,[loc,comm,loc],n)],
noun(bar,[loc,comm,building(name/owner)],n)]).
def([noun(barrack,[loc,comm,building(name/loc)],n)]).
noun(base,[loc,comm,loc],n)]).
def([noun(bathroom,[loc,comm,building(prop)],n)]).
def([noun(bay,[loc,comm,unit],n)]).
def([noun(beach,[loc,comm,loc],n)]).
def([noun(bedroom,[loc,comm,building(prop)],n)]).
def([noun([birth,place],[loc,comm,loc],n)]).
noun(block,[loc,comm,building(name/loc)],n)]).
def([noun(border,[loc,comm,loc],n)]).
noun(bridge,[loc,comm,unit],n)]).
def([noun(cafe,[loc,comm,building(name/owner)],n)]).
def([noun(camp,[loc,comm,loc],n)]).
def([noun(capital,[loc,comm,loc],n)]).
def([noun(castle,[loc,comm,building(name/loc)],n)]).
def([noun(cathedral,[loc,comm,building(name/loc)],n)]).
def([noun(cell,[loc,comm,building(prop)],n)]).
noun(centre,[loc,comm,unit],n)]).
noun(center,[loc,comm,unit],n)]).
noun(chamber,[loc,comm,unit],n)],
noun(channel,[loc,comm,loc],n)]).
noun(church,[loc,comm,building(name/loc)],n)]).
def([noun(cinema,[loc,comm,loc],n)]).
def([noun(city,[loc,comm,unit],n)],
def([noun(coalfield,[loc,comm,loc],n)]).
def([noun(coast,[loc,comm,unit],n)]).
def([noun(college,[loc,comm,unit],n)]).
def([noun(colony,[loc,comm,loc],n)]).
def([noun(common,[loc,comm,unit],n)]).
def([noun(compound,[loc,comm,loc],n)]).
 noun(corner,[loc,comm,loc],n)]).
def([noun(country,[loc,comm,loc],n)]).
def([noun(department,[loc,comm,loc],n)]).
 noun(desert,[loc,comm,loc],n)]).
def([noun(destination,[loc,comm,loc],n)]).
def([noun(district,[loc,comm,loc],n)]).
 noun(drive,[loc,comm,unit],n)]).
def([noun(east,[loc,comm,loc],n)]).
noun(edge,[loc,comm,loc],n)]).
def([noun(enclave,[loc,comm,loc],n)]).
 noun(end,[loc,comm,loc],n)]).
def([noun(estate,[loc,comm,loc],n)],
def([noun(facility,[loc,comm,building(loc/name)],n)]).
def([noun(factory,[loc,comm,building(owner)],n)]).
 noun(farm,[loc,comm,loc],n)]).
def([noun(field,[loc,comm,loc],n)]).
def([noun(forest,[loc,comm,unit],n)]).
def([noun(frontier,[loc,comm,loc],n)]).
def([noun(garage,[loc,comm,building(name/owner)],n)]).
def([noun(garden,[loc,comm,unit],n)]).
def([noun(ghetto,[loc,comm,loc],n)]).
 noun(ground,[loc,comm,loc],n)]).
def([noun(gulf,[loc,comm,unit],n)]).
def([noun(hall,[loc,comm,building(name/loc)],n)]).
def([noun([headquarters],[loc,comm,building(owner)],n)]). % def for 'HQ'
def([noun([headquarters],[loc,comm,building(owner)],n)]).
% def for headquarters
def([noun([headquarters],[loc,comm,building(owner)],n)]).
% def for head-quarters or head quarters
def([noun(here,[loc,comm,loc],n)]).
def([noun(highway,[loc,comm,unit],n)]).
def([noun(hill,[loc,comm,unit],n)]).
def([noun(home,[loc,comm,building(name)],n)],
def([noun(homeland,[loc,comm,loc],n)]).
def([noun(hospital,[loc,comm,building(name/loc)],n)]).
def([noun(hotel,[loc,comm,building(name/loc)],n)]).
def([noun(house,[loc,comm,unit],n)]).
def([noun(infirmary,[loc,comm,building(loc/name)],n)]).
def([noun(interior,[loc,comm,loc],n)],
def([noun(island,[loc,comm,loc],n)]).
def([noun(isle,[loc,comm,loc],n)]).
 noun(jail,[loc,comm,building(loc/owner)],n)]).
def([noun(jurisdiction,[loc,comm,loc],n)]).
def([noun([labour,camp],[loc,comm,loc],n)]).
def([noun(lake,[loc,comm,loc],n)]).
 noun(land,[loc,comm,loc],n)]).
 noun(left,[loc,comm,loc],n)],
def([noun(line,[loc,comm,unit],n)]).
def([noun(marshes,[loc,comm,loc],n)]).
def([noun(mine,[loc,comm,loc],n)]).
def([noun(moon,[loc,comm,loc],n)]).
def([noun(mosque,[loc,comm,building(name)],n)]).
def([noun(mount,[loc,comm,loc],n)]).
def([noun([mountain,range],[loc,comm,loc],n)]). % for 'X mountains'
def([noun([mountain,top],[loc,comm,loc],n)]).
def([noun(museum,[loc,comm,unit],n)]).
def([noun(neighbourhood,[loc,comm,loc],n)]).
def([noun(north,[loc,comm,loc],n)]).
def([noun(ocean,[loc,comm,loc],n)]).
 noun(office,[loc,comm,building(name)],n)]).
def([noun(outskirt,[loc,comm,loc],n)]).
def([noun(palace,[loc,comm,unit],n)]).
def([noun(parish,[loc,comm,loc],n)]).
def([noun(park,[loc,comm,unit],n)]).
def([noun(passage,[loc,comm,loc],n)]).
def([noun(peninsula,[loc,comm,loc],n)]).
def([noun(penitentiary,[loc,comm,loc],n)]).
def([noun(pit,[loc,comm,loc],n)]).
 noun(place,[loc,comm,loc],n)]).
def([noun(plateau,[loc,comm,loc],n)]).
def([noun(platform,[loc,comm,loc],n)]).
def([noun([police,station],[loc,comm,loc],n)]).
def([noun(polytechnic,[loc,comm,building(loc)],n)]).
def([noun(port,[loc,comm,loc],n)]).
 noun(position,[loc,comm,loc],n)]).
def([noun(prefecture,[loc,comm,loc],n)]).
def([noun(prison,[loc,comm,building(loc/owner)],n)]).
def([noun(province,[loc,comm,loc],n)]).
def([noun(pub,[loc,comm,building(name)],n)]).
def([noun(pub,[loc,comm,building(name)],n)]). % def for 'public house'
def([noun(quarter,[loc,comm,loc],n)]).
def([noun(region,[loc,comm,loc],n)]).
def([noun(republic,[loc,comm,loc],n)]).
noun(reserve,[loc,comm,loc],n)]).
 noun(resort,[loc,comm,loc],n)]).
def([noun(restaurant,[loc,comm,building(name/owner)],n)]).
def([noun(river,[loc,comm,loc],n)]).
def([noun(road,[loc,comm,unit],n)]).
def([noun(room,[loc,comm,loc],n)]).
def([noun(school,[loc,comm,unit],n)]).
def([noun(sea,[loc,comm,unit],n)]).
noun(shed,[loc,comm,loc],n)]).
def([noun(shipyard,[loc,comm,loc],n)]).
def([noun(sky,[loc,comm,loc],n)]).
def([noun(south,[loc,comm,loc],n)]).
def([noun(square,[loc,comm,unit],n)],
  noun(station,[loc,comm,building(name/loc)],n)]).
def([noun([stock,exchange],[loc,comm,building(loc)],n)]).
noun(store,[loc,comm,loc],n)]).
def([noun(strait,[loc,comm,loc],n)]).
def([noun(stream,[loc,comm,loc],n)]).
def([noun(street,[loc,comm,unit],n)]).
def([noun(stronghold,[loc,comm,loc],n)]).
def([noun(studio,[loc,comm,building(name/owner)],n)]).
def([noun(surgery,[loc,comm,unit],n)]).
def([noun(territory,[loc,comm,loc],n)]).
def([prn(there,[[loc,comm,loc],_,_,n])]).
 def([noun(tip,[loc,comm,loc],n)]).
 def([noun(town,[loc,comm,loc],n)],
def([noun([town,hall],[loc,comm,building(loc)],n)]).
def([noun(township,[loc,comm,loc],n)]).
def([noun(university,[loc,comm,building(loc)],n)]).
def([noun(verge,[loc,comm,loc],n)]).
def([noun(village,[loc,comm,loc],n)]).
def([noun(west,[loc,comm,loc],n)]).
def([noun(wood,[loc,comm,unit],n)]).
def([noun(world,[loc,comm,loc],n)]).
def([noun(yard,[loc,comm,unit],n),
def([noun(zone,[loc,comm,loc],n)]).
```

# Appendix M

# Examples of Test Corpora

## M.1 Wall Street Journal Corpus

This comprised 100 stories selected from the ACL/DCI Wall Street Journal CD-ROM. The only criteria used was length — stories longer than 60 lines were excluded — and presence of PN's. The stories were not being used to assess numbers of PN's, but to test system on performance on handling PN's. Therefore stories with many different types of PN were preferred.

The FUNES pre-processor was adapted to enable it to cope with the particular input format of these stories.

Below we show a sample of the test corpus:

```
<DOC>
<DOCNO>  891102-0187. </DOCNO>
<DD> = 891102 </DD>
<AN> 891102-0187. </AN>
<HL> McDermott Completes Sale </HL>
<DD> 11/02/89 </DD>
<SO> WALL STREET JOURNAL (J) </SO>
<CO> MDR EUROP </CO>
<IN> TENDER OFFERS, MERGERS, ACQUISITIONS (TNM)
COMPUTERS AND INFORMATION TECHNOLOGY (CPR) </IN>
<DATELINE> NEW ORLEANS  </DATELINE>
<TXT>
<p>
<s> McDermott International Inc. said its Babcock & Wilcox unit completed the
sale of its Bailey Controls Operations to Finmeccanica S.p.A. for $295 million.
</s>
</p>
<p>
<s> Finmeccanica is an Italian state-owned holding company with interests in
the mechanical engineering industry. </s>
</p>
<p>
<s> Bailey Controls, based in Wickliffe, Ohio, makes computerized industrial
controls systems. </s>
<s> It employs 2,700 people and has annual revenue of about $370 million. </s>
</p>
```

```
</TXT>
</DOC>
<DOC>
<DOCNO>  891102-0164. </DOCNO>
<DD> = 891102 </DD>
<AN> 891102-0164. </AN>
<HL> Japan Investors Grab
@  2 Mortgage-Backed
@  Mutual Funds in U.S. </HL>
<DD> 11/02/89 </DD>
<SO> WALL STREET JOURNAL (J) </SO>
<CO> JAPAN </CO>
<IN> MUTUAL AND MONEY-MARKET FUNDS (FND) </IN>
<DATELINE> TOKYO  </DATELINE>
<TXT>
<p>
<s> Japanese investors nearly single-handedly bought up two new mortgage
securities-based mutual funds totaling $701 million, the U.S. Federal National
Mortgage Association said. </s>
</p>
<p>
<s> The purchases show the strong interest of Japanese investors in U.S.
 mortgage-based instruments, Fannie Mae's chairman, David O. Maxwell, said at a
news conference. </s>
<s> He said more than 90% of the funds were placed with Japanese institutional
investors. </s>
</p>
<p>
<s> Earlier this year, Japanese investors snapped up a similar, $570 million
mortgage-backed securities mutual fund. </s>
<s> That fund was put together by Blackstone Group, a New York investment bank.
 </s>
<s> The latest two funds were assembled jointly by Goldman, Sachs & Co. of the
U.S. and Japan's Daiwa Securities Co. </s>
</p>
</TXT>
</DOC>
<DOC>
<DOCNO>  891102-0185. </DOCNO>
<DD> = 891102 </DD>
<AN> 891102-0185. </AN>
<HL> Who's News:
@  Mazda Motor of America Inc. </HL>
<DD> 11/02/89 </DD>
<SO> WALL STREET JOURNAL (J) </SO>
<CO> J.MZD WNEWS </CO>
<DATELINE> MAZDA MOTOR OF AMERICA Inc. (Irvine, Calif.)  </DATELINE>
<TXT>
<p>
```

```
<s> Clark J. Vitulli was named senior vice president and general manager of
this U.S. sales and marketing arm of Japanese auto maker Mazda Motor Corp. </s>
<s> In the new position he will oversee Mazda's U.S. sales, service, parts and
marketing operations. </s>
<s> Previously, Mr. Vitulli, 43 years old, was general marketing manager of
Chrysler Corp.'s Chrysler division. </s>
<s> He had been a sales and marketing executive with Chrysler for 20 years.
</s>
</p>
</TXT>
</DOC>
<DOC>
<DOCNO> 891102-0181. </DOCNO>
<DD> = 891102 </DD>
<AN> 891102-0181. </AN>
<HL> Bid Is Dropped
@  By New England
@  Electric System
@  ----
@  By Lawrence Ingrassia
@  Staff Reporter of The Wall Street Journal </HL>
<DD> 11/02/89 </DD>
<SO> WALL STREET JOURNAL (J) </SO>
<CO> PNH NES NU UIL </CO>
<IN> BANKRUPTCIES (BCY)
TENDER OFFERS, MERGERS, ACQUISITIONS (TNM)
UTILITIES (UTI) </IN>
<TXT>
<p>
<s> New England Electric System bowed out of the bidding for Public Service
Co. of New Hampshire, saying that the risks were too high and the potential
payoff too far in the future to justify a higher offer. </s>
</p>
<p>
<s> The move leaves United Illuminating Co. and Northeast Utilities as the
remaining outside bidders for PS of New Hampshire, which also has proposed
an internal reorganization plan in Chapter 11 bankruptcy proceedings under
which it would remain an independent company. </s>
</p>
<p>
<s> New England Electric, based in Westborough, Mass., had offered $2
billion to acquire PS of New Hampshire, well below the $2.29 billion value
United Illuminating places on its bid and the $2.25 billion Northeast says
its bid is worth. </s>
<s> United Illuminating is based in New Haven, Conn., and Northeast is based
in Hartford, Conn. </s>
<s> PS of New Hampshire, Manchester, N.H., values its internal reorganization
plan at about $2.2 billion. </s>
</p>
```

```
<p>
<s> John Rowe, president and chief executive officer of New England Electric,
said the company's return on equity could suffer if it made a higher bid and
its forecasts related to PS of New Hampshire -- such as growth in electricity
demand and improved operating efficiencies -- didn't come true. </s>
</p>
<p>
<s> Wilbur Ross Jr. of Rothschild Inc., the financial adviser to the troubled
company's equity holders, said the withdrawal of New England Electric might
speed up the reorganization process. </s>
</p>
<p>
<s> Separately, the Federal Energy Regulatory Commission turned down for now a
request by Northeast seeking approval of its possible purchase of PS of New
Hampshire. </s>
<s> Northeast said it would refile its request and still hopes for an expedited
 review by the FERC so that it could complete the purchase by next summer if
its bid is the one approved by the bankruptcy court. </s>
</p>
</TXT>
</DOC>
<DOC>
<DOCNO>  891102-0180. </DOCNO>
<DD> = 891102 </DD>
<AN> 891102-0180. </AN>
<HL> Who's News:
@  Circuit City Stores Inc. </HL>
<DD> 11/02/89 </DD>
<SO> WALL STREET JOURNAL (J) </SO>
<CO> CC WNEWS </CO>
<DATELINE> CIRCUIT CITY STORES Inc. (Richmond, Va.)  </DATELINE>
<TXT>
<p>
<s> Norman Ricken, 52 years old and former president and chief operating
officer of Toys "R" Us Inc., and Frederick Deane Jr., 63, chairman of Signet
Banking Corp., were elected directors of this consumer electronics and
appliances retailing chain. </s>
<s> They succeed Daniel M. Rexinger, retired Circuit City executive vice
president, and Robert R. Glauber, U.S. Treasury undersecretary, on the
12-member board. </s>
</p>
</TXT>
</DOC>
```

## M.2 UK News Corpus

This was a manually-entered set of 100 short news stories from the UK papers The Daily Telegraph, The Times, The Guardian, The Independent, The Financial Times, and their

Sunday equivalents. The same criteria were used as above for selecting the stories. Apart from these two criteria the selection was made as randomly as possible.

Below is a sample of the corpus.

```
Albanian police were given orders to shoot and the army distributed bread
yesterday after two people died in food riots. "Looting and robbery of state
shops and factories continued throughout Saturday despite large-scale
mobilisation of security forces", the Ministry of Public Order said.
There was unrest in the town of Lac, north-east of Tirana, where a
policeman and a civilian were killed in food riots on Saturday. In the town
of Reshen mobs attacked warehouses and state shops on Saturday, the ministry
said. "Security forces intervened energetically and the crowd were halted,"
it added.
In Tirana, partly blacked out by power shortages, army trucks took
over bread distribution and police escorted bakery vans.
##
Police had to free governors of the North London Polytechnic who barricaded
themselves in a boardroom at the James Leicester Hall in Market Road,
Holloway, last night to escape a student picket.
The students were trying to disrupt a governor's meeting in protest
at overcrowding and underfunding. Four of the polytechnic's six sites are
now being occupied. The president of the students' union, Kay Bridge, said:
"It was a peaceful picket. We just wanted to express our feelings. Last
year we had 7,500 students at the polytechnic; this year there are 8,500
and there has been no increase in resources.
"They are just trying to squash as many of us in as they can."
##
John Drummond, Controller of BBC Radio 3, and artistic director of the Proms,
is to be Director of the European Arts Festival, which will have a budget of
6 million pounds. The festival will celebrate the UK's Presidency of the
European Community from 1 July to 31 December 1992.
##
Britain's largest National Nature Reserve is to be created by English Nature,
the Government's advisory body on the environment, on 4,000 hectares of the
Ribble Marshes in Lancashire.
##
Jalal Talabani, the Iraqi Kurdish leader, appealed for international
intervention to help 800,000 Kurdish refugees facing harsh winter , lack of
food and medicine and the threat of an Iraqi attack, write Hugh Pope.
##
The family of the late reggae singer Bob Marley was granted the right to buy
his estate by the Jamaican Supreme Court, at the end of a complicated legal
battle, writes Zoe Heller.
Mr Justice Walker rule in favour of an $11m (6.1 million pound) bid
posted by Island Logic, a company owned by Marley's former record producer,
Chris Blackwell, and supported by his wife, Rita Marley, his mother, Cedella
Booker, and his 11 children.
##
A fresh controversy over police conduct has erupted in California after
officers shot an unarmed black man 28 times, killing him instantly, writes
```

Phil Reeves. The victim, Darryl Stephens, 27, was not wanted or classed as a suspect.
Last year four officers in Los Angeles were filmed beating a black motorist after dragging him from his car.
The shooting took place during a night raid on an apartment complex by West Covina police investigating murders, kidnappings and robberies in the San Gabriel Valley, near Los Angeles.
##
President F W de Klerk announced a surprise cabinet reshuffle, bringing in a senior business leader as Minister of Trade and Industry and Economic Co-ordinator. In a statement, Mr de Klerk said Derek Kys, chairman of the giant General Mining Union Corporation (Gencor), would replace the current Trade and Industry Minister, Org Marais.
##
President Albert Rene of the Seychelles said the country would adopt a multi-party political system, abandoning 14 years of single-party rule.
The President had resisted pressure over the past year to follow an Africa-wide trend away from single-party rule.
##
Togo's reformist prime minister, Joseph Kokou Koffigoh, captured in a coup by rebellious troops, was still at the helm yesterday. Diplomats said he would have to make concessions to hardliners from the former ruling party.
##
Finals results of Sunday's Paraguayan constituent assembly elections confirmed a comfortable victory for the ruling Colorado Party, which gained 57 per cent of the votes, against 28 per cent for the opposition Authentic Radical Liberal Party, writes Colin Harding. The Colorados will have 123 of the 198 seats in the assembly, which has to draft a replacement for the 1967 constitution within six months.
##
The Australian opposition leader , John Hewson, has surged ahead of the Prime Minister, Bob Hawke. An opinion poll published yesterday shows support for the opposition coalition rose 2.5 prcentage points to 47 percent against Labor's 37, a rise of one point from two weeks ago. Mr Hewson's rating as preferred prime minister jumped seven points to 45 per cent while Mr Hawke's fell four to 43 per cent.
##
President Mikhail Gorbachev could lose his summer house as a result of the Ukrainian independence vote, Komsomolskaya Pravda said. The paper quoted the Ukrainian President, Leonid Kravchuk, as saying his government would own all property on its territory - including the Crimean dacha. "We could also consider the possibility of selling it, but for the standard prices," he added.

# Appendix N

# Formal Model of the Syntax and Semantics of Proper Names

In this appendix we show all the syntactic patterns and their corresponding semantics which were not actually shown in chapter 6.

## N.1 Personal PN's

Conjunction Patterns for Personal PN's:

1) Conjunction of the appositive pattern:

i) **X ⟨and⟩ X ⟨comma⟩ plural_KW_NP (PP) ⟨comma⟩**
e. g. Marcel Proust and James Joyce, arguably the most significant authors of the 20th century,
ii) **X ⟨comma⟩ KW_NP1 (PP1) ⟨and⟩ KW_NP2 (PP2) ⟨comma⟩**
e. g. 'Pierre Vallin, the town's mayor and a company director,'.
iii) **plural_KW_NP (PP) ⟨comma⟩ X ⟨and⟩ X ⟨comma⟩**
e. g. The authors of the report, Tim Jenkinson and Colin Meyer,
iv) **KW_NP2 (PP2) ⟨and⟩ KW_NP2 (PP2) ⟨comma⟩ X ⟨comma⟩**
e. g. Chairman and chief executive, Steve Scott,

2) Conjunction following a plural preceding KW:

> v) **plural_KW_NP X ⟨and⟩ X**
> e. g. Presidents Mitterand and Bush

It is easier to see these patterns as giving rise to two different NDF's — one applies two groups of descriptive information to a singular PN, the other applies a single item of information to two different PN's. Thus:

> ROLE NDF3 (the semantics for patterns ii and iv):
> PN Genus = [human,name]
> PN role = KW1 ∧ KW2
> PN gender = KW gender
> PN differentia = sem(Adj1*) ∧ sem(Noun_comp1*) ∧ sem(PP1)
> ∧ sem(Adj2*) ∧ sem(Noun_comp2*) ∧ sem(PP2)

and

ROLE NDF4 (the semantics for patterns i, iii and v):
PN1 Genus = [human,name] ∧ PN2 Genus = [human,name]
PN1 role = KW ∧ PN2 role = KW
PN1 gender = KW gender ∧ PN2 gender = KW gender
PN1 differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem(PP)
∧ PN2 differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem(PP)

3) Conjunction of both PN and descriptive NP:

**X ⟨and⟩ X ⟨comma⟩ KW_NP1 (PP1) ⟨and⟩ KW_NP2 (PP2) ⟨comma⟩** e. g.
1) Peter Marks and Andrew Winstanley, the president and treasurer,
2) Peter Marks and Andrew Winstanley, the president and treasurer of Amoco corp,
3) Peter Marks and Andrew Winstanley, the Amoco president and treasurer,
**X ⟨and⟩ X ⟨comma⟩ plural_KW_NP (PP) ⟨and⟩ PP/NP ⟨comma⟩** e. g.
4) Paddy Ashdown and Neil Kinnock, the leaders of the Liberal Democrat and (of the) Labour parties (respectively),

We have produced a partial semantics for this type of conjunction:

ROLE NDF5 :
PN1 Genus = [human,name] ∧ PN2 Genus = [human,name]
PN1 role = KW1 ∧ PN2 role = KW2
PN1 gender = KW1 gender ∧ PN2 gender = KW2 gender

## N.2 Corp PN's

The NDF's for conjunction of apposition patterns are the same as those for personal and place PN's, i. e. they are identical to the non-conjoined NDF except that either two descriptive NP's or two corp PN's are involved.

The NDF for syntactic patterns 3 where we have two separate PN's is:

PN1= Init_cap nouns from plural KW_NP
PN2= Init_cap nouns from conjoined NP
PN1 Genus = [corp,name] ∧ PN2 Genus = [corp,name]
PN1 Gen = n ∧ PN2 Gen = n
PN1 differentia = sem(Adj) ∧ sem(Noun_comp)
PN2 differentia = sem(Adj) ∧ sem(Noun_comp)

Where a single corp PN is given (as in 'the building society Bradford and Bingley') then the NDF is exactly the same as NDF1. The PN is formed from the init_cap nouns from the corp KW_NP and the init_cap nouns from the conjoined NP.

NDF for conjunction of Pattern 4. The conjunction of two separate corps reveals NDF:

PN1= (Adj1*) Noun_comp1* KW
PN2= (Adj2*) Noun_comp2* KW
PN1 Genus = [corp,name] ∧ PN2 Genus = [corp,name]
PN1 Gen = n ∧ PN2 Gen = n
PN1 differentia = sem(Adj1) ∧ sem(Noun_comp1)
PN2 differentia = sem(Adj2) ∧ sem(Noun_comp2)

The pattern for a single PN containing a conjunction has an almost identical semantics to CORP NDF2, except that we have two potential Adj and Noun Comp groups. Cases of two conjoined corps, where one or both also has a conjunction in its name (e. g. The Licensed Brewers and Victuallers and Port and Harbour Authorities), have (thankfully) not been observed. As we only seek to describe observed cases we ignore those cases which exist purely in the theoretical realm.

Conjunction Pattern for Corp PN Pattern 5:

a) **X ⟨and⟩ X plural KW_NP**
e. g. the Abbey National and Halifax building societies
b) **Pnoun* ⟨and⟩ Pnoun* KW_NP**
e. g. Bradford and Bingley building society

In a) the PN's are formed from the first NG, and from the init_cap nouns in the conjoined NP. The semantics are the same as those for pattern 3. Where we have a single PN (case b) it is formed from the first NG and the init_cap nouns of the conjoined NP. Its semantics are the same as NDF1.

Conjunction Patterns for Corp PN Pattern 6:

**Plural_KW_NP PP1* ⟨and⟩ PP2*/ (Det) (Adj*) Noun (PP2*)**, where
PN1 = Plural_KW_NP (with plural KW in singular form) PP1*
PN2 = Plural_KW_NP (with plural KW in singular form) PP2*/(Det) (Adj) Noun (PP2*)
e. g. The Societies for Abolition of Vivisection and Legislation on Medical Experimentation.

The semantics for this pattern as the same as Corp NDF 2, except that PN1 is described by sem(PP1*), and PN2 by sem(PP2).

More common is a single such name involving a conjunction, which follows the pattern:

**KW_NP PP1* ⟨and⟩ (Det) (Adj*) Noun (PP2*)**, where
the PN is comprised of the entire pattern
e. g. The Association of Futures Brokers and Dealers
The National Association for the Care and Resettlement of Offenders

Its semantics are the same as corp NDF 2, except that sem(PP1*) and sem(PP2*) contribute differentia information.

## N.3 Place PN's

Syntactic Pattern for the Dir_NP used in Place PN Patterns:

**(Adv) ((Number) Measure) Direction ⟨of⟩ Ref_NP**
e. g. south of the boarder
e. g. some (200) miles south of the boarder
**(Adv) (Number) Measure ⟨from⟩ Ref_NP**
e. g. only 20 miles from the capital
**(Adv) (Number) Measure (Adv) (to) (Det) Direction ⟨of⟩ Ref_NP**
e. g. (about) 200 miles to the south of Basra
e. g. 50 kilometres directly north of Kuwait
e. g. a hundred miles due west of the coast

Number → Digit*/a
Measure → mile/kilometre/yard/metre
Direction → south/north/east/west/north-west etc
Ref_NP → (Det) (Adj*) Noun/PN

Semantics for Place PN Conjunction Patterns:

Place NDF4 :
PN1 Genus = [loc,name] ∧ PN2 Genus = [loc,name]
PN1 supertype = KW ∧ PN2 supertype = KW
PN1 gender = n ∧ PN2 gender = n
PN1 differentia = sem(Adj*) ∧ sem(Dir_NP) ∧ sem(PP)
∧ PN2 differentia = sem(Adj*) ∧ sem(Dir_NP) ∧ sem(PP)

## N.4 Legis PN's

Legis PN Conjunction Patterns:

Pattern 1:
**Det (Adj1*) Noun_comp1* ⟨and⟩ (Adj2*) Noun_comp2* plural_KW**
where PN1= (Adj1*) Noun_comp1* KW
and PN2= (Adj2*) Noun_comp2* KW

Pattern 2:
**Det Plural_LegisKW PP1 and (Prep) Noun***
where PN1=Det LegisKW PP1
and PN2=Det LegisKW Prep Noun*

Pattern 3:
**Det (Adj*) Noun_comp1* ⟨and⟩ (Adj*) Noun_comp2* KW**
where the whole pattern is the PN

Patterns 1 and 2 have essentially the same semantics:

PN1 Genus = [legis,name] ∧ PN2 Genus = [legis,name]
PN1 supertype = KW ∧ PN2 supertype = KW
PN1 Gen = n ∧ PN2 Gen = n
PN1 differentia = sem(Noun_comp1*) ∧ sem(PP1) ∧
PN2 differentia = sem(Noun_comp2*) ∧ sem(Prep Noun*)

The semantics for Pattern 3:

PN Genus = [legis,name]
PN Supertype = KW
PN Gen = n
PN differentia = sem(Noun_comp1*) ∧ sem(Noun_comp2*)

## N.5 Isource PN's

Isource PN Conjunction Patterns:

**Det (Adj*) plural i_sourceKW X ⟨and⟩ X**
**X ⟨and⟩ X plural i_sourceKW**

Semantics for Isource Conjunction Patterns, (Isource NDF3):

PN1 Genus = [isource,name] ∧ PN2 Genus = [isource,name]
PN1 supertype = KW ∧ PN2 supertype = KW
PN1 Gen = n ∧ PN2 Gen = n
PN1 differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧
PN2 differentia = sem(Adj*) ∧ sem(Noun_comp*)

It is possible to have a single Isource PN containing a conjunction, but we have never encountered any. It is also possible to have isources with PP's involved in elliptical conjunction, e. g. the Journals of Applied and Clinical Psychology, but we have excluded such types as outside the scope of this thesis.

Common Syntactic Contexts for English Newspaper PN's:

1. **⟨writes/writing/wrote/written⟩ ⟨in/for⟩ X**, e. g. John Thornhill, writing in the **Times**,

2. **(Det) i_source_noun ⟨in⟩ (Det/(Time_expr) Day X**, e. g. a report in the **Daily Telegraph**, an article in yesterday's **Independent**

3. **(Det) ⟨editor/publisher/correspondent/journalist⟩ ⟨with/of/for/⟩ X** , e. g. Mr Maxwell, the former publisher of the **Daily Mirror,**

i_source_noun → report/article/editorial/story ...
Day → monday's ... sunday's/yesterday's/today's/tomorrow's
Time_expr → next/last/this

These patterns give rise to a not very detailed NDF, I_source NDF1:

PN Genus = [i_source,name]
PN Gen = n

The KW 'daily' presents some problems for implementation in that it often forms part of the name of the paper, in English language newspapers, whereas in foreign papers it is used solely to indicate a following PN. This is clearly indicated though by its case, when part of a name it is capitalised, when not it is in normal case.

## N.6 Event PN's

Event PN Conjunction Patterns:

**Det (Adj1*) (Noun_comp1*) ⟨and⟩ (Adj2*) (Noun_comp2*) plural_EventKW**
where PN1 = (Adj1*) (Noun_comp1*) EventKW
and PN2 = (Adj2*) (Noun_comp2*) EventKW

This pattern has the following semantics:

PN1 Genus = [event,name] ∧ PN2 Genus = [event,name]
PN1 supertype = KW ∧ PN2 supertype = KW
PN1 gender = n ∧ PN2 gender = n

## N.7 Object PN's

Object PN Conjunction Patterns:

**(Det) X ⟨and⟩ (Det) X ⟨comma⟩ Plural_KW_NP (PP) ⟨comma⟩**
e. g. the Sea Dart and Sea Wolf, the most effective anti-missile systems in the world,
**Plural_KW_NP (PP) ⟨comma⟩ (Det) X ⟨and⟩ (Det) X ⟨comma⟩**
e. g. Ford's most popular family cars, The Cavalier and Escort,
**Plural_KW_NP (Det) X ⟨and⟩ (Det) X**
e. g. The nuclear submarines Nautilus and Seaview
**(Det) Pnoun* ⟨and⟩ (Det) Pnoun* Plural_KW**
e. g. The Gemini and Mercury space probes

These patterns all have the same semantics — Object NDF3:

PN1 Genus = [object,name] ∧ PN2 Genus = [object,name]
PN1 supertype= KW ∧ PN2 supertype = KW
PN1 Gen = n ∧ PN2 Gen = n
PN1 differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem (PP)
PN2 differentia = sem(Adj*) ∧ sem(Noun_comp*) ∧ sem (PP)

# Bibliography

[1] Association for Computational Linguistics Data Collection Initiative CD-ROM 1. Available from Mark Liberman or Don Walker at the ACL.

[2] F. Abate, editor. *Omni Gazetteer of the United States of America.* Omnigraphics, 1991.

[3] R. Agaarwal and L. Boggess. A simple but useful approach to conjunct identification. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 1992.

[4] J. Allen. *Natural Language Processing.* Benjamin Cummings, 1987.

[5] D. J. Allerton. The linguistic and sociolinguistic status of proper names. *The Journal of Pragmatics*, 1987.

[6] D. Allport. The TICC: Parsing interesting text. In *Proceedings of the 2nd Conference on Applied NLP*, 1988.

[7] H. Alshawi. *Memory and Context for Language Interpretation.* Cambridge University Press, 1987.

[8] H. Alshawi. Analysing the dictionary definitions. In Boguraev and Briscoe, editors, *Computational Lexicography for Natural Language Processing.* Lawrence Erlbaum, 1989.

[9] H. Alshawi, editor. *The Core Language Engine.* MIT Press, 1992.

[10] R. A. Amsler. Research towards the development of a lexical knowledge base for natural language processing. *SIGIR Forum*, 23(1–2), 1989.

[11] P. M. Andersen, P. J. Hayes, et al. Automatic extraction of facts from press releases to generate news stories. In *3rd Conference on Applied NLP*, 1992.

[12] J. R. Anderson. The induction of ATNs. *Artficial Intelligence*, 1977.

[13] J. R. Anderson. A theory of language acquisition based on general learning principles. In *Proceedings of the IJCAI*, 1981.

[14] Association for Computational Linguistics. *The Third Conference on Applied Natural Language Processing*, 1992.

[15] D. Ayuso, R. Bobrow, et al. Toward understanding text with a very large vocabulary. In *Proceedings of the DARPA Speech and Natural Languag Workshop*, June 1990.

[16] B. Ballard and D.E. Stumberger. TELI : A transportable, user-customised natural language processor. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, 1986.

[17] R. Berwick. Learning word meanings from examples. In *Proceedings of the IJCAI*, 1983.

[18] R. Berwick. *The Acquisition of Syntactic Knowledge.* MIT press, 1985.

[19] B. Boguraev. *Automatic Resolution of Linguistic Ambiguities.* PhD thesis, Cambridge University, 1979. Available as Tech Report no. 11, Computer Laboratory, Cambridge University.

[20] B. Boguraev and T. Briscoe, editors. *Computational Lexicography for natural language processing.* Longman, 1989.

[21] B. Boguraev and B. Levin. Models for lexical knowledge bases. In *Proceedings of the 6th University of Waterloo Conference on the New OED and Electronic Text Researach*, 1990.

[22] J. L. Borges. FUNES the memorious. In *Ficciones.* Emece Editores, 1956.

[23] M. R. Brent. Automatic acquisition of subcategorisation frames from untagged text. In *Proceedings of the 29th Meeting of the ACL*, 1991.

[24] P. Brown, J. Cocke, et al. A statistical approach to language translation. In *Proceedings of the International Congress on Computational Linguistics, Budapest (COLING 88)*, 1988.

[25] A. Burton. *A Sublanguage of English for Database Query in a Managerial Enviroment.* PhD thesis, Sunderland Polytechnic, 1991.

[26] R. Byrd. Word formation in natural language processing systems. In *Proceedings of the IJCAI*, 1983.

[27] J. Carbonell. Towards a self-extending parser. In *Proceedings of the 17th meeting of the ACL*, 1979.

[28] J. Carbonell. POLITICS : An experiment in subjective understanding and integrated reasoning. In R. Schank and C. Riesbeck, editors, *Inside Computer Unserstanding.* Lawrence Erlbaum, 1981.

[29] J. Carbonell and P. Hayes. Robust parsing using multiple construction-specific strategies. In L. Bolc, editor, *Natural Language Parsing Systems.* Springer-Verlag, 1987.

[30] J. Carroll. *What's in a Name.* Freeman, 1985.

[31] D. Carter. Lexical acquisition in the Core Language Engine. In *Proceedings of the IJCAI*, 1989.

[32] D. Carter. Lexical analysis. In *The Core Language Engine.* MIT Press, 1992.

[33] N. Castell and M. Felisa Verdejo. Automatic extraction of factual information from text and its integration in a knowledge base. In *RIAO 91 — Conference on Intelligent Text and Image Handling*, 1991.

[34] E. Charniak. Passing markers : a theory of contextual influence in language comprehension. *Cognitive Science*, 7, 1983.

[35] E. Charniak. A neat theory of marker passing. In *Proceedings of the AAAI*, 1986.

[36] N. Chinchor. MUC-3 lingusitic phenomena test experiment. In *Proceedings of the 3rd Message Understanding Conference.* Morgan Kaufmann, 1991.

[37] M. V. Chitrao and R. Grishman. Statistical parsing of messages. In *Proceedings of the DARPA Speech and Natural Language Workshop*, June 1990.

[38] Y. Choueka. Looking for needles in a haystack. In *RIAO 88*, 1988.

[39] K. Church, W. Gale, et al. Using statistics in lexical analysis. In U. Zernik, editor, *Lexical Acquisition: exlpoiting on-line resources to build a lexicon.* LEA, 1991.

[40] K. Church and P. Hanks. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1), 1990.

[41] S. Coates-Stephens. Expectation based word learning. Technical Report TCU/CS/1990/7, City University Dept Of Computer Science, 1990.

[42] S. Coates-Stephens. A review of word learning and implementation of an inference-based word learner. Master's thesis, City University, 1990.

[43] S. Coates-Stephens. Automatic acquisition of proper noun meanings. In Z. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems - 6th International Symposium, ISMIS '91.* Springer Verlag, 1991.

[44] S. Coates-Stephens. Automatic lexical acquisition using within text descriptions of proper nouns. In *Proceedings of the 7th Conference of the UW Centre for the New OED and Text Research: Using Corpora*, 1991.

[45] S. Coates-Stephens. Lexical acquisition of proper nouns as a by-product of text processing. In David Powers, editor, *IJCAI 1991 Workshop on Natural Language Learning*, 1991.

[46] S. Coates-Stephens. The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, 26(5–6), 1993.

[47] A. Copestake. The ACQUILEX LKB : Representation issues in semi-automatic acquisition of large lexicons. In *3rd Conference on Applied NLP*, 1992.

[48] R. Cullingford. SAM. In B. Grosz, K. Sparck-Jones, and B. Webber, editors, *Readings in Natural Language Processing.* Morgan Kaufmann, 1986.

[49] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part of speech tagger. In *Proceedings of the 3rd Conference on Applied NLP*, 1992.

[50] DARPA. *Proceedings of the 3rd Message Understanding Conference.* Morgan Kaufmann, 1991.

[51] G. Dejong. Skimming news stories. In W. Lehnert and C. Ringle, editors, *Strategies for Natural Language Processing.* Lawrence Elrbaum, 1982.

[52] G. DeJong and R. Mooney. Learning schemata for natural language processing. In *Proceedings of the IJCAI*, 1985.

[53] D. Diaper. Identifying the knowledge requirements of an expert system's natural language processing interface. In M. Harrison and A. Monk, editors, *People and Computers: Designing for Usability.* Cambridge University Press, 1986.

[54] M. Dyer. *In-depth understanding.* MIT press, 1983.

[55] K. Eiselt. Recovering from erroneous inference. In *Proceedings of the AAAI*, 1987.

[56] W. Emde. Managing lexical knowledge in LEU/2. In O. Herzog and C. R. Rollinger, editors, *Text Understanding in LILOG.* Springer Verlag, 1992.

[57] M. Evans and R. Wimmer. Searle's theory of proper names, from a linguistic point of view. In *Speech Acts, Meanings and Intentions - critical approaches to the philosophy of John R Searle.* Walter de Gruyter, 1990.

[58] C. J. Fillmore. The case for case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory.* Holt, Rineheart, and Winston, 1968.

[59] T. W. Finin. The semantic interpretation of compound nominals. In *Proceedings of the AAAI*, 1980.

[60] G. Frege. Uber Sinn und Bedeutung. Zeitschrift fur Philosophie und philosophische Kritik NF 100. In G. Patzig, editor, *Funktion, Begriff und Bedeuteung.* Vandenhoeck & Ruprecht, 1980.

[61] M Frixone, S. Gaglio, and G. Spinelli. Are there individual concepts ? Proper names and individual concepts in SI-Nets. *International Journal of Man-Machine Studies*, 30, 1989.

[62] N. Fung-a-Fat. Acquisition of language semantics and base level categories. Master's thesis, New Jersey Institute of Technology, 1991.

[63] R. Garside and G. Leech. Grammatical tagging of the LOB corpus. In S. Johansson, editor, *Computer Corpora in English Langauge Research.* Norwegian Computing Centre for the Humanities, 1982.

[64] G. Gazdar, E. Klein, G. Pullman, and I. Sag. *Generalised Phrase Structure Grammar.* Harvard University Press, 1985.

[65] G. Gazdar and C. Mellish. *Natural Language Processing in Prolog.* Addison Wesley, 1989.

[66] J. Geller. Acquisition of attribute applicability. In David Powers, editor, *IJCAI 91 Workshop on Natural Language Learning*, 1991.

[67] A. Gershman. Conceptual analysis of noun groups in English. In *Proceedings of the IJCAI*, 1977.

[68] A. Gershman. A framework for conceptual analysers. In W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing.* Lawrence Erlbaum, 1982.

[69] R. Granger. FOUL-UP : a program that figures out meanings of words from context. In *Proceedings of the IJCAI*, 1977.

[70] R. Granger. The NOMAD system : Expectation-based detection and correction of syntactically and semantically ill-formed text. *Journal of Computational linguistics*, 9(3–4), 1983.

[71] R. Grishman, J. Sterling, and C. Macleod. New York University PROTEUS system: MUC-3 test results and analysis. In *Proceedings of the 3rd Message Understanding Conference*, 1991.

[72] A. Gross. Getty Synoname: The development of software for personal name pattern matching. In *Proceedings of the Conference on Intelligent Text and Image Handling*, 1991.

[73] B. J. Grosz. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the IJCAI*, 1977.

[74] B.J. Grosz, D. Appelt, P. Martin, and F. Pereira. TEAM : An experiment in the design of transportable natural-language interfaces. *Artificial Intelligence*, 32, 1987.

[75] S. Harnad. The symbol grounding problem. *Physica D*, 42, 1990.

[76] L. R. Harris. A system for primitive natural langauge acquisition. *Int. Journal of Man-Machine Studies*, (9), 1977.

[77] P. Hastings, S. Lytinen, and R. Lindsay. Learning words from context. In *Proceedings of the Conference on Machine Learning*, 1991.

[78] P. Hayes, L. Knecht, and M. Cellio. A news story categorisation system. In *Proceedings of the 2nd Conference on Applied NLP*, 1988.

[79] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 92*, Nantes, France, July 1992.

[80] J. Hendler. *Integrating Marker Passing and Problem Solving.* Morgan kaufman, 1987.

[81] O. Herzog and C. R. Rollinger, editors. *Text Understanding in LILOG — Integrating Computational Linguistics and Artificial Intelligence. Final Report on the IBM Germany LILOG project.* Springer Verlag, 1992.

[82] D. Hindle. Deterministic parsing of syntactic non-fluencies. In *Proceedings of the 21st meeting of the ACL*, 1983.

[83] D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th Meeting of the ACL*, 1990.

[84] G. Hirst. *Anaphora in Natural Language Understanding.* Springer-Verlag, 1981. Lecture Notes in Computer Science 119.

[85] G. Hirst. *Semantic Interpretation and the Resolution of Ambiguity.* Cambidge University Press, 1987.

[86] J. R. Hobbs. Resolving pronoun references. *Lingua*, 44, 1978.

[87] J. R. Hobbs. Description of the TACITUS system as used for MUC-3. In *Proceedings of the 3rd Message Understanding Conference*, May 1991.

[88] J. R. Hobbs, D. Appelt, et al. Robust processing of real-world natural-language texts. In *Proceedings of the 3rd Conference on Applied NLP*, April 1992.

[89] R. Hudson. *Word Grammar*. Blackwell, 1984.

[90] H. Ishara and S. Ishizaki. Computerized analysis of syntactic and semantic information in Japanese newspaper articles. *Bulletin of the Electrotechnical Lab (Japan)*, 54(3), 1990.

[91] R. Jackendoff. *Semantics and Cognition*. MIT press, 1983.

[92] P. Jacobs, editor. *Text-Based Intelligent Systems: Current Reserch in Text Analysis, Information Extraction, and Retrieval*, September 1990. Available as GE Tech Report no. 90CRD198.

[93] P. Jacobs. From parsing to database generation : Applying natural language systems. In *Proceedings of the 7th IEEE Conference on AI Applications*, 1991.

[94] P. Jacobs, G. Krupka, et al. Generic text processing : a progress report. In *DARPA Speech and Natural Language Workshop*, June 1990.

[95] P. Jacobs, G. Krupka, and L. Rau. Lexico-semantic pattern matching as a companion to parsing in text understanding. In *DARPA Speech and Natural Language Workshop*, February 1991.

[96] P. Jacobs and L. Rau. SCISOR: Extracting information from online news. *Communications of the ACM*, 33(11), November 1990.

[97] S. Johansson and M. Jahr. Grammatical tagging of the LOB corpus : Predicting word class from word endings. In S. Johansson, editor, *Computer Corpora in English Language Research*. Norwegian Computing Centre for the Humanities, 1982.

[98] R. M. Kaplan. A general syntactic processor. In R. Rustin, editor, *Natural Language Processing*. Prentice Hall, 1973.

[99] N. Katoh, N. Uratani, and T. Aizawa. Processing proper nouns in machine translation for English news. In *Proceedings of the Conference on 'Current Issues in Computational Linguistics', Penang, Malaysia*, 1991.

[100] N. Katoh, N. Uratani, and T. Aizawa. Proper noun processing for E-J machine translation. Technical report, NKH Science and Technical Research Laboratories, 1991.

[101] D. M. Keirsey. Word learning with hierarchy guided inference. In *Proceedings of the IJCAI*, 1981.

[102] R. Kittredge and J Lehrberger. *Sublanguage: Studies of Languages in Restricted Domains*. De Gruyter, 1982.

[103] S. F. Kripke. Naming and necessity. In D. Davidson and G. Harman, editors, *Semantics of Natural Language*. Reidel, 1972.

[104] R. J. Kuhns. A news analysis system. In *COLING*, 1988.

[105] R. J. Kuhns. News analysis: A natural language application to text processing. Presented at AAAI Spring Symposium on Text-Based Intelligent Systems, Stanford University, March 1990.

[106] W. Lehnert. Knowledge based natural language understanding. In *Exploring Artificial intelligence.* Morgan Kaufman, 1988.

[107] D. Lenat, R. Guha, et al. CYC: towards programs with common sense. *CACM*, August 1990.

[108] J. Levi. *The Syntax and Semantics of Complex Nominals.* Academic Press, 1979.

[109] E. Liddy and W. Paik. An intelligent semantic relation assigner: preliminary work. In D. Powers, editor, *IJCAI 91 Workshop on Natural Language Learning*, 1991.

[110] F. C. Liu and L. J. Haas. Synthetic speech technology for enhancement of voice-store-and-forward-systems. In *Proceedings of the American Voice Input/Output Society*, 1988.

[111] P. Ludewig. Incremental vocabulary extensions in text understanding systems. In O. Herzog and C. R. Rollinger, editors, *Text Understadning in LILOG.* Springer Verlag, 1992.

[112] S. Lytinen and A. Gershman. ATRANS : Automatic processing of money transfer messages. In *Proceedings of the AAAI*, 1986.

[113] S. Lytinen and S. Roberts. Lexical acquisition as a by-product of natural language processing. In *IJCAI 1989 Workshop on Lexical Acquisition*, 1989.

[114] M. Marcus. *A Theory of Syntactic Recognition for Natural Language.* MIT Press, 1980.

[115] A. S. Marmaridou. *What's so Proper about Names.* University of Athens, 1991.

[116] I. Marshall. Choice of grammatical word class without global syntactic analysis : Tagging words in the LOB corpus. *Computers and the Humanities*, 17, 1983.

[117] M. T. Maybury. Future directions in natural language processing: the Bolt Beranek and Newman language symposium. *AI magazine*, 1989.

[118] D. B. McDonald. Compound : a program that understands noun compounds. In *Proceedings of the IJCAI*, 1981.

[119] D. B. McDonald and F. Hayes Roth. Inferential searches of knowledge networks as an approach to extensible langauge understanding systems. In D. Waterman and F. Hayes-Roth, editors, *Pattern Directed Inference Systems.* Academic Press, 1978.

[120] D. D. McDonald. Robust partial parsing through incremental, multi-level processing: rationales and biases. In P. Jacobs, editor, *Text-Based Intelligent Systems: Current Research in Text Analysis, Information Extraction, and Retrieval*, 1990. Available as GE Tech Report 90CRD198.

[121] D. D. McDonald. An efficient chart-based algorithm for partial-parsing of unrestricted texts. In *The 3rd Conference on Applied NLP*, 1992.

[122] C. Mellish. *Computer Interpretation of Natural Language Descriptions.* Ellis Horwood, 1985.

[123] M. Meteer, R. Schwartz, and R. Weischedel. POST : Using probabilities in language processing. In *Proceedings of the IJCAI*, 1991.

[124] J. S. Mill. *A system of logic, ratiocinative and inductive.* Open Court, Illinois, 1988.

[125] L. A. Miller. The FactFinder system. Scientific Applications Inc Internal Document.

[126] R. Mitton. A partial dictionary of English in computer usable form. *Literary and Linguistic Computing*, 1, 1986.

[127] R. Mooney. Integrating learning of words and their underlying concepts. In *Proceedings of the 9th Annual Conference of the cognitive Science Society*, 1987.

[128] J. Nakamura and M Nagao. Extraction of semantic information from an ordinary English dictionary and its evaluation. In *COLING 88*, 1988.

[129] P. Norvig. Marker passing as a weak method for text inferencing. *Cognitive Science*, 13, 1989.

[130] J. Terry Nutter, E. Fox, and M. Evens. Building a lexicon from machine-readable dictionaries for improved information retrieval. *Literary and Linguistic Computing*, 5(2), 1990.

[131] B. Oshika and F. Machi. Computational techniques for improved name search. In *Proceedings of the 2nd Conference on Applied NLP*, 1988.

[132] F. Pereira and D. Warren. Definite clause grammars for language analysis — a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13, 1980.

[133] C. Pollard and I. Sag. *Information-Based Syntax and Semantics.* CLSI, 1987.

[134] D. Powers and C. Turk. *Machine Learning of Natural Language.* Springer Verlag, 1989.

[135] P. Proctor, editor. *Longmans Dictionary of Contemporary English.* Longman, 1978.

[136] R. Quirk, S. Greenbaum, G. Leech, and J. Svartik. *A Grammar of Contemporary English.* Longman, 1984.

[137] L. Rau. Extracting company names from text. Draft of 7th IEEE conference paper.

[138] L. Rau. Information retrieval from never ending stories. In *Proceedings of the AAAI*, 1987.

[139] L. Rau. Extracting company names from text. In *7th IEEE Conference on AI Applications*, 1991.

[140] L. Rau and P. Jacobs. Integrating top-down and bottom-up strategies in a text processing system. In *Proceedings of the ACL*, 1987.

[141] M. Rayner, A. Herguson, and G. Hagert. Using a logic grammar to learn a lexicon. Technical Report R88001, Swedish Institute of Compputer Science, 1988.

[142] U. Reimer. Automatic acquisition of knowledge from terminological text. In *Proceedings of the European Conference on AI*, 1990.

[143] E. Rosch, D. Gray, et al. Basic objects and natural categories. *Cognitive Psychology*, 8:382–439, 1976.

[144] B. Russell. *The Problems of Philosophy.* Oxford University Press, 1912.

[145] J. Sager, D. Dungworth, and P. McDonald. *English Special Languages: Principles and Practice.* Brandstetter, 1980.

[146] G. Sampson. How fully does a machine-usable dictionary cover English text. *Literary and Linguistic Computing*, 4(1), 1989.

[147] G. Sampson. Review of 'Computational Lexicography for Natural Language Processing', edited by Boguraev and Briscoe. *Computational Linguistics*, 16(2), 1990.

[148] R. Schank. Conceptual dependency: a theory of natural language understanding. *Cognitive Psychology*, 3, 1972.

[149] R. Schank and C. Rieger. Inference and the computer understanding of natural language. *Artificial Intelligence*, 5, 1974.

[150] J. R. Searle. Proper names. *Mind*, 67, 1958.

[151] P. F. Seitz and V. Gupta et al. A dictionary for a very large word recognition system. *Computer Speech and Language*, 4(2), 1990.

[152] M. Selfridge. A computer model of child language learning. *Artificial Intelligence*, 29, 1986.

[153] C. Sidner. Focusing in the comprehension of definite anaphora. In M. Brady and R. Berwick, editors, *Computational Models of Discourse.* MIT Press, 1983.

[154] S. Siegfried and J. Bernstein. Matching artists names. The Getty Art History Information Program, 401 Wilshire Blvd, Santa Monica, California.

[155] J. M. Siskind. Acquiring core meanings of words represented as Jackendoff-style conceptual structures, from correlated streams of linguistics and non linguistic input. In *Proceedings of the ACL*, 1990.

[156] J. M. Siskind. Dispelling myths about language bootstrapping. In D. Powers and L. Reeker, editors, *Proceedings of the Stanford Spring Symposium on Machine Learning of Natural Language and Ontology.* DFK, Kaiserlautern, FRG, 1991.

[157] J. M. Siskind. Naive phyiscs, event perception, lexical semantics and language acquisition. In D. Powers and L. Reeker, editors, *Proceedings of the Stanford Spring Symposium on Machine Learning of Natural Language and Ontology.* DFK, Kaiserlautern, FRG, 1991.

[158] B. Slator. Using context for sense preference. In U. Zerenik, editor, *Lexical Acquisition.* Lawrence Erlbaum, 1991.

[159] J. Slocum. Morphological processing in the Nabu processor. In *Proceedings of the 2nd Conference on Applied NLP*, 1988.

[160] K. Sparck Jones. So what about parsing compound nouns. In *Automatic Natural Language parsing.* Ellis Horwood, 1983.

[161] M. Spiegel. Pronouncing surnnames automatically. In *Proceedings of the American Voice Input/Output Society*, 1985.

[162] Europa Editorial Team, editor. *International Who's Who.* Europa Publications Ltd, 54th edition, 1990.

[163] Longmans Editorial Team, editor. *Who's Who in Science in Europe.* Longmans, fifth edition, 1987.

[164] M. Tomita. Linguistic sentences and real sentences. In *COLING 88*, 1988.

[165] J. Tsujii. Reasons why I do not care about grammar formalisms. In *COLING 88*, 1988.

[166] University of Chicago Press. *Chicago Manual of Style*, 13th edition, 1982.

[167] UW Centre for the New OED. *Using Corpora : The 7th Annual Conference of the UW Centre for the New OED.* The University of Waterloo Centre for the New Oxford English Dictionary and Text Research, October 1991.

[168] P. Velardi, M. Fasolo, and M. Pazienza. How to encode semantic knowledge: a method for meaning representation and computer-aided acquisition. *Computational Linguistics*, 17(2), 1991.

[169] P. Velardi, M. T. Pazienza, and S. Magrini. Acquisition of semantic patterns from a natural corpus of texts. *SIGART*, 1989.

[170] T. Vitale. An algorithm for high accuracy name pronounciation by parametric speech synthesizer. *Computational Linguistics*, 16, September 1991.

[171] A. Walker, M. McCord, J. Sowa, and W. Wilson. *Knowledge systems and Prolog - a Logical Approach to Expert Systems and Natural Language.* Addison Wesley, 1987.

[172] D. Walker and R. Amsler. The use of machine-readable dictionaries in sublanguage analysis. In Grishman and Kettridge, editors, *Analysing Language in Restricted Domains.* Lawrence Erlbaum, 1986.

[173] B. Webber. So what can we talk about now. In R. Berwick and M. Brady, editors, *Computational Models of Discourse.* MIT Press, 1983.

[174] R. Weischedel, A. Ayuso, et al. Partial parsing: a report in progress. In *DARPA Speech and Natural Language Workshop*, February 1991.

[175] P. Wheeler. Changes and improvements to the European Commission's Systran MT system 1979/84. In I. Kelly, editor, *Progress in Machine Translation.* Sigma, 1989.

[176] P. Whitelock, M. Woods, et al., editors. *Linguistic Theory and Computer Applications.* Academic Press, 1987.

[177] Y. Wilks. Consortium for lexical research information sheet. Available from CLR, Comp Res Lab, New Mexico University.

[178] W. A. Woods. Transition network grammars for natural language analysis. *Communications of the ACM*, 3(10), 1970.

[179] U. Zernik. Lexicon acquisition: learning from corpus by capitalising on lexical categories. In *Proceedings of the IJCAI*, 1989.

[180] U. Zernik. Lexical acquisition : where is the semantics. *Machine Translation*, 5(2), 1990.

[181] U. Zernik, editor. *Lexical Acquisition: exploiting on-line resources to build a lexicon.* LEA, 1991.

[182] U. Zernik and M. Dyer. The self-extending phrasal lexicon. *Computational linguistics*, 13, 1987.

[183] U. Zernik and Paul Jacobs. Acquiring lexical knowledge from text : a case study. In *Proceedings of the AAAI*, 1988.