An engineering approach to knowledge acquisition by the interactive analysis of dictionary definitions

Sanjay Poria

University of Durham

An Engineering Approach to Knowledge Acquisition by the Interactive Analysis of Dictionary Definitions

Sanjay Poria

Laboratory for Natural Language Engineering,

Department of Computer Science.

Submitted in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

Abstract

It has long been recognised that everyday dictionaries are a potential source of lexical and world knowledge of the type required by many Natural Language Processing (NLP) systems. This research presents a semi-automated approach to the extraction of rich semantic relationships from dictionary definitions. The definitions are taken from the recently published "*Cambridge International Dictionary of English*" (CIDE). The thesis illustrates how many of the innovative features of CIDE can be exploited during the knowledge acquisition process.

The approach introduced in this thesis uses the LOLITA NLP system to extract and represent semantic relationships, along with a human operator to resolve the different forms of ambiguity which exist within dictionary definitions. Such a strategy combines the strengths of both participants in the acquisition process: automated procedures provide consistency in the construction of complex and inter-related semantic relationships, while the human participant can use his or her knowledge to determine the correct interpretation of a definition.

This semi-automated strategy eliminates the weakness of many existing approaches because it guarantees feasibility and correctness: feasibility is ensured by exploiting LOLITA's existing NLP capabilities so that humans with minimal linguistic training can resolve the ambiguities within dictionary definitions; and correctness is ensured because incorrectly interpreted definitions can be manually eliminated.

The feasibility and correctness of the solution is supported by the results of an evaluation which is presented in detail in the thesis.

Declaration

The material contained within this thesis has not previously been submitted for a degree at the University of Durham or any other university. The research reported within this thesis has been conducted by the author unless indicated otherwise.

Acknowledgements

I would like to thank my supervisor Roberto Garigliano for his guidance throughout the course of this research. I can only wish for Roberto's broad intuition about language combined with his expertise in moving from high level ideas to low level implementation. I am grateful for all he has taught me during the last few years.

Past and present members of the LOLITA group have made my time in Durham extremely enjoyable. As well as becoming good friends, many of them have contributed to the production of this thesis in one way or the other. Thanks to David Nettleton for his work on the user-interface, Rick Morgan for discussing various technical details with me, and to Agnieszka, Kevin, Dominika, and Heather who proof read various drafts of this thesis.

This thesis is dedicated to my parents, particularly my mother who never had the opportunity to persue her academic interests when she was younger.

Contents

List of Figures

Chapter 1

Methodology

1.1 Introduction

Natural Language Processing (NLP) systems require different types of knowledge in order to perform non-trivial language comprehension tasks. Lexical knowledge is particularly important because it forms a core upon which levels of richer knowledge can be added.

It has long been recognised that dictionaries contain a wealth of lexical and world knowledge of the type required by NLP systems. Much of this knowledge exists in the definitions of words. There have been a number of attempts, using varying degrees of automation, to extract the semantic relationships from within dictionary definitions.

This thesis presents a semi-automated approach to the extraction of semantic relationships from within the definitions of words contained in the "*Cambridge International Dictionary of English*" (CIDE). To our knowledge it the first attempt which uses CIDE as the knowledge source. The strategy presented is to use LOLITA[1] Natural Language Processing (NLP) system in order to represent the complex se-

[1] Large-Scale Object-based Linguistic Interactor, Translator and Analyser

mantic relationships extracted from within the dictionary definitions, together with a human disambiguator who resolves the various types of ambiguities which arise in the interpretation process.

Initially we consider the end use for the semantic relationships to be extracted from the CIDE definitions. These issues are not considered by existing approaches. We show that dictionary definitions are not written so that they can be fully exploited by NLP systems. Consequently, the end use for the semantic relationships dictates the framework for the extraction process.

In addition, the semi-automated solution presented in this thesis balances feasibility and correctness which are ignored in existing approaches. A fully automated approach cannot filter erroneous semantic structures while a partially manual approach requiring linguistic experts cannot be feasible because of the sheer volume of analysis. We show how the existing NLP capabilities of LOLITA can be exploited so that the implicit and explicit semantic knowledge within CIDE definitions can be extracted in a feasible way. The extraction of implicit knowledge within definitions relies upon the novel layout of dictionary entries in CIDE.

1.2 Context of The Research

The research is to be conducted in the context of the parent project, the LOLITA system. The field of research is traditionally known as Natural Language Processing (NLP), a sub-branch of Artificial Intelligence (AI). Researchers in the field of NLP have come from many diverse backgrounds, ranging from AI and Cognitive Science through to Linguistics and Logic. Each field has diverse views on what the goals of NLP research should be and how these goals should be met.

In our opinion, the term 'Natural Language Processing' is used to cover a much wider enterprise (such as the investigation of linguistic theories by computational means) than the approach taken in the development of LOLITA. We believe the new field of Natural Language Engineering (NLE is a sub-field of NLP) more accurately reflects the LOLITA methodology and tradition.

The remainder of this chapter discusses the methodological view (and criteria which result from it) that Natural Language Engineering takes. Without these criteria it would be difficult to judge the results of the research, as they constitute a yard stick with which to measure and compare the outcome of the project. It shall be shown that our criteria have been selected after due consideration and according to methodologically sound principles.

1.3 Why the LOLITA System?

The LOLITA system is a natural language system which is currently under development in the Laboratory for Natural Language Engineering, at the University of Durham. It is a large scale system designed around a core of natural language capabilities (a detailed description follows in Chapter 4).

The key features of the system are, on the one hand, its large semantic network, which can store all kinds of knowledge and support various forms of reasoning, and on the other, its approach to the analysis of natural language. The system attempts a full, 'deep' analysis of the input (including a full parse) and aims to produce a semantic representation of the text.

There are two important reasons why the large-scale acquisition of lexical knowledge is to be performed in the context of the LOLITA system.

Firstly, there is a need for rich knowledge in LOLITA because of the diverse range of language processing tasks that need to be carried out to achieve a deep analysis of input text, and secondly, the task of knowledge acquisition requires the type of deep analysis which LOLITA is able to perform.

The inherent circularity in the knowledge acquisition process which is apparent in the paragraph above is discussed in Section 3.6.

1.4 Natural Language Engineering

Traditional approaches to NLP, whether originating from a cognitive, linguistic or AI point of view, have tried to formulate either universal theories that cover all aspects of language or to develop very restricted or detailed (often logical) theories that model small areas. The utilisation or expansion of these ideas to produce realistic systems which are not highly restricted by their task or domain has proved highly problematic, a fact often disguised by, or hidden in complex logical formalisations of intuitively simple ideas; Wilks [Wilks, 1996] writes:

> *"Theoretical issues remain very important, but there is growing skepticism about the importance of small-scale, research systems and whether many of them are genuinely original as opposed to being notational variants in a field not very aware of its own history."*

The often repeated view amongst computational linguists that the movement from core ideas to a working NLP system should be just a matter of software engineering seems, on this observation, to be unfounded.

The current inadequacies of NLP to produce working systems has led to the creation of the new field of Natural Language Engineering (NLE). NLE is a field which applies the ideas and practices of other engineering disciplines to the field of NLP. [Boguraev *et al.*, 1995] states:

> *"The principle, defining characteristic of NLE work is its objective: to engineer products which deal with natural language and which satisfy the constraints in which they have to operate."*

In other words it is a pragmatic view of current NLP. NLE attempts to direct contemporary work towards the medium-term production of useful NL tools.

The belief adopted is that there is a set of critical engineering criteria which should be applied to the field of NLP so as to utilise existing technology to produce useful systems. It is the hope that if these engineering principles are adhered to, new technology which becomes available can be incorporated into a NL engineered system. The formalisation of these principles reflect a sort of *pragmatic but principled* view which sits at the neat fringe of scruffy AI[2].

The following sections list important NLE criteria (some detailed in [Smith, 1996]) providing examples of how the success of each criterion may be judged.

1.4.1 Scale

The scale of NLP systems has only recently become an important issue to many AI researchers. Only after decades of research has it been accepted that the expansion of 'toy-systems' into their large-scale counterparts, capable of processing real-life, free text (which is the eventual goal of NLP research) poses research problems of its own.

Programs written by AI researchers often process only a few sentences (a fact that is often hidden in their literature). A well known account is documented in [Guthrie *et al.*, 1996]:

> *"In a moment of great honesty five years ago, a group of AI researchers*

[2] Kautz [Kautz, 1987] writes:

> *"Workers in Artificial Intelligence are often divided into the neat and scruffy camps, with the neats trying to create formal theories which systematise the heuristics uncovered by the intuition driven scruffies.*
>
> *In practice, the two camps often degenerate into unrealizable logicism or unprincipled (and unreproducable) hackery."*

> *of natural language processing (NLP) admitted in public (in an answer to a question by Bran Boguraev) how many words there really were in the vocabularies of their systems. Of the answers, the average was 36, a figure often taken to be a misprint when it appears, though it was all too true."*

and it is many examples of such toy systems which has led to the explicit consideration of scale as an important indicator of the utility of NLP systems in the real world.

In general, the size of the grammar, the number of entries in the lexicon and the amount and depth of semantic knowledge all provide good indicators of the scale of an NL system.

1.4.2 Robustness

Robustness in NLE concerns not only the linguistic scope of the system, but also the acceptability of results when input falls outside this scope. The recognition that robustness is a serious problem which must be faced up to in general has been the prime motivation in the development of the Cyc [Lenat *et al.*, 1986, Guha and Lenat, 1990] knowledge base. The Cyc project takes the view that the robustness bottleneck in AI systems is caused by their lack of world knowledge.

At the very least, a system should not crash when it receives input which is outside its scope; it should carry on and try its best to cope with the conditions it is working under.

1.4.3 Maintainability

Maintainability is a measure of how useful the system is over a long period of time. As in any software engineering project, the system should allow its configuration to be easily altered.

A system which has successfully evolved over a long period of time with a high turnaround of researchers indicates good maintainability. To be successful, it must be possible for both the original developer and other programmers to understand the system so they can perform maintenence in a reasonable time.

1.4.4 Flexibility

Flexibility is the ability to modify the system for different tasks in different domains. An indication of the flexibility of a system is given by the amount of time spent on development of a particular domain compared to work on the core of the system. For a highly flexible system this proportion of task specific development will be low.

1.4.5 Integration

Integration concerns the ease with which components may be added to an existing system, whether at the present time or in the future. Specifically there are two aspects of integration which should be considered when designing a component to be integrated into an NLE system.

- System components should not make unreasonable assumptions about the function of other components which may not presently exist. Likewise components should not implement aspects which clearly belong to other modules.

  In the former case, such assumptions are often made when specific NLP problems are tackled in isolation since there is greater opportunity to simply assume that certain functions (which often turn out to be the most complex) will be carried out by other components which do not yet exist. For example, any desired inference can be made if a particular rule is assumed to be available in the knowledge base.

- Components should be designed and built to actively assist other components. This should be the case even if the other components do not yet exist. Again a common example is the building of knowledge in a particular form which can only be utilised by the inference algorithm currently being implemented *e.g.* deduction. The knowledge may be in a form which is too restrictive to be usable by other types of inference *e.g.* analogy, induction *etc.*

The ability of a system to be used as a prototype for many diverse applications is a good indication that it is well integrated. One possible measurement could be the proportion of code dedicated to a specific application compared to the core code.

1.4.6 Feasibility

This concerns ensuring that constraints on the running of the system are acceptable. For example hardware requirements (execution speed) should not be assumed to be too great. It incorporates making the system efficient.

Some areas of Computer Science and AI use complexity analysis as a measure of the feasibility of algorithms. However this is not always paramount in NLE since it is often the case that in reality there is some upper bound on the amount of data that is processed, *i.e.* the worst case scenario may occur with such infrequency that a theoretically complex algorithm may be justified.

1.4.7 Usability

Systems produced using NLE techniques should support the functions that end users require, *i.e.* the system must satisfy a need [Boguraev *et al.*, 1995]. Ultimately it is satisfaction of the users with the product that provides a measure of usability.

In a research environment the use of simulation experiments with potential end users is important to show this aspect has not been ignored.

1.4.8 Techniques Used

Often there will be no universal theory which can be taken "off the shelf" and utilised to solve some task. Take for example, the well known and widely studied task of anaphora resolution. After decades of research in many fields there exists no generally accepted theory of how one should tackle it.

In such cases where no universal theory exists alternative approaches will range from localised theories able to cover only partial cases, to purely heuristic approaches and finally to adaptive or evolutionary techniques. The particular mixture of strategies will often depend upon the evolution of the system and the current state of the art.

1.4.9 Cost-Benefit Analysis

Often the best theoretical solution is not the best practical one. There may exist a trade-off between the depth and breadth of the solution to some problem. If a simple algorithm has only slightly worse case coverage than a complex one then it may be better to use the former. Cost-benefit analysis involves reaching a balance between two or more aspects of NLE, *e.g.* a simple algorithm may not have the same *robustness* as a more complex one but may lead to a more *feasible* system.

Often the cost of this sort of analysis may outweigh the benefits. Despite this, informal investigations of alternatives to various aspects of the system during its development are useful and should be undertaken.

1.5 Summary

The subject of this research is traditionally thought of as belonging to the field of NLP. However we believe that NLP is a term used to cover a much wider enterprise than the approach taken in the development of LOLITA. The new field of NLE which applies engineering techniques to NLP in an attempt to produce large-scale usable NL systems best describes our methodology.

The range of engineering criteria which are to be applied to NLE research were listed. They provide us with a yardstick by which the success of the research should be judged and hence form an integral part of the design process.

It is hoped that the pragmatic emphasis of NLE can help to address the bottleneck which causes the disparity between the large amount of theoretical work done in the area and the relatively small number of realistic working systems.

Chapter 2

The Problem Area and Project Aims

2.1 Introduction

In this chapter we discuss in detail the specific problem which forms the subject of this research, how this problem is broken down into its constituent parts and the criteria with which the success of the project is to be measured. These criteria are determined with respect to the methodological criteria discussed in Chapter 1 and further domain specific criteria which are determined by the problem being tackled.

2.2 The Problem Area

Many of the early NLP systems developed algorithms which were tested on lexicons of no more than a few carefully selected words arranged into a neat inheritance hierarchy. The demand for systems which could process (*e.g.* by translating, summarising, *etc.*) unrestricted text has led to the realisation that, firstly, the knowledge base is a key component of many NLP systems, and secondly, that the construction of such a knowledge base poses many research problems.

There is general agreement within the NLP community that various levels of knowledge are required to solve complex NLP tasks. Lexical knowledge contains information about the meanings of individual words known to the system. The importance of lexical information can be illustrated by considering a core NLP task[1] such as pronoun resolution in the examples:

2(a) *John gave Michael the invoice. He ripped it up into little pieces.*

(b) *John took the invoice from Michael. He ripped it up into little pieces.*

These sentences are both structurally similar but have different antecedents for the pronoun '*he*'– *Michael* for the first case and *John* for the latter case. The knowledge required to solve this relies upon utilising the meanings of the verbs '*give*' and '*take*' respectively.

The long term aim of the parent project is to build a rich knowledge base which can be used for a wide variety of NLP applications. Lexical knowledge is particularly important because it forms a minimal base to which richer knowledge (*e.g.* one based on the normal course of events in the world) can be added.

It has long been recognised that ordinary dictionaries are a potentially rich source of lexical information of the type required by NLP systems. However extraction of this knowledge in a form which can be used by an NLP system has proved more difficult than first anticipated.

The aim of this thesis is to investigate how the rich knowledge available in dictionary definitions (*i.e.* the knowledge which is beyond the basic grammatical categorisation of words) can be exploited by NLP systems.

[1] a core NLP task is considered to be one which is essential in a wide range of non trivial NLP applications.

In addition, the aim is to implement a feasible knowledge acquisition procedure for the extraction of semantic relationships from dictionary definitions. To ensure the success of the project, the design and implementation of the system should follow the NLE methodology whose major principles were listed in Chapter 1.

2.3 Criteria for Success

The particular criteria which indicate the success of the research are discussed below. They are either derived by the application of the NLE methodological criteria (discussed in Chapter 1) to the domain of knowledge acquisition, or are more independently derived criteria from the problem domain.

To show the motivation for these criteria, it is helpful, in our view, to list the different ways in which many research projects in the field of NLP fail to meet the criteria of NL engineered systems.

2.3.1 Scale

There are many problems which may not manifest themselves when a small amount of knowledge is acquired and are consequently often ignored. However, one of the explicit criteria of engineered systems is to consider the problems which often prohibit the scaling up of solutions to the full domain for which they are intended.

The criterion is addressed by considering those problems which may prohibit the scaling up of knowledge acquisition procedures:

- although early research in NLP (*e.g.* text understanding work in the 60's and 70's) had seemed promising, few large scale applications existed. One of the reasons for this was that the systems were often tested and evaluated on unrepresentative, hand selected examples. It is of critical importance that ideas are tested on the range and depth of input that can be expected in the real case.

- often the effect of badly acquired knowledge (meaning knowledge which is interpreted or represented wrongly) can have a snowball effect on subsequently acquired knowledge. This may not be apparent when only a few isolated cases are considered.
- when a lot of knowledge needs to be acquired the only feasible way may be to divide the acquisition problem. If that is the case then knowledge integration becomes an important issue which may bring as many new problems as it solves.

2.3.2 Integration

Modules should be designed and integrated into a system to assist other components whether they currently exist or may exist in the future.

In terms of knowledge acquisition integration means that the representation of lexical knowledge should be as compatible with the current knowledge representation language of LOLITA as possible. Consequently, existing components (*e.g.* inference procedures) do not have to be modified to take advantage of the newly acquired knowledge.

This constraint is often overlooked in small research projects where only one procedure is tested in isolation. In such cases, the representation of knowledge is less important because the module is not integrated into a fully operational system.

2.3.3 Correctness

The semantic correctness of knowledge acquired for use in NLP systems is more important than the acquisition process for other domains for two reasons:

1. the arrangement of knowledge into hierarchies is an integral part of most NLP systems. The hierarchical representation is desirable because it not only provides spatial efficiency but also allows inferences to be made in an efficient way, *i.e.* by inheritance.

   Consequently, the correctness of knowledge in such hierarchies is doubly important because inaccurate knowledge is inherited across multiple levels.

2. the aim of this research is to build a lexicon for a domain independent large-scale NLP system. The knowledge base will form the core of any language processing applications that are subsequently built. The accuracy required of such applications is unknown at present. Given the chance that certain applications may require extremely high levels of precision, bad knowledge may jeopardise the potential to build any such applications.

This means that it is important to detect badly written dictionary definitions together with errors resulting from the acquisition process.

2.3.4 Feasibility

Feasibility of the acquisition process is often ignored in much of the research in the field. Consider a situation where the acquisition of each item in the lexicon requires human intervention to make certain decisions (much of the early work in knowledge acquisition for NLP was of this type). Clearly it is unreasonable to expect skilled people (*e.g.* researchers, linguists, *etc.*) to carry out the task of acquiring the knowledge contained in an entire dictionary. The cost would be far too great. Therefore it is important to give consideration to the feasibility of the long term aim of a project.

2.3.5 Testing

The testing of many NLP projects, which are built in isolation, is done by simulating one or more components of a large scale system. If the various components can be separated (which is in some cases far from obvious), there is a tendency to assume the existence of components whose functionality is past the current state of the art.

This is dangerous because the real complexity of the overall task may be hidden in the component which is assumed to exist. In reality the existence of this component may have unforseen implications for the other parts of the system which have been designed. To avoid this trap it is important that unreasonable assumptions are not made concerning the functionality of other components of an NLP system.

2.4 Summary

The aim of the research introduced in this chapter involves building a lexicon by utilising the rich knowledge of word meanings contained in ordinary dictionaries. This lexicon is to form the core of the knowledge possessed by the LOLITA NLE system.

Various criteria relevant to the task have been discussed. They are important considerations by which the success of the project can be judged. The criteria reflect the engineering paradigm which is inherent in the development of the LOLITA system.

Chapter 3

Review of Literature

3.1 Introduction

Since researchers in NLP have tried to scale up their solutions to language understanding problems, which, before then, operated on knowledge bases of only a few hundred nodes, there has been increased acceptance that the knowledge acquisition bottleneck is the key obstacle to the development of robust NLP systems.

Published dictionaries have long been seen as potential sources of knowledge that could alleviate the knowledge acquisition bottleneck, not only because they can provide lexical and real world knowledge of the type that NLP systems need, but they are ideally structured for taxonomic organisation. Until recently it was mainly grammatical knowledge of words that was extracted from these dictionaries. The largely implicit but potentially richer knowledge present in the definitions of words has proved more challenging for automated analysis.

The first part of this chapter reviews the work in *computational lexicography* which forms the subject of this research: the extraction of semantic knowledge from the definitions of words contained within dictionaries. The final section illustrates the LOLITA philosophy by comparing and contrasting two different knowledge-based projects in AI: WordNet and Cyc.

3.2 Knowledge for NLP

It is widely recognised that in order for NLP systems to perform non-trivial tasks involving the comprehension, production and acquisition of both written and spoken media, they require various types of knowledge[1]. Two of the most important types are[2]:

Lexical knowledge — is information about the meaning of individual words, the sum of which is often called a "*lexicon*". Although there is no consensus as to the precise content of a lexical entry, there is little doubt it will contain some aspect of the following kinds of knowledge:

1. *Grammatical lexical knowledge* which includes information about the grammatical construction of the word *e.g.* phonological, morphological information and syntactic knowledge.
2. *Semantic lexical knowledge* concerns the meaning of the individual words and how they are combined, *e.g.* the verb "*sell*" involves a transfer of possession of an object from the *seller* to the *buyer*, who are indicated by the subject and object of the verb respectively.

World knowledge — is the type of knowledge that goes beyond the meaning of individual words which is required when common NLP tasks (*e.g.* pronoun resolution, attachment problems, plan recognition) are undertaken. Examples typically include: causal knowledge, knowledge of the normal course of events in the real world, and knowledge about the properties of objects.

[1] see [Cater, 1995] for an in-depth discussion of the knowledge requirements of NLP systems.

[2] Other kinds of knowledge include frequency information, topic *etc.*

Although the description above presents a division between knowledge contained in the lexicon and world knowledge, in reality no such clear distinction exists [Boguraev and Briscoe, 1989b][Guthrie *et al.*, 1996]. This is reflected by the fact that in many NLP systems (*e.g.* LOLITA), both types of knowledge are encoded in a uniform representation (*e.g.* a semantic network, first-order logic), with no real distinction made between them.

3.3 Computational Lexicography

Knowledge of the structure and meaning of words (that traditionally thought of as belonging to the lexicon), is viewed as a key component of an NLP system. The emerging field of *computational lexicography* aims to tackle the knowledge acquisition bottleneck by transforming the knowledge in existing lexical resources into a form which can be utilised by NLP systems.

The pioneering work in computational lexicography (by Amsler and White, see Section 3.4) in the early 80's aimed at creating a semantic network from the definition of headwords (the words being defined) in a dictionary.

Since the pioneering work of Amsler and White, the goal of computational lexicography has moved far beyond the construction of semantic networks, which utilise only a tiny fragment of the information provided in dictionaries. Research in the last decade has concentrated upon developing techniques to transform all the lexical knowledge in machine readable dictionaries (henceforth MRDs) into lexical knowledge bases which can subsequently be utilised by different NLP systems and applications. The idea is the creation of neutral knowledge "repositories" whose contents are easily exploited by transformation into the representation required by different NLP systems.

Some of the benefits of using dictionaries as a lexical resource are noted below:

1. Dictionaries provide *diverse* knowledge of word senses which is required by different components of NLP systems. Examples of this knowledge include:

    - headword information including spelling, hyphenation and phonology of each entry
    - morphological knowledge, *e.g.* information about the past tense and past participle of verbs
    - semantic lexical knowledge describing the distributional behaviour of the entry, *e.g.* its grammatical class, preferred type of subjects and objects of verbs where appropriate, *etc.*
    - meaning definitions of each entry augmented by a series of examples of its usage and cross-reference information

2. Each defining word appears in the headword list that forms the dictionary itself and consequently it is *ideally structured for taxonomic organisation* [Guthrie *et al.*, 1996].

    Although there is no consensus in NLP on the types of structure that are best suited to capturing the meaning of lexical entries, more often than not the knowledge representation language (KRL) of NLP systems is based upon the notion of concepts organised in an inheritance hierarchy along a generalisation/specialisation axis. The utility, for NLP systems, of hierarchically structured networks of concepts has been demonstrated beyond doubt [Boguraev and Briscoe, 1989b].

    Therefore there is good reason to believe that the taxonomic structure of knowledge contained within dictionaries is ideally suited for NLP purposes.

Although the potential utility of dictionaries as a source for the construction of lexical knowledge bases is undisputed, progress during the last decade of research in computational lexicography has been slow.

An examination (by [Boguraev and Briscoe, 1989a]) of the cases where information from MRDs has been extracted and subsequently utilised by NLP systems shows that it is primarily the lexical knowledge of words that has been of interest, *e.g.* spelling, hyphenation, morphology, *etc.* This is not surprising since the lexical information contained within entries is relatively simple to process automatically; the fields containing this information are typically structured objects with a limited number of values and whose contents are clearly delimited, [Boguraev and Briscoe, 1989a] write:

> "The use of form and function information is an obvious place to start in the computational exploitation of MRDs both because this information is represented moderately formally in most MRDs and because it seems plausible *a priori* that NLP systems can treat this information as straightforward 'data' in a fashion which would not be possible for, say, sense definitions."

Work which analyses the meaning component of a dictionary entry (which contains more real world knowledge) useful for many NLP applications has been slow (see below), not only because of the lack of structure in the defining language, but also the lack of consensus amongst NLP researchers in the representation language that should be used to encode the resulting analysis.

The bulk of this review provides an up to date account of the research which aims to tackle the complex problem of extracting semantic knowledge from the definitions of headwords in MRDs. There are two categories of important issues: local issues concerning the structure of individual definitions, and more global issues which arise when processing entire dictionaries of individual definitions. Both types are discussed below.

3.4 The Structure of Dictionary Definitions

The first (and pioneering) work in *computational lexicography* during the late 70's and early 80's which attempted to extract semantic content from dictionary definitions was done by Amsler and White (see [Wilks, 1996] for a review of their work) using the *Merriam-Webster Pocket Dictionary.*

They constructed a semantic network of noun senses by using human disambiguators to sense-tag the head of the first NP in each definition together with any others which "made a significant semantic contribution to an IS-A[3] link" [Amsler and White, 1979]. For example, in the definition of the noun:

deuterium — a form of hydrogen that is twice the mass of ordinary hydrogen

the resulting network would contain links from the headword '*deuterium*' to the nouns '*form*' and '*hydrogen*' with the latter being marked in a special way to indicate that it is not the head of the first NP [Wilks, 1996].

The network was subsequently (automatically) built by assuming that the head of the first NP (noun phrase) of the definition was related to the headword by an IS-A relation, while the rest of the marked nouns were simply labelled as contributing semantic content to the headword. Unfortunately the resulting network was never made publicly available for further research.

The fact that definitions are written in such a way, that within each definition, a term that is a generalisation of the headword is often present, had been recognised long before the work of Amsler and White. This term is named the *genus* and is often related to the headword via an IS-A relation.

The following definition (see [Guthrie *et al.*, 1996][Wilks, 1996]) taken from the *Longman Dictionary of Contemporary English* (LDOCE) illustrates this structure:

knife — a blade fixed in a handle, used for cutting as a tool or weapon.

in which "*knife*" is the headword of the definition, "*blade*" is the **genus** term and

[3] the IS-A relationship is also referred to as subordination/superordination, subset/superset, or hyponym/hypernym.

the phrase "*fixed in a handle, used for cutting as a tool or weapon*" is called the **differentia**. A taxonomy can be built using these definitions because an IS-A relationship is established between the headword and the genus term, *i.e.* "*knife* IS-A *blade*". The differentia shows how a headword with the same genus term differs from the one being defined.

Arguably, all the research which aims at the semantic analysis of definitions is rooted in the work of Amsler and White. Subsequent work in *computational lexicography*[4] which has had the aim of adding further degrees of automation to the process of constructing genus networks have encountered two main problems:

1. Fully automated approaches must disambiguate the sense of the genus term from other senses of the same word. In the example above, there are other senses of the word "*blade*" (noted by [Wilks, 1996])[5]:

   blade$_1$ — *a gay sharp amusing fellow*
   blade$_2$ — *a flat cutting part of a tool or weapon*

   and so it is important that "*knife* IS-A *blade*$_2$" is identified and not the erroneous structure, "*knife* IS-A *blade*$_1$ IS-A *fellow*".

2. The strategy of taking the head of the first noun phrase of a definition as the IS-A related genus[6] term (in LDOCE) is successful approximately 90% of the time [Wilks *et al.*, 1996], the syntactic structure of the definition being [Nakamura and Nagao, 1988]:

   {determiner} {adjective}* `Genus Noun` {adjective phrase}*

   There are many examples (*e.g.* see [Nakamura and Nagao, 1988] and [Wilks *et al.*, 1996]) however, where the syntactic and semantic heads of definitions do not coincide, *e.g.*,

[4] A term coined by Amsler and White.

[5] different word senses of a word are indicated by the differing subscripts.

[6] following convention, we often use "genus" to refer to the IS-A related generalisation of the headword where we should more accurately give a name for the relationship.

abbey — the group of people living in such a place

academic — a member of a college or university

cyclamate — any of various manmade sweeteners

These examples define the headword using different semantic relations than the normal hyponym one. Nakamura and Nagao use the term "`function noun`" to refer to the word that expresses the particular semantic relationship between the headword and what they call "`key noun`" (*i.e.* the target of the semantic relation). In the examples above the function nouns are *group*, *member* and *any* respectively. They note that the syntactic form of a large class of these non-standard definitions is:

{det.} {adj.}* ⟨`Function Noun`⟩ of ⟨`Key Noun`⟩ {adj. phrase}*

In addition they list 41 functional nouns (from LDOCE) which are subsequently used to identify the many different semantic relations encoded within definitions.

There exist many approaches (*e.g.* [Nakamura and Nagao, 1988][Vossen, 1990]) to the characterisation and representation of the semantic relationships identified in the non-standard definitions of nouns. [Wilks *et al.*, 1996] provides an excellent history of research in the automated construction of genus hierarchies.

In summary, the techniques introduced above aim to construct genus hierarchies (possibly augmented with other semantic relations), by processing the first NP of a definition to identify, within it, a term which is more general than the headword being defined.

The most recent large-scale attempt at exploiting existing lexical resources is the EC funded ACQUILEX[7] project which was setup to explore the utility of constructing a multilingual lexical knowledge base from machine-readable versions of

[7] Acquisition of lexical knowledge for Natural Language Processing.

conventional dictionaries[8]. Part of the project involved the design of LKB, a lexical knowledge base system which allows the representation of syntactic and semantic information semi-automatically extracted from MRDs.

The project takes the view that previous approaches to the construction of lexicons for NLP systems from MRDs produce networks which are not directly utilisable as NLP lexicons because they claim that these projects do not use a formally specified representation language [Copestake *et al.*, 1993]. Therefore, LKB's knowledge representation languague (LRL) is a formally specified language which can be viewed as an augmentation of a type graph-based unification formalism with minimal default inheritance [Copestake, 1993b, Copestake *et al.*, 1993, Copestake, 1993a]. A complete formal description of LRL is given in [Copesteak, 1993].

Copestake [Copestake, 1990, Copestake, 1993b, Copestake *et al.*, 1993] takes the view that the relationship between headword and genus term is best represented by default inheritance rather than a strict ISA relationship. She gives the following LDOCE definitions to illustrate the point:

dictionary — a book that gives a list of words in alphabetical order with their pronunciations and meanings

lexicon — a dictionary esp. of an ancient language

Although dictionary is a book, the purpose of a dictionary is to be referred to rather than read. However, the purpose of a lexicon is the same as that of a dictionary. Consequently, the properties of the concept of book are potentially defeasible at deeper levels in the taxonomy. LRL provides a formal language in which to represents this kind of default inheritance.

Although issues of representation are seen to be extremeley important within the ACQUILEX project, only genus taxonomies are constructed. Many other issues of

[8] see `http://www.cl.cam.ac.uk/Research/NL/acquilex/acqhome.html` for various details about the ACQUILEX project.

representation may arise if the semantic content of differentia were extracted and represented in the knowledge base. In the example above, presumably the overiding of the default purpose from book to dictionary depends upon the particular notion of the word '*read*'. Due to the vagueness of language it is unlikely that it will be defined to such a fine level of granularity. In addition, it is unclear how knowledge about the purpose of objects such as books and dictionaries can be transformed such that the purpose of the dictionaries is recognised to be a specialised version of the purpose of books. Presumably an object can have many purposes. In conclusion, although the ACQUILEX project stresses the utility of a formal lexical framework, it is unclear whether this level of formailty can actually be explioted by an NLP system.

Research on the processing of complete definitions (beyond the first NP) has been slow partially due to the inherent problems in dealing with unstructured text. Below we review some of the main work on this topic.

3.4.1 Analysis Beyond Genus Extraction

The potential utility, for NLP systems, of knowledge contained beyond the initial NP (or VP) of a definition (*i.e.* in the modifier or differentia) has been demonstrated by some researchers, *e.g.* Slator and Wilks [Slator and Wilks, 1990] show how it can be used to identify the semantic relationships intended by prepositions, in text being analysed by an NLP system.

However, the techniques required in the identification of elements within the differentia, and the semantic relations between them, are considerably more complex than in the construction of genus hierarchies; whereas the main techniques for identifying genus terms has been an exhaustive structural analysis of the initial NP (or VP) of definitions, no such technique is possible here. In the worst case it is arbitrary NL fragments which need to be interpreted. Consequently, the generality of the problem results in techniques (presented below) which subsume the genus extraction process.

3.4.1.1 Alshawi

Alshawi [Alshawi, 1989][Alshawi, 1987] notes that a common problem with experimental NLP systems is the need to process unknown words which can be the result of using an incomplete lexicon. He notes that "missing vocabulary" is the most frequent cause of errors for the FRUMP system [DeJong, 1979], a system designed to achieve a high degree of robustness.

A potential remedy to this problem is to appeal to the appropriate word sense definition of the unknown word in a MRD to acquire the necessary semantic knowledge which will enable further processing. He has implemented an automated mechanism which processes the definitions of nouns, verbs, adjectives and adverbs contained in LDOCE. The output of the processing are structures of the form shown in Figure 3.1.

Alshawi assumes that there exists some system which uses the types of structures produced by his algorithm. This system is assumed to have a domain model within which to interpret the structures produced [Alshawi, 1989], the idea being that gaps in the lexicon of the end system can be filled by appealing to and interpreting the structures produced by the analysis of the unknown word.

The advantage of using LDOCE is that it defines words using a restricted subset of English words. This set of words is called a defining vocabulary. The list contains 2000 words which are used in their central "well understood" senses. Consequently the target system needs to know the meaning of the 2000 words in order to process and interpret subsequent structures produced by Alshawi's algorithm.

The structures are built by matching the definition text against a hierarchy of pattern matching rules. Each rule has a number of children each of which are more specific than their parents. General rules are matched against the input text,

```
mug (noun) --- a foolish person who is easily deceived

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


club (verb) --- to beat or strike with a heavy stick

  ((CLASS STRIKE) (OTHER-CLASSES ((BEAT)))
    (ADVERBIAL
     ((CASE WITH)
      (FILLER (CLASS STICK) (PROPERTIES (HEAVY))))))


bushy (adj) --- (of hair) growing thickly

  ((CLASS PROPERTY)
    (PREDICATION (CLASS GROW) (MANNER THICKLY))
    (RESTRICTED-TO ((CLASS HAIR))))


overland (adv) --- across or by land and not by sea or air

  ((MANNER
    (CASE ACROSS) (FILLER ((CLASS LAND)))))
```

Figure 3.1: Examples of structures produced by Alshawi's phrasal matching rules.

and in the case of a successful match, its more specific children are tried on each iteration until the matching process fails. Example of the type (and structure) of the matching rules are:

```
(n-100
  (n  &&  +0det  &&  &0adj  &noun  &&)
      n-110  n-120  n-130  n-140)

(n-110
  (n  +0det  +0intens  &0adj  &noun  *0pp-mod  &&))
```

which show two matching rules (n-100 and n-110). The second line in each case shows the pattern to be matched and the third (in the former case only) shows the direct descendants of the rules. The lone 'n' in the pattern indicates the lexical category to which the rule applies (nouns in this case). The matching of individual elements in patterns is indicated below:

```
&& an arbitrary segment of input words
+0det zero or one determiner
&noun one or more nouns
&0adj zero or more adjectives
*0pp-mod zero or one prepositional phrase modifier
```

It is clear that the definitions of nouns which match the general rule n-100 are a superset of those that match the more specialised n-110. The rules are matched top-down, only trying the latter, if the former match succeeds. Once the most specific match is determined, structure building rules associated with each of the matching rules build the output structures shown in Figure 3.1.

An appealing aspect of using these pattern matching rules is the ease with which *known* structure present in the definitions of LDOCE can be exploited. The ability to ignore the more complex (or unstructured) fragments of input (*e.g.* by use of &&)

while at the same time the ability to concentrate on particular structures present in the definition seem to be key to such an approach. Consequently, the rules are most successful at finding the genus terms [Alshawi, 1989] of a definition (those appearing under a `CLASS` label in Figure 3.1).

Although the approach that Alshawi takes is an appealing one there are many aspects which have been largely ignored. Firstly, the problem of disambiguating words senses is not tackled. For example, consider the definition of the verb `club` given in Figure 3.1; not only is the sense of the verb `strike` unknown (possibilities include: `strike-stop-work`, or `strike-think-of-idea`, *etc.*), but the meaning of the preposition `with` is not made more specific than a `CASE` label, when it might mean, more specifically, *to be in a location*, *to use an instrument* and so on.

In addition, competing semantic structures, which can result from processing a definition where there is more than one successful analysis rule, are dealt with in a simplistic way. To quote Alshawi [Alshawi, 1989], "one such analysis is chosen by an over-simplistic heuristic that basically prefers analyses accounting for more words of the input definition". It seems that competing analyses which could result from common language understanding tasks such as prepositional phrase attachment are not tackled by Alshawi's matching approach.

In conclusion, Alshawi's approach is appealing because pattern matching rules seem an intuitively simple way of exploiting the known structure present in dictionary definitions. However there are two major drawbacks of Alshawi's work. Firstly, he assumes the target system will possess a strong domain model in which it will carry out non-trivial tasks such as sense disambiguation, and secondly, although Alshawi only aims at extracting partial information from definitions, it is unlikely that the pattern matching approach will generalise to extract knowledge in more unstructured text present in many definitions.

3.4.1.2 Slator and Wilks

A popular approach in NLP is to use a precomputed lexicon containing knowledge in a form which can be used directly by the NLP system, if and when required. Slator and Wilks [Wilks, 1996] take a different approach. They construct a "lexicon provider" which is an NLP subsystem to provide text-specific lexicons from selected MRD definitions. The input to the subsystem is unconstrained text and the output is a collection of lexical semantic objects, one for every sense of every word in the text. Hence the idea is that that the lexical semantic objects are provided on the fly, during the analysis of input text. For example, during the processing of the input sentence

"The technician measures alternating current with an ammeter."

the lexicon provider will be required to produce a number of semantic frames corresponding to the words in the input: 'alternate' (three adjective senses, one verb sense), 'measure' (two nouns, two verb senses), 'ammeter' (one noun sense), *etc.*

The semantic frames contain sub-categorisation information, semantic selection codes and contextual knowledge. The text of selected dictionary definitions is analysed, to enrich the frame representation because, as [Wilks, 1996] (p. 154) comments, "there is a hidden wealth of further information implicit within the text of definitions themselves, namely, in the genus and differentia". They use a chart parser which produces phrase-structure trees to parse the definitions of LDOCE, claiming a success rate beyond 90%. The parses are constructed to be as flat as possible because, with certain minor exceptions, no procedure associates constituents with what they modify. There is no motivation for assigning competing syntactic structures, since the choice of one over the other has no semantic consequence in their set-up.

The output of the parser is passed to an interpreter for pattern matching and inferencing. For example, the '*ammeter*' frame will be enriched with details from its dictionary definition[9]:

an "ammeter" is "an instrument for measuring electric current"

the idea being that parsed segments of the definition can then be isolated and subsequently matched against rules, *e.g.* analysing the fragment as an "ammeter" is "for measuring" permits the creation of a case slot labelled PURPOSE in the ammeter frame whose contents are filled with "measuring".

Different prepositions in English predict different semantic relationships between the objects they relate, *e.g.* the preposition 'for' suggests PURPOSE, while the preposition 'with' suggests POSSESSION, INSTRUMENT, ACCOMPANIMENT, *etc.* Slator [Slator *et al.*, 1990a][Slator *et al.*, 1990b] has carried out an extensive study of the most commonly occurring English prepositions to extract the set of possible semantic relationships that a preposition may represent.

Slator and Wilks [Wilks *et al.*, 1996] go on to show how the knowledge in the enriched *ammeter* frame can be used to select the INSTRUMENT case role between the 'measuring' and the 'ammeter' in the original input sentence, by having an inference rule which associates the PURPOSE and INSTRUMENT case roles, that is, "`X is for PURPOSE Y`" then "`Y uses X as an INSTRUMENT`".

There are a number of problems with this work. Firstly, the example they present is extremely simple and many questions remain unanswered. Much of the complexity is hidden inside pattern matching rules which are not illustrated in detail. For example, the grounds upon which Wilks and Slator decide to choose the appropriate sense of a preposition is unclear[10]. In addition they provide no experimental results

[9] the relationship between a word and its definition can trivially be viewed as an IS-A relation.

[10] *The Cambridge International Dictionary of English* (henceforth CIDE), for instance, lists 16 different senses of the preposition '*with*'.

to demonstrate the number of case roles that can be predicted in an input text although they do stress that the heuristics are the subject of further investigation.

3.5 The Structure of Dictionaries

Several researchers have studied the structure of taxonomies implicit in dictionaries. Many interesting observations have been made regarding the organisation of concepts at the top level of the hierarchy.

Vossen [Vossen, 1990] divides the words occurring in a dictionary (his analysis is of LDOCE) into three distinct levels:

Bottom level — words which do not occur as the heads[11] of other definitions,

Core Level — words which are the most frequent definition heads,

Top Level — a small set of words circularly defined characterised by a high level of polysemy. The practice of dictionaries will necessarily lead to circularity in those cases in which a language has no more abstract words left to describe other words as is the case with *object* and *thing* below:

> *co-star — a famous actor or actress who ...*
> *actor — a man who acts a part in a play*
> *man — a fully-grown human male*
> *male — a male person or animal*
> *person — human being*
> *being — a living thing*
> *thing — any material object*
> *object — a thing*

[11] the word "head" refers to the key noun in a definition.

The top level of the hierarchy is important because it classifies a large number of concepts, *i.e.* properties of these abstract concepts are inherited by potentially many others. Some other interesting aspects of taxonomic chains constructed from LDOCE analysed by Vossen [Vossen, 1990] are:

1. At the highest levels totally distinct objects are attached to the same categories, *e.g. object* and *thing.* This is expected because these concepts are so vague (abstract) that they have few properties which are able to partition the hierarchy of nouns.

2. Very closely related concepts are sometimes not attached to the same category, *e.g. aircraft* and *aeroplane* described as *machine* and *vehicle* respectively, and *skin* versus other body parts like *bone* and *organ.*

3. There are a number of cycles, *e.g. animal* is circularly defined by *creature*, in the hierarchy which stands alone, *i.e.* they are in no way related to other *beings* like *plant* and *person.*

Various strategies of dealing with the circularities at the top of hierarchies have been suggested. Many of these approaches introduce atomic *primitives* (possibly external to the language) to be the initial elements of the taxonomic chains which are extracted.

One such approach is that taken by [Wilks *et al.*, 1996] who create a noun hierarchy (called NounSense) from LDOCE by using the semantic codes provided in LDOCE as the primitive elements. They view genus terms as differentiating headwords with the same semantic code (see section above) and consequently the semantic code is viewed as a primitive concept which supplements otherwise deficient genus terms (and therefore connected headwords) with the appropriate properties. For example, the definition:

> *accessory — (non-movable solid) something which is not necessarily part of something else . . .*

whose semantic code is given in brackets before the definition would give rise to a IS-A hierarchy from the appropriate senses of: *accessory* to *something* to *non-movable solid*. In the case of circularly defined word sets, the most general genus sense is chosen and linked to the semantic category of its headword.

3.6 Existing Knowledge Bases

In general, the goal of AI systems is to reproduce intelligent human behaviour through commonsense reasoning about the world. This undoubtedly requires a vast amount of knowledge. There are many different approaches to constructing knowledge bases which will sit at the heart of these AI systems.

Although there are many knowledge bases in existence (too many to mention) the section below compares and contrasts two of the well known ones: WordNet and Cyc. See [Wilks *et al.*, 1996] for a summary of many of the others used in NLP.

3.6.1 WordNet

WordNet is described as a 'lexical database' in which English nouns, verbs, adjectives and adverbs are organised into sets of synonyms (henceforth synsets), each representing a lexicalised concept (also called word meaning) [Miller, 1990][Miller, 1995]. For example, the two synsets {board,plank} and {board,committee} can serve as two distinct designators for two distinct meanings (or senses) of the word 'board'. Although WordNet contains compounds, phrasal verbs, collocations and idiomatic phrases, the word is the basic unit [Fellbaum, 1998a].

According to [Miller, 1995] WordNet contains more than 118,000 different word forms and more than 90,000 different word senses. Approximately 17% of the words in WordNet are polysemous (have more than a single meaning) and approximately 40% have one or more synonyms.

WordNet is organised by five semantic relations between synsets. These are shown in Table 3.1. By far the most important relations are hyponomy and troponomy. They are symmetric relations which organise nouns and verbs into hierarchies.

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	N V Aj Av	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonomy (opposite)	Aj Av N V	wet, dry rapidly, slowly happiness, unhappiness appear, disappear
Hyponomy (subordinate)	N	sugar maple, maple maple, tree tree, plant
Meronymy (part)	N	brim, hat gin, martini ship, fleet
Troponomy (manner)	V	march, walk whisper, speak
Entailment	V	drive, ride divorce, marry

Table 3.1: Semantic Relations in WordNet.

The combination of the sheer scale (in terms of coverage) of WordNet together with its accessibility[12] means that the project has spawned a lot of diverse research in NLP and computational linguistics. For example, it enables the production of *semantic concordances.* A semantic concordance is "a textual corpus and a lexicon so combined that every substantive word in the text is linked to its appropriate sense in the lexicon" [Miller *et al.*, 1993]. The WordNet team have hand tagged two textual corpora: the Brown corpus which consists of 103 passages from the Standard Corpus of Present-Day Edited American English and the complete text of Stephen Crane's novella "*The Red Badge of Courage*". The procedures associated with the tagging of these corpora are documented in [Landes *et al.*, 1998]. The resulting concordances are useful because they enable the evaluation and training

[12] WordNet is freely available at `http://www.cogsci.princeton.edu/wn/`.

of various NLP applications, *e.g.* WSD algorithms, information extraction systems, *etc.* A recent book [Fellbaum, 1998b] provides a representative summary of different WordNet applications.

WordNet has undoubtedly been a great success in the NLP community. The interest has meant that the project has received constant funding which has resulted in a number of releases, each one attempting to iron out the problems (of which there are undoubtedly many [Fischer, 1997]) in previous releases.

However, with a KB the size of WordNet, it is not surprising to find that there exist a number of problems regarding formal consistency and correctness. Kilgarriff [Kilgarriff, 1998] comments:

> "WordNet is wonderful. It says something about most words of English, it is hierarchically organised, and the particularly wonderful part - it is available free and without restrictions over the net. Computer scientists the world over download it, perform death-defying callisthenics with it and show a 2% improvement in Information Retrieval performance"

His sarcasm is related to the fact that although WordNet has many attractive features, its lexicographic quality is often overlooked by researchers. Fischer [Fischer, 1997] has built a tool to analyse the structure of knowledge bases and used WordNet as a case study. Some of the problems he identifies with WordNet 1.5 are:

Redundancy — there are many types of redundancies in the KB which are classified according to the relationships involved.

For example, Figure 3.2(a) shows that the entailment link from *rub* to *touch* is redundant, and, Figure 3.2(b) shows that the entailment link from *bring along* to *come* is redundant.

Consistency — a number of type consistency errors exist in the KB. For example, in Figure 3.2(c) the troponymy (which is the inverse of a troponomy

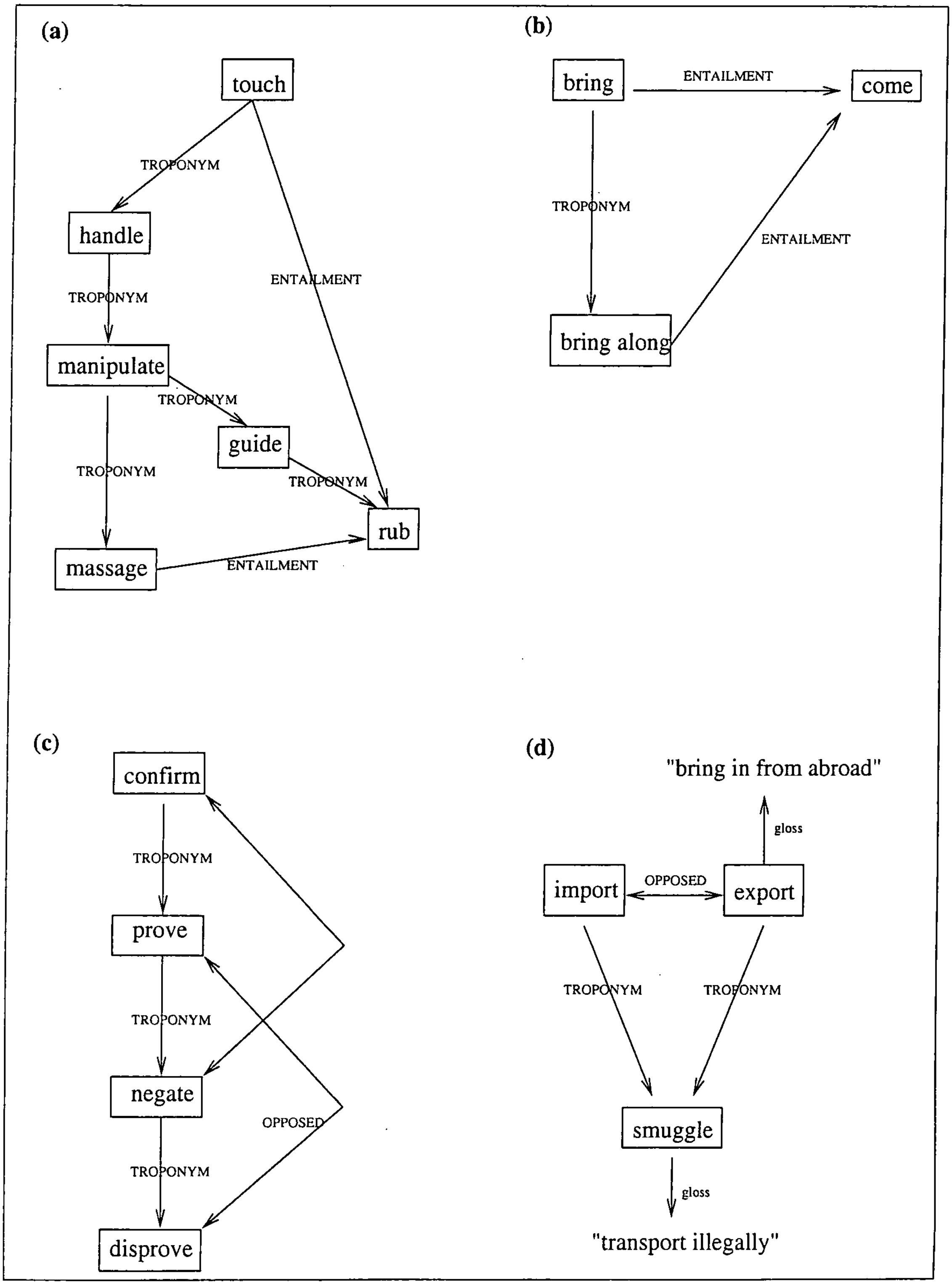


Figure 3.2: Redundancies and inconsistencies in WordNet 1.5

relationship) link from *prove* to *negate* is an error, and it may have its origin in a fallacy [Fischer, 1997]:

> "*To prove by negation* is a troponym of *to prove*, but this is different from *to negate* in the sense of *to show to be false*. In other words, one may prove A by showing that the negation of A is false, but the point is, that the *negation of A* is another object than A, *i.e.* the object to be proved has changed, and indeed, it cannot reasonably be maintained that to negate A is a special way to prove A.

Another example of inconsistency is with a class of disjunctive hypernyms, one of which is shown in Figure 3.2(d). The hypernym is implemented in a way which is harmful because the concept *smuggle* should be a troponym of a concept *export or import*.

The point in illustrating these problems is not simply that the various classes need to be identified and eliminated by the simple removal of a link or the addition of a new concept, but that the problems may reflect a more deep-rooted misunderstandings about word meanings.

3.6.2 The Cyc Project

Perhaps the most famous large-scale knowledge acquisition initiative is the Cyc project at MCC Corporation. The team, led by Doug Lenat identifies a weakness in existing expert systems as "breaking" when faced with unexpected inputs [Lenat *et al.*, 1986] . That is, they operate in very specialised domains, under severe constraints and restrictions and are consequently brittle in more generalised situations. The reason for the brittleness is that they do not possess the mass of general commonsense knowledge that is required. Lenat and his team aim to build Cyc, a flexible knowledge based system capable of reasoning and representing knowledge in generalised domains. They see this system as being at the core of the new wave of expert systems.

Lenat takes a brute force approach in which commonsense knowledge about the world is encoded in some representation with inference viewed as extraction with heuristics. The spirit of the approach is that whenever one level of depth of knowledge fails, the program can reason at the next deeper level.

The representation language chosen at the beginning of the project was a frame-based hierarchy with inference by inheritance. Each frame consisted of a number of slots which contain the distinguishing features of the concept. Many of these frames had exception slots ("Unless" conditions) which contained details about when a concept did not belong in the certain place in the hierarchy, *e.g.* if a *askedInInterview* frame was an instance of the *communication* frame, then, 'person is deaf' could be an exception.

However, in subsequent years, the original frame based language turned out to be too restrictive in its expressiveness [Guha *et al.*, 1990][Guha and Lenat, 1990][Whitten, 1994]. Instead, it is now embedded in a first order framework called CycL. There is not any one particular inference mechanism but a rich variety of techniques are used including default reasoning, support for non-monotonicity, circumscription, logical deduction and so on.

The project had been allotted 10 years to *hand-code* a million entries from a desk encyclopaedia into a large, general purpose knowledge base, an estimated 2 person centuries of work [Lenat *et al.*, 1986]. This point has not yet arrived. It is interesting to note that Guha and Lenat [Guha and Lenat, 1990] have claimed that the notion of Cyc being an electronic encyclopaedia is a myth. Instead the aim is that one day, Cyc will contain enough commonsense knowledge to support NLU capabilities that enable it to read and assimilate any encyclopaedia article. Viewed in this way, it is seen not as an alternative to using a encyclopaedia, but as a complement to it.

The principal criticism of this approach is the sheer volume of effort that it has required [Wilks *et al.*, 1996]. The popular approach in AI is to use lexical knowledge together with basic NLU capabilities as a bootstrapping process in order to acquire more detailed commonsense knowledge. This is the approach taken by the LOLITA developers.

3.6.3 Discussion

WordNet is not everything to everyone. It makes certain assumptions about the nature of lexical concepts in a knowledge representation. A major assumption is that word meanings form useful conceptual partitions which will play a central role in knowledge representations. This view is not shared by all NLP researchers. As Doug Lenat points out [Lenat *et al.*, 1995]:

> "words are often red herrings. They cut up the world along lines drawn for reasons mostly of historical accident of cognation to other languages, of the need for words to be short to allow humans to breathe regularly, and other reasons"

However, he does not provide a convincing argument to support his view. He does stress that [Lenat *et al.*, 1995]:

> "developers must still take one final step — to include concepts that are worth naming but cannot be described by a single word or synset"

which is not necessarily inconsistent with WordNet-based NLP systems, because a lexically-based conceptual-hierarchy, together with basic NLP capabilities can be used as a bootstrap for richer lexical knowledge and subsequently for world knowledge. Lexical concepts are useful in so far as they provide a hook to which other concepts can be attached.

This is at the heart of the different approaches between Cyc and WordNet. Cyc developers take a high-risk high-payoff gamble of encoding a large amount of knowledge in the hope that it will eventually contain the necessary rules and data to enable bootstrapping to even richer levels of knowledge. Other researchers, particularly those whose systems are based on comparatively little knowledge, like that contained in WordNet, hope that the appropriate set of techniques (whether or not based on psychological evidence) can be found to acquire the knowledge and rules which have been hand coded in the former approach.

The LOLITA philosophy is to take the bootstrapping approach, not only because of the limited resources of the project, but because there is no reason to assume that world knowledge cannot be acquired by bootstrapping in this way.

An important point of general agreement is that the amount of knowledge required to achieve the levels of reasoning comparable with humans is undoubtedly still a long way off.

3.7 Summary

This chapter has reviewed some of the research most closely related to the subject of this thesis. There has been plenty of research in the last decade which attempts to exploit the rich knowledge contained in MRDs. Most of this research has concentrated on formalising (*e.g.* identifying commonly occurring patterns in definitions) and processing the most structured part of dictionary entries. The processing of the main text of a definition (particularly the differentia) has posed a more serious problem because it requires the resolution of ambiguity in order to extract the semantic relations from within it. A feasible solution to this problem is the subject of this thesis.

Two well known large-scale KB projects were introduced. They illustrate two very different positions in the construction of KBs: a bootstrapping approach vs. a hand coding approach. The approach taken with LOLITA is the former one. The hope is that knowledge of word meanings will form a core upon which richer types of knowledge can be built.

Chapter 4

The LOLITA System

The LOLITA (Large-Scale, Object-based, Linguistic Interactor, Translator and Analyser) system has been under development at the University of Durham since 1986. It is a large project with many researchers simultaneously working on different aspects of the system. Research in the group follows a pragmatic approach to NLP; it is the production of a robust and useful working system that is of primary interest. This pragmatic view has spawned a new field of Natural Language Research termed NLE which we feel best describes the adopted methodology.

The LOLITA system is designed as a *general purpose* base which forms a core platform upon which different applications can be built. One motivation for developing a general purpose base is that the core of the system may be reused for different tasks and applications. This addresses one of the defining criteria of NLE – that of *flexibility*.

The remainder of this chapter is used to provide an introduction to the LOLITA system by introducing components of the core, followed by descriptions of the various applications which utilise it.

4.1 Architecture of the LOLITA Core

The core components of the LOLITA system are shown in Figure 4.1. Conceptually it can be thought of as consisting of three major processes:

1. Analysis – the mapping of text into some logical representation of its meaning.
2. Inference – the process of deriving inferences from the logical representation of some text.
3. Generation – the conversion of information represented in a logical form into text.

Each of these processes interacts with the heart of LOLITA: the knowledge base, which is a type of **Sem**antic **Net**work called **SemNet**. The analysis phase mainly writes the logical form of text into SemNet[1], the inference component reads knowledge from and possibly writes inferences to SemNet and then the generation module traverses SemNet in order to verbalise knowledge.

4.2 SemNet Basics

In common with semantic networks, SemNet is a graph-based representation, where concepts and relationships are represented by nodes and arcs respectively, with knowledge being elicited by graph traversal. The power of SemNet lies in the efficient inference procedures it is designed for, namely inheritance [Long and Garigliano, 1988]. In addition, it supports many other forms of reasoning, *e.g.* analogy [Long and Garigliano, 1994], epistemic reasoning, time and location [Baring-Gould, forthcoming] and standard logical connective reasoning.

[1] in actual fact the analysis phase also reads knowledge about word meanings from SemNet

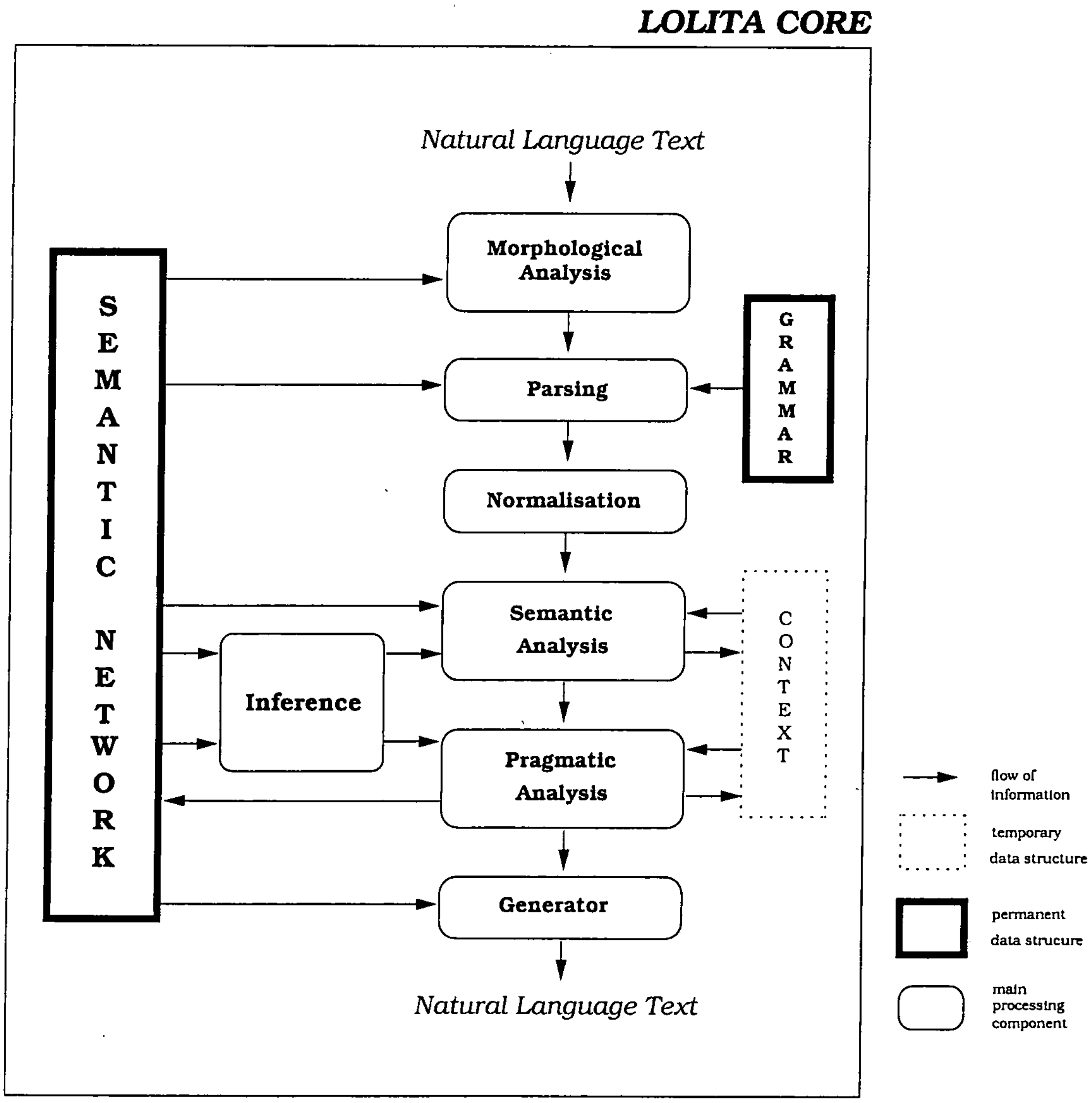


Figure 4.1: The LOLITA core

Currently, SemNet comprises approximately 100,000 nodes most of which have been derived from WordNet [Miller, 1990][Fellbaum, 1998b] whose basic structure was outlined in Section 3.6.1. The original WordNet 1.4 hierarchy from which SemNet was derived has been altered in many ways:

- the addition of various types of controls (see below) some of which represent the subject and box codes available in the MRD of LDOCE.
- the addition of useful concepts which are not present in WordNet, *e.g.* WordNet contains a single meaning of '*buoy*' as in the sense `buoy-as-signal`. A second meaning has been added to represent the sense `buoy-as-entity`. The former sense represents the concept of *buoy* in the input sentence "they saw the buoy and turned back" and the latter sense in the input "the buoy punctured the dingy".
- the removal of many WordNet concepts representing sense distinctions not considered to be useful for NLP purposes.

There are three types of nodes in SemNet: entities, events (assertions) and actions (roles), and three main types of directed arcs[2]: subject, object and action which can be read/traversed in either direction. Only event nodes can have a subject, object or action arc, and only action nodes can be an action for an event node.

There are a number (approximately 50) of control variables which are stored at each node for quick access[3]. The most important of these are:

Rank – provides information about the quantification of the node. Instead of having two kinds of entities – variables and constants, and quantifying over

[2] in actual fact, there are a large number of arcs for efficiency reasons, *e.g.* location, time, source *etc.* However these are reducible to the three basic ones listed.

[3] alternatively each control may be attached to the node as an event expressing the desired information.

variables (as in first-order logic), all entities are simply different types of constants, indicated by the rank, which obey different inferential rules [Garigliano *et al.*, 1993]. The most common values for the rank control are:

- **Universal** [U] refers to all instances of some concept, *e.g.* in a representation of the statement, "*every human has a head*" the concept representing *humans* will have rank: universal.
- **Existential** [E] refers to all instances of a concept, where any particular instance depends on the particular instance of some other universally quantified concept involved in the event. In the example given above, the concept of "*head*" will have rank: existential, because any particular instance of the concept "*head*", will depend upon the particular instance of "*human*".
- **Individual** [I] refers to a concept as a whole, *e.g.* in representing the statement, "*John ate a meal*", the concept of "*a meal*" is a single object and not, for example, the set of all meals.
- **Named Individual** [NI] is similar to the individual rank except that the concept has a fixed name, *e.g.* as in "*John*" above.

Type – records information concerning the type of node, the most important being: entity, action, event and attribute. The type often corresponds to the grammatical category of the linguistic word which represents the meaning of the node. For example, the relation type mainly corresponds to verbs, attribute type to adjectives and entity type to nouns.

Family – this control classifies nodes into particular semantic groups to which they belong. Examples of families are: living, animal, human, abstract, concrete, organisation and location. Although this information could easily be inferred by traversing the static IS-A hierarchy for any particular node, the families provide convenient boundaries useful in core language understanding tasks such as word sense disambiguation and reference resolution because they represent categories where many selectional restrictions change.

The particular categories divide nodes in the hierarchy which are conceptually different. They form the basis of selectional restrictions. For example, there are not many properties which hold of the set of *bankers* which do not hold of all *humans* except that they work in a bank. Conceptually they are very similar and so there is little point in introducing a family called *bankers*. However there are many properties that hold of *humans* which do not hold of *machines*. Consequently the family of *humans* represents an important category within the hierarchy.

4.2.1 Concepts, Meaning and Language

No concept in SemNet has a pre-defined meaning. The meaning of a concept is defined by its location in the network. To put things another way, the meaning of a concept can only be established when the meaning of all its neighbours has been established, and so on. Ultimately a concept can only be interpreted in the presence of the entire network.

A concept can represent a simple entity such as an "apple" or in the case of events it may represent a more complex phenomenon, *e.g.* "a terrorist bombing". Some concepts in the network will be 'static' (those corresponding to LOLITA's background world knowledge), and other 'dynamic' which will be built as the system analyses some text.

One immediate question is, which concepts should form the static knowledge of LOLITA? Our view is that language is concept driven: language has evolved so that words are available for concepts that need to be talked about. Whether a concept is needed depends mainly upon the culture and environment. For example, the Eskimos have more different words for describing types of snow than the English language because these types of snow need to be talked about in their environment. A consequence of this view is that concepts are seen to have a smaller grain size than words; every word has an associated concept but not vice versa.

The practical effect of this observation is that many SemNet concepts correspond directly to (or, represent the meaning of) English words, since LOLITA mainly processes English text (this is where WordNet has proved fruitful [Miller, 1990]). However SemNet is language independent in the sense that concepts are not necessarily *indexed* by these English words. In fact, LOLITA is able to process Italian, Spanish [Fernandez, 1995] and Chinese [Morgan *et al.*, 1994].

One last thing should be mentioned about the abuse of notation which takes place throughout this thesis (although it is standard practise). In diagrammatic representations of SemNet, nodes are given linguistic names such as "animal" and "food" where they should correctly be given some other language independent referent (such as a number) and have an attached event which indicates the word used to commonly refer to the node (if this information exists). Indeed, in the cases where the concept has a word which describes it in many languages, the node will have many different linguistic events attached, one for each language.

4.2.2 Some Examples

Figure 4.2(a) shows an example of a traditional IS-A (or subset) hierarchy. The subset relationship between entities (represented as universal sets, *e.g.* the set of all humans) is specified in two different ways; either by a direct link between the two concepts via a specialisation (abbreviated `spec_`) link or by having an `is_a` event[4] linking the two concepts. The difference is that whilst the former has no associated time (*e.g.* humans are always animals) the latter method allows the event to have an associated time (*e.g.* sellers of an object are only owners of the object before it is sold). Hence `spec_` links are faster equivalents of `is_a` events since they do not require the overhead of reasoning about time. Part (b) of the figure shows an action hierarchy. It is used to derive inferences about the occurrence (or non occurrence) of events (in this sense it can be viewed as an event hierarchy). For

[4] often we name an event by the action or use the name of an action when we more correctly mean the event of the action's occurrence.

example, knowing that "*Sanjay fried a steak*" we can infer that "*Sanjay cooked a steak*", and knowing that "*Sanjay did not cook an egg*" we may infer that "*Sanjay did not fry an egg*".

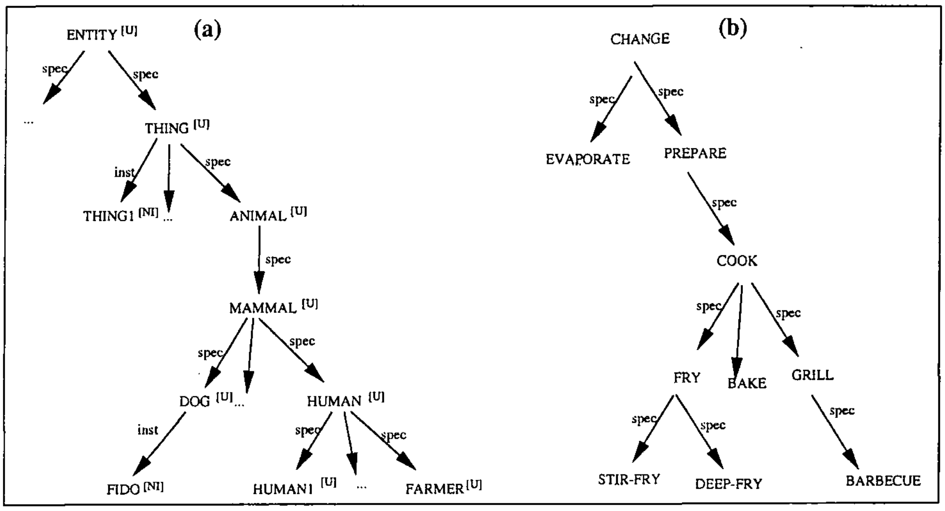


Figure 4.2: Figure (a) shows a fragment of an entity hierarchy in SemNet, while (b) shows a part of an action hierarchy

Figure 4.3 shows some concepts located around an event **E**. Events generally (depending on the action) have a *subject* and *object* arc (often abbreviated `sub_` and `obj_` resp.) which specify the particular entities related by the action. SemNet also contains inverse arcs between related concepts for efficiency, *e.g.* concepts related by `sub_` and `obj_` arcs have associated `sub_of_` and `obj_of_` arcs running in opposite direction (not shown in the diagram). The inverse of the `spec_` arc is called `generalisation` (abbreviated `gen_`) which is interpreted as set inclusion. Set membership is represented by an *instance* (abbreviated `inst_`) arc, *e.g.* Sanjay is an *instance* of all humans. The reverse of an *instance* arc is called a *universal*.

4.3 Analysis

The front end analysis phase of LOLITA was previously described as the process of transforming natural language into its logical meaning represented as newly created events and entities in SemNet. In actual fact, analysis consists of a number

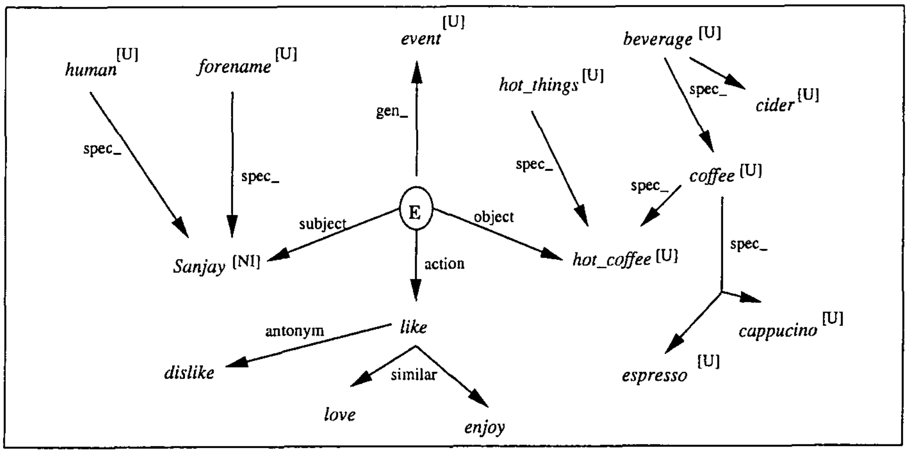


Figure 4.3: A portion of SemNet around an event **E**, expressing the statement *"Sanjay likes hot coffee"*.

of sub-processes outlined below.

4.3.1 Morphology and Parsing

Morphology is the initial preparation of input text before it can be parsed. It consists of using punctuation and spaces in the input to separate it into grammatical units, and replaces short hand words by their longer versions, *e.g. I'll* by *I will*. In addition, there is also a facility for recovering misspelt words [Parker, 1994] and guessing unknown ones.

The LOLITA parser provides large-scale coverage, *i.e.* it is able to deal with full and serious text such as newspaper articles. The grammar rules (over 1500 of them) are expressed in a BNF notation. The parser uses a top-down Tomita-style approach, building a parse forest of all possible parses from the input (see Figure 4.4). The grammar uses a set of features and penalties to order and hence select the most likely parses of the input. The LOLITA grammar was built with the aim of also dealing with erroneous and incomplete input (*e.g.* real-life speech and fragments of NL utterances).

The parser produces the best parse or a list of possible parses representing the grammatical structure of the input, with all word features extracted, errors (structural or feature caused) printed, missing parts inferred and un-parseable parts isolated.

```
sen                                                 sen
 full_propernoun                                     full_propernoun
   propernoun_not_comp SANJAY [New]                    propernoun_not_comp SANJAY [New]
 transvp                                             auxphrase_advprepph
   verb EAT [Past] * 3                                 transvp
   detph                                                 verb EAT [Past] * 3
     det THE                                             detph
     relprepcl                                             det THE
       comnoun STEAK [Sing,Neutral,Per3]                   comnoun STEAK [Sing,Neutral,Per3]
       prepp                                           prepp
         prepNormRel IN                                  prepNormMode IN
         detph                                           detph
           det THE                                         det THE
           comnoun KITCHEN [Sing,Neutral,Per3]             comnoun KITCHEN [Sing,Neutral,Per3]
```

Figure 4.4: The two possibles parses of the sentence "*Sanjay ate a steak in the kitchen*"

4.3.2 Semantic and Pragmatic Analysis

The task of semantic analysis is to map the deep grammatical representation of the input provided by the parsing component onto nodes in SemNet (as shown in Figure 4.5).

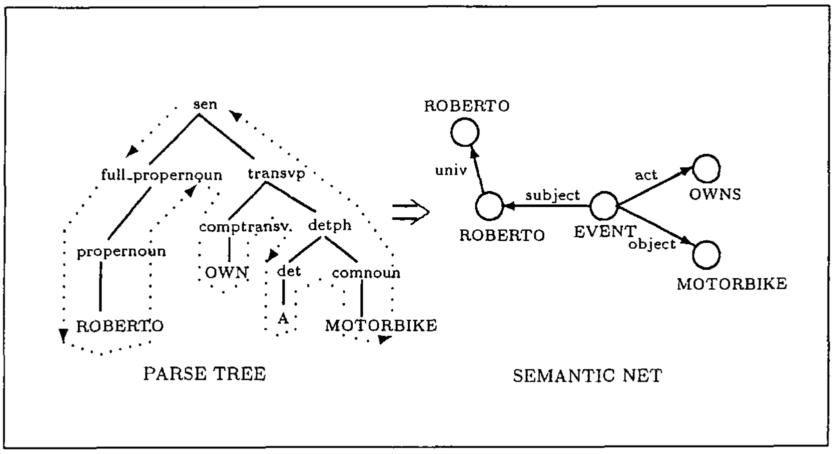


Figure 4.5: A simplified example of semantic analysis. The input (a parse tree) is transformed into a section of SemNet.

To do this, the network has to be checked for the existence of nodes that already represent the concepts in the input, and decisions have to be taken on when to generate new nodes and how to connect them to the rest of SemNet. Amongst other things, this involves anaphora resolution and making deictic references absolute;

"tomorrow" will be expanded into the date after the utterance event, "I" will be resolved into a reference of the speaker, *etc.*

An example of nodes which are created by LOLITA, during semantic analysis of the input sentence "*Roberto owns a motorbike*" are shown in the trace in Figure 4.6. The former node (which is identified by an internal reference number called a noderef, in this case 96177) corresponds to the concept of the particular motorbike and the latter node (which has noderef 96178) represents the assertion of the entire sentence, *i.e.* the event of ownership. The trace shows different types of links (see Section 4.2.1) from the newly created concept, to existing or other new concepts. The numbers are the noderefs of the target nodes. Finally each node is informally described by the LOLITA generator [Smith, 1996].

Further semantic and pragmatic analyses ensure that, after a new or modified portion of the SemNet has been built on the basis of previous stages, this portion is consistent with the existing network. Pragmatic rules come into play with input sentences like "I saw a tree move over the house", where no obvious syntactical and semantic rules are violated, or "I bought one of those Japanese TVs made by Philips", where it is highly unlikely and undesirable to extend the coverage of the semantic representation to world knowledge (stating that Philips is not a Japanese manufacturer of consumer electronics). If pragmatic analysis cannot resolve a conflict between new and existing information in SemNet, a low level of belief is attached to the new portion of the semantic network resulting from the input.

Another way of deciding on the acceptability of input is to use a technique called source control [Bokma and Garigliano, 1992]. It takes into account from whom the information came and the way in which it was provided, *e.g.* a reliable source, an unknown source, part of a chat or a factual news report. A model of source control is incorporated into LOLITA.

```
    * motorbike: 96177 *
 universal_:
   motorbike - 19864 - rank: universal - family: inanimate manmade
 object_of:
   ownership - 96178 - rank: individual  (own)

**********************************
 object:
  Roberto's motorbike.
**********************************


    * ownership: 96178 *
 universal_:
   ownership - 20946 - rank: universal  (own)
 subject_:
   roberto - 19845 - rank: named individual - family: propername human
 action_:
   own - 16943 -
 object_:
   motorbike - 96177 - rank: individual - family: inanimate manmade
 time_:
   present_ - 20989 -
 source_:
   Rick - 96175 - rank: named individual - family: propername human

**********************************
 event:
  Roberto owns a motorbike.
**********************************
```

Figure 4.6: SemNet nodes created by LOLITA during the analysis of the input "*Roberto owns a motorbike*"

4.3.3 Referring back to the Original Text

Before MUC-6, LOLITA did not have a method of referring back to its input: the previous orientation was to move from language-dependent surface forms to a language-independent logical representation. Therefore, information about the surface form was discarded. Since the ability of this sort of reference has many uses (*e.g.* in MUC tasks [DAR, 1995]), a more general mechanism was designed and added to the core. It allows fine-grained connection of the analysis results to the sections of the document giving rise to those results. The system allocates new SemNet nodes to components of the document (words, phrases, sentences, ...), which act as references into the document. This is called the 'Textref' system and has several uses:

- It allows the core to analyse input which talks about surface components of the input text. For example, a user might be able to ask 'What is meant by "organisation" in the second paragraph of the document?', or make statements such as 'When I wrote "pointing", I was referring to brickwork'.
- It enables applications to produce output which is highly related to the original text. The MUC tasks are an example of this, since they require the exact reproduction of the original text. Another possibility is the provision of hypertext-style links to the relevant parts of the original documents in information extraction or summarisation tasks.
- Many LOLITA applications have relied on the system's generator [Smith, 1996] to produce output. This generator relies heavily on the core analysis, and although it performs well given a correct analysis, errors in the analysis can produce very strange output and a drastic reduction in the perceived performance of the system. Textrefs enable more robust reporting of results (*e.g.* see Section 6.6), as witnessed in a significant performance improvement in the non-MUC template generation applications.
- The Textref system can also be used to provide convenient debugging infor-

mation, since it allows developers to relate internal structures produced by the system to the portions of the text from which they were derived.

Textrefs allow the document structure to be fully represented in the net, and represented uniformly with the other information in the system. At the word level, a Textref signifies a specific *occurrence* of a word at a certain position in the input, and is distinct from the nodes representing the lexical or semantic forms of its root form. It is an `instance_` of the universal concept of all occurrences of that word. Concept nodes and Textref nodes are linked by an event with the internal action `words_used`.

Three examples of `words_used` events can be seen in the semantic representation of the phrase "*Sanjay likes hot coffee*" shown in Figure 4.7: the entity '*Sanjay*', the '*hot coffee*' and the entire sentence "*Sanjay likes hot coffee*".

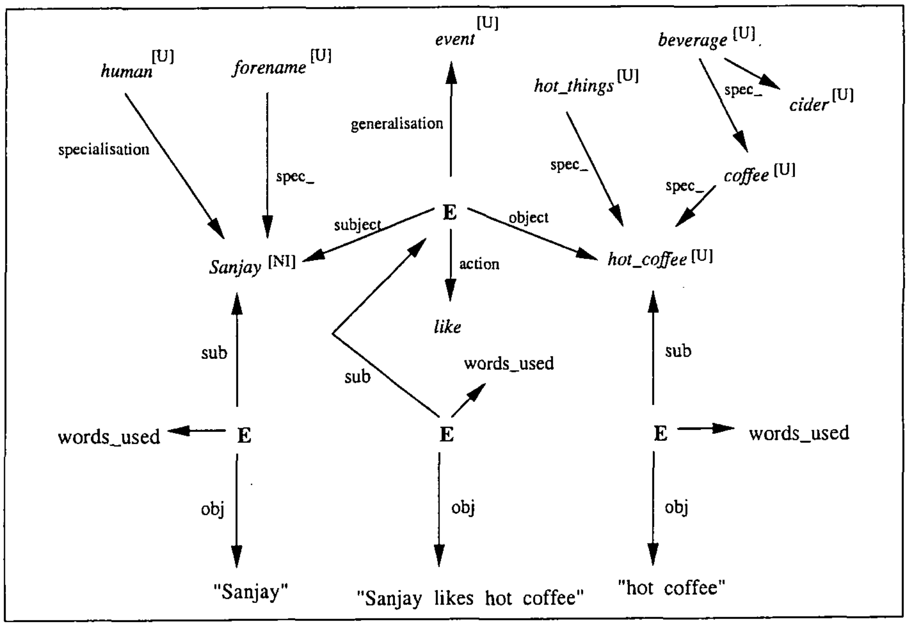


Figure 4.7: A portion of SemNet with internal Textref events, resulting from the analysis of the statement "Sanjay likes hot coffee"

4.4 Generation

The LOLITA generator [Smith, 1996] was, like the rest of the LOLITA system, developed without any specific restrictions imposed by a particular application and is thus very flexible. It is capable of generating NL utterances from SemNet and is widely used as an interface to LOLITA and as a debugging tool. Its input consists of a chunk of SemNet, and its output is an NL utterance whose complexity (*e.g.* phrase, entity, sentence, *etc.*) depends on parameters such as the particular application, the context, the required style and the previous dialogue where applicable.

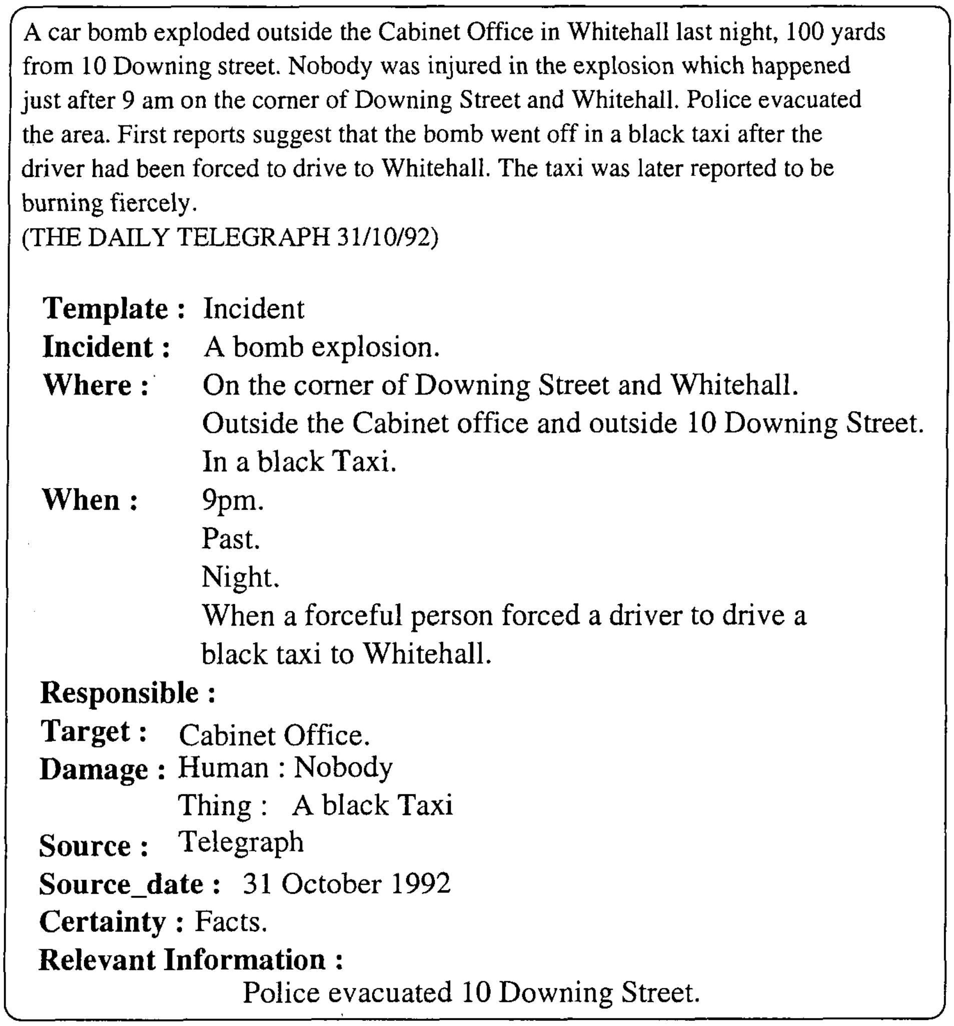
4.5 LOLITA Applications

There exist many applications built around the general purpose base outlined previously. Their great diversity illustrates the flexibility of the system. These applications are briefly introduced below:

- **Contents Scanning**

  Summarising templates are filled from input texts. Input text is parsed and semantically analysed to arrive at a representation of its meaning in SemNet. An application, *i.e.* a domain dependent module, then accesses SemNet in order to find relevant information to fill the template slots. Further information on the use and applications of contents scanning can be found in the literature [Garigliano *et al.*, 1993]. The task of contents scanning is one of the standard tests for evaluating NL systems [Long and Garigliano, 1994]. Figure 4.8 gives an example of content scanning in the LOLITA system. Recent work has increased the flexibility of the contents scanning application by allowing user definable templates [Costantino *et al.*, 1996].

- **Machine Translation**



A car bomb exploded outside the Cabinet Office in Whitehall last night, 100 yards from 10 Downing street. Nobody was injured in the explosion which happened just after 9 am on the corner of Downing Street and Whitehall. Police evacuated the area. First reports suggest that the bomb went off in a black taxi after the driver had been forced to drive to Whitehall. The taxi was later reported to be burning fiercely.
(THE DAILY TELEGRAPH 31/10/92)

Template : Incident
Incident : A bomb explosion.
Where : On the corner of Downing Street and Whitehall.
Outside the Cabinet office and outside 10 Downing Street.
In a black Taxi.
When : 9pm.
Past.
Night.
When a forceful person forced a driver to drive a
black taxi to Whitehall.
Responsible :
Target : Cabinet Office.
Damage : Human : Nobody
Thing : A black Taxi
Source : Telegraph
Source_date : 31 October 1992
Certainty : Facts.
Relevant Information :
Police evacuated 10 Downing Street.

Figure 4.8: Example of a template produced by the contents scanning application

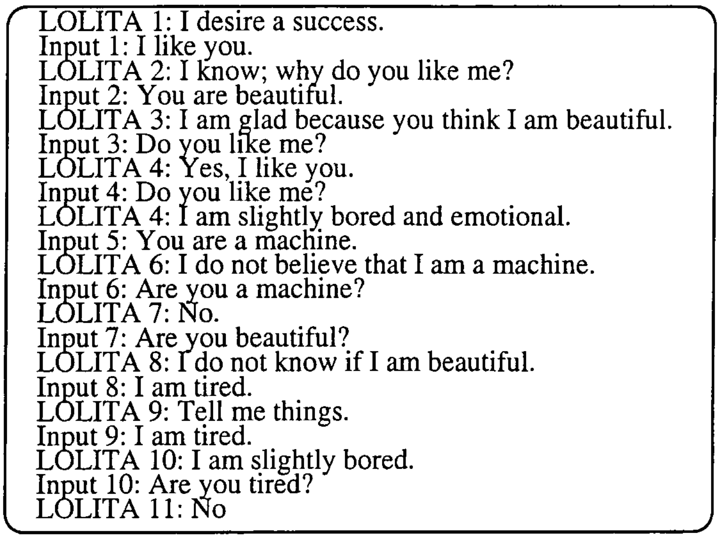
Although machine translation (MT) was not one of the goals the developers of LOLITA had in mind, a prototype MT system was built with only a small amount of modification. After the addition of a number of rules to the grammar, it became possible to analyse Italian and Chinese texts, adding their semantic content to SemNet. This information could subsequently be realised as English NL utterances by the generation module. Clearly the quality of the translation is not as refined as that of dedicated MT systems, but it has to be kept in mind that the MT application was neither one of the original goals nor is it a finished application.

- **Chinese Language Tutoring**

  The Chinese tutoring prototype helps students learning Chinese overcome the problem of transfer errors caused by mother tongue influence [Wang and Garigliano, 1992][Wang, 1994]. By using the existing parser and adding Chinese words and grammar rules a tutoring module has been built which uses intelligent tutoring techniques (by using and updating various models of the situation, *e.g.* the student and expert model) to ask users to perform English to Chinese translations. Users' translations are then parsed and the resulting parse tree is diagnosed for transfer errors.

- **Dialogue**

  A dialogue model loosely based on Schank's script theory [Schank and Abelson, 1977] has been implemented [Jones, 1993][Jones, 1994]. An example of the dialogue application [Nettleton and Garigliano, 1995] is shown in Figure 4.9. Since communicative behaviour in a dialogue, *e.g.* the appropriateness of a response or the right to initiate a dialogue and to change the topic, depends mainly on different situations and contexts in which the dialogue takes place, and since there are, theoretically, an infinite number of possible contexts or dialogue situations, the dialogue module of LOLITA makes use of prototypical Dialogue Structure Models (DSMs) representing a class of dialogues. DSMs are composed of Dialogue Structure Elements (DSE), each of which describes a fundamental property of a dialogue. The Dominance DSE, for example, comprises such dialogue constraints as whether it is permissable to initiate and terminate a (sub-) dialogue or to interrupt other participants (by choosing a different topic or referring to a previously mentioned one).



```
LOLITA 1: I desire a success.
Input 1: I like you.
LOLITA 2: I know; why do you like me?
Input 2: You are beautiful.
LOLITA 3: I am glad because you think I am beautiful.
Input 3: Do you like me?
LOLITA 4: Yes, I like you.
Input 4: Do you like me?
LOLITA 4: I am slightly bored and emotional.
Input 5: You are a machine.
LOLITA 6: I do not believe that I am a machine.
Input 6: Are you a machine?
LOLITA 7: No.
Input 7: Are you beautiful?
LOLITA 8: I do not know if I am beautiful.
Input 8: I am tired.
LOLITA 9: Tell me things.
Input 9: I am tired.
LOLITA 10: I am slightly bored.
Input 10: Are you tired?
LOLITA 11: No
```

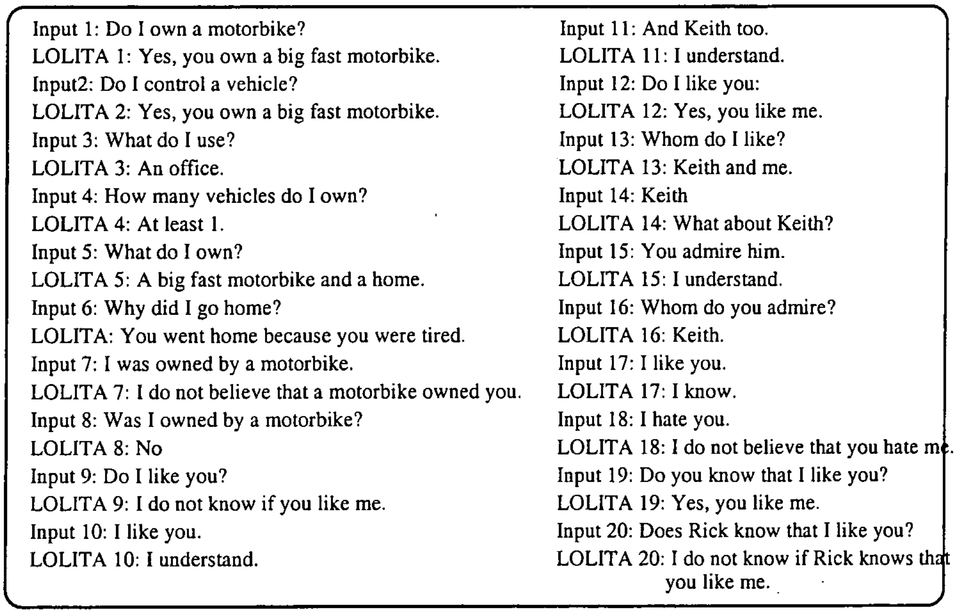
Figure 4.9: An Example of the Dialogue Application

- **Query and Database Interfacing**

  This application allows information to be entered into the LOLITA knowledge base, SemNet, in the form of NL, either by typing in sentences or by having the system analyse files in batch mode, *e.g.* newspaper articles. The information extracted from the input is stored in the SemNet and can be retrieved using NL input, by typing a question relating to information contained in the original input [Garigliano *et al.*, 1996]. An example can be seen in Figure 4.10.

4.6 Summary

An overview of the LOLITA NLP system was presented. Input to the system consists of a natural language text to be processed. This text is analysed, *i.e.* parsed and a representation of its meaning is created as new events asserted within SemNet. In order to do this the analysis must handle the inherent ambiguity in



```
Input 1: Do I own a motorbike?
LOLITA 1: Yes, you own a big fast motorbike.
Input2: Do I control a vehicle?
LOLITA 2: Yes, you own a big fast motorbike.
Input 3: What do I use?
LOLITA 3: An office.
Input 4: How many vehicles do I own?
LOLITA 4: At least 1.
Input 5: What do I own?
LOLITA 5: A big fast motorbike and a home.
Input 6: Why did I go home?
LOLITA: You went home because you were tired.
Input 7: I was owned by a motorbike.
LOLITA 7: I do not believe that a motorbike owned you.
Input 8: Was I owned by a motorbike?
LOLITA 8: No
Input 9: Do I like you?
LOLITA 9: I do not know if you like me.
Input 10: I like you.
LOLITA 10: I understand.
Input 11: And Keith too.
LOLITA 11: I understand.
Input 12: Do I like you:
LOLITA 12: Yes, you like me.
Input 13: Whom do I like?
LOLITA 13: Keith and me.
Input 14: Keith
LOLITA 14: What about Keith?
Input 15: You admire him.
LOLITA 15: I understand.
Input 16: Whom do you admire?
LOLITA 16: Keith.
Input 17: I like you.
LOLITA 17: I know.
Input 18: I hate you.
LOLITA 18: I do not believe that you hate me.
Input 19: Do you know that I like you?
LOLITA 19: Yes, you like me.
Input 20: Does Rick know that I like you?
LOLITA 20: I do not know if Rick knows that
           you like me.
```

Figure 4.10: An Example of the Query Application

natural language, *e.g.* parsing ambiguities, word sense ambiguity and anaphora resolution being the most common.

Once the unambiguous logical form of the input is represented in SemNet, inference procedures may operate upon the knowledge. These procedures range from basic inheritance to more complex forms such as induction and analogy.

The back-end of LOLITA consists of the generation module which traverses SemNet and verbalises the knowledge contained within it as NL utterances.

This core system has many applications built upon it. They range from language tutoring and machine translation through to dialogue. The range of applications illustrates the versatility of the core system.

As mentioned in Chapter 1, this research is to be conducted within the framework of the LOLITA system. In order to meet the methodological criteria of the research, any knowledge which is extracted from a dictionary will need to be encoded in a form which can be utilised by the LOLITA system. This will undoubtedly place many constraints on the knowledge acquisition framework. These constraints are discussed at the relevant points in the thesis.

Chapter 5

A Framework for Knowledge Acquisition

This chapter introduces a framework for the acquisition and interpretation of dictionary definitions. Although some of the issues are discussed in the context of the LOLITA system, they are generally applicable to any knowledge acquisition process which attempts to extract and represent semantic relationships for use by an NLP system.

The combination of approaches which are outlined below are chosen with consideration to the criteria laid out in Chapter 1 and in answer to more problem dependent questions: Can MRDs provide knowledge in the form required by NLP systems? If not, how might the knowledge be transformed to accommodate this? What are the problems in the interpretation process? How might these be tackled in a feasible way? Many of the issues involved in answering these questions are discussed in this chapter.

5.1 CIDE as the Lexical Source

The LOLITA project has acquired a machine readable version of the *Cambridge International Dictionary of English* [Procter, 1995] (CIDE henceforth) together with a licence which permits knowledge to be extracted from it and used as input to an NLP system.

CIDE is a learners' dictionary containing 100,000 words and phrases arranged alphabetically under 50,000 headwords. Within each entry there is a wide range of information: inflected word forms are given, as are examples and usage, idioms, collocations, false friends and grammatical description. The grammar codes are simple, with every one being attached to an example sentence.

There are three important features which make CIDE particularly suitable for the extraction of semantic information:

1. definitions are written in a way which allows them to be processed easily. For example, compare the definitions of the adjectives *cutesy* in CIDE and COBUILD respectively:

   cutesy $_{adj}$ — artifically attractive and charming, esp. in a childlike way

   cutesy $_{adj}$ — If you describe someone or something as cutesy, you dislike them because you think they are unpleasantly pretty and sentimental; an informal word.

   The latter definition (COBUILD prefer to call it an explanation) is much more verbose and informal. It would require considerably more complex rules to process it.

2. every definition is written using a limited set of words known as the *defining vocabulary.* The defining vocabulary contains 2000 words, many of which are used only in restricted senses. CIDE claim that words in the defining vocabulary:

- are common words of high frequency
- are useful for explaining other words
- have the same meaning in British and American English

The use of a dictionary in which definitions are written using a restricted set of words is desirable because it will reduce the number of possible interpretations of a definition, by restricting the possible meanings of the words used within it. To our knowledge, CIDE and LDOCE are the only two dictionaries with this property[1].

3. the other innovative feature of CIDE is the way in which entries are laid out in the dictionary.

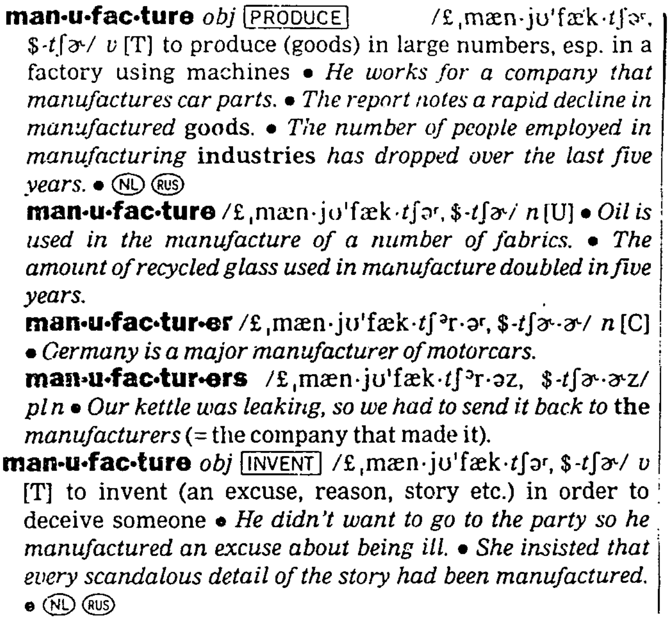
   Successive entries in most dictionaries represent distinct homographs of a word form; that is a set of senses of a word form when it serves as noun, verb or some other part of speech. However, each entry in CIDE contains all the words which share similar form and meaning (even though they may be different parts of speech) to the headword of the main definition. Figure 5.1 shows the CIDE and LDOCE entries around the verb '*manufacture*'.

   The CIDE layout is advantageous because semantically related words are grouped in the same entry. The way in which this feature is exploited is elaborated in sections 6.8 and 6.11.

5.2 A General Approach to Knowledge Acquisition

In order to design a procedure for extracting knowledge from CIDE, a number of issues need to be dealt with. They will give rise to a framework for the algorithm introduced in Chapter 6.

[1] It is interesting to note that Paul Procter was editor of both LDOCE and CIDE.



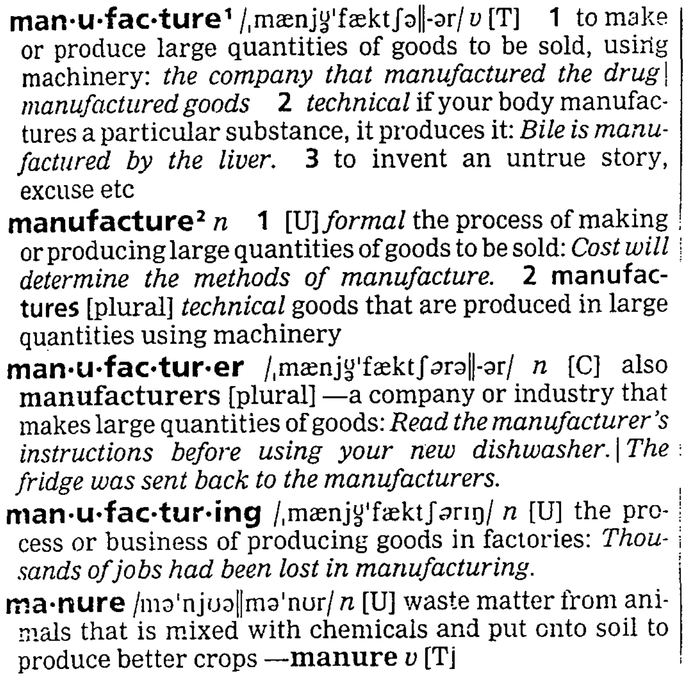
man·u·fac·ture *obj* PRODUCE /£ˌmæn·jʊ'fæk·*t*ʃəʳ, $-*t*ʃɚ/ *v* [T] to produce (goods) in large numbers, esp. in a factory using machines • *He works for a company that manufactures car parts.* • *The report notes a rapid decline in manufactured* goods. • *The number of people employed in manufacturing* **industries** *has dropped over the last five years.* • (NL) (RUS)

man·u·fac·ture /£ˌmæn·jʊ'fæk·*t*ʃəʳ, $-*t*ʃɚ/ *n* [U] • *Oil is used in the manufacture of a number of fabrics.* • *The amount of recycled glass used in manufacture doubled in five years.*

man·u·fac·tur·er /£ˌmæn·jʊ'fæk·*t*ʃᵊr·əʳ, $-*t*ʃɚ·ɚ/ *n* [C] • *Germany is a major manufacturer of motorcars.*

man·u·fac·tur·ers /£ˌmæn·jʊ'fæk·*t*ʃᵊr·əz, $-*t*ʃɚ·ɚz/ *pl n* • *Our kettle was leaking, so we had to send it back to* **the** *manufacturers* (= the company that made it).

man·u·fac·ture *obj* INVENT /£ˌmæn·jʊ'fæk·*t*ʃəʳ, $-*t*ʃɚ/ *v* [T] to invent (an excuse, reason, story etc.) in order to deceive someone • *He didn't want to go to the party so he manufactured an excuse about being ill.* • *She insisted that every scandalous detail of the story had been manufactured.* • (NL) (RUS)



man·u·fac·ture¹ /ˌmænjʊ'fæktʃə‖-ər/ *v* [T] **1** to make or produce large quantities of goods to be sold, using machinery: *the company that manufactured the drug* | *manufactured goods* **2** *technical* if your body manufactures a particular substance, it produces it: *Bile is manufactured by the liver.* **3** to invent an untrue story, excuse etc

manufacture² *n* **1** [U] *formal* the process of making or producing large quantities of goods to be sold: *Cost will determine the methods of manufacture.* **2** **manufactures** [plural] *technical* goods that are produced in large quantities using machinery

man·u·fac·tur·er /ˌmænjʊ'fæktʃərə‖-ər/ *n* [C] also **manufacturers** [plural] —a company or industry that makes large quantities of goods: *Read the manufacturer's instructions before using your new dishwasher.* | *The fridge was sent back to the manufacturers.*

man·u·fac·tur·ing /ˌmænjʊ'fæktʃərɪŋ/ *n* [U] the process or business of producing goods in factories: *Thousands of jobs had been lost in manufacturing.*

ma·nure /mə'njʊə‖mə'nʊr/ *n* [U] waste matter from animals that is mixed with chemicals and put onto soil to produce better crops —**manure** *v* [Tj

Figure 5.1: Dictionary entries around the definition of the verb *manufacture* for CIDE and LDOCE respectively.

There are two different approaches to the extraction and representation of semantic knowledge, which may subsequently be used by LOLITA. The two possibilities are discussed below.

- The first approach involves *constructing* a semantic network in which the genus terms of definitions form the IS-A hierarchy which will be the spine of the network. The genus network can be enriched by extracting and integrating the semantic relationships contained in the differentia of each definition, in much the same way as the semantic content of free text is currently represented in SemNet (as illustrated in Figure 4.3).

  Although the approach above suggests a two phase process in the building of a semantic network (*i.e.* the construction of a genus hierarchy followed by an enriching process) it can be accomplished by allowing LOLITA to analyse a complete dictionary definition, and extract the semantic relationships from within it. In this way the relationship between headword and genus is extracted in much the same way as other relationships in the differentia.

  The extraction of semantic knowledge in this way requires a bootstrapping process which, at the very least, will extract basic syntactic knowledge (*e.g.* spelling forms, part of speech, *etc.*) of the word senses which are used to define each dictionary entry. The advantage of using a dictionary with a defining vocabulary is clear because this bootstrapping process only needs to extract syntactic information for the restricted set of senses in the defining vocabulary.

- An alternative strategy is to *integrate* the semantic knowledge encoded within CIDE to LOLITA's existing knowledge base, SemNet.

  Given a headword and accompanying definition, this approach involves a mapping process which identifies the concepts in SemNetwhich correspond to the concepts defined by each dictionary definition. The equivalent SemNet concept needs to be identified because it is used as a referent point to which semantic relationships extracted from the definition can be added. Since there are many different ways of viewing the concept defined by a word sense, this mapping process is not necessarily straightforward.

For this research we have opted for the latter approach. The reasons for this decision are illustrated in detail below: the first reason concerns the importance of keeping the knowledge in LOLITA's existing KB, and the other reasons concern the problems of using a dictionary as the only knowledge source in the construction of a KB. These problems are solved if the knowledge contained in a MRD is integrated into LOLITA's existing KB. The reasons are:

WordNet Compatibility — As previously noted, LOLITA's current knowledge base, SemNet, has been derived from WordNet. The LOLITA project would like to maintain this compatibility because:

- there is a desire amongst much of the NLP community to standardise the common parts of an NLP system, *e.g.* the use of Sowa's conceptual graphs [Sowa, 1984] as a knowledge representation language is just one such aim of a core of NLP researchers [NCITS.T2 Committee on Information Interchange and Interpretation, 1998].

  WordNet is a project which has gathered considerable interest in the NLP community because, to our knowledge, it is the largest lexical database which is freely available to the NLP community. Consequently, there is little doubt that WordNet is the single candidate which would emerge if the lexical knowledge bases of NLP systems were to become standardised in the near future. The cost of relinquishing this compatibility with WordNet may be too great in the long run.

- The WordNet project receives a lot of feedback from the language processing community, and it has the necessary resources to make changes in response to this feedback. The multiple releases of WordNet are testament to this fact.

  If a knowledge base is built entirely from CIDE, then inevitably this knowledge will need to be maintained; anomalies will be present and knowledge may need to be restructured. However, this ongoing maintenance is not feasible within a single project such as LOLITA. The hope is that this maintenance will come free if the knowledge base is kept compatible with ongoing releases of WordNet.

- Compatibility with WordNet ensures that many resources can be exploited.

  For example, the EuroWordNet[2] project aims at developing a multilingual database with basic semantic relations between words for several European languages (Dutch, Italian and Spanish) [Vossen, 1998]. The wordnets will be linked to the English WordNet, and a shared top-ontology will be derived. Amongst other applications the databases can be used for multilingual information retrieval [Vossen, 1997].

  Other resources include tagged corpora (*e.g.* the Brown corpus) which can be used to train Word Sense Disambiguation (WSD) algorithms, and graphical viewers for consistency checking.

Deeper Hierarchy — The benefits of hierarchically structured knowledge, to NLP systems, has been demonstrated by many researchers [Boguraev and Briscoe, 1989a]. One of these benefits is the spatial efficiency which results from valid inheritance rules operating on the hierarchy forming relationships.

Specifically, the properties (*e.g. has no limbs*, *has scales*) of an element (*snake*) can be inherited to each of its possible types (*e.g. mamba*, *anaconda*) by the application of inference rules. The spatial efficiency results from the non-duplication of these properties at the level of each of these types.

The degree of exploitation of hierarchically structured knowledge is related to the depth of the hierarchy. A deeper hierarchy means that properties can be inherited to more concepts. However dictionaries tend to encode shallow hierarchies because they are aimed at human readers. Figure 5.2 illustrates this by showing fragments of hierarchies from WordNet and the genus terms

[2] see `http://www.let.uva.nl/ ewn/` for more details.

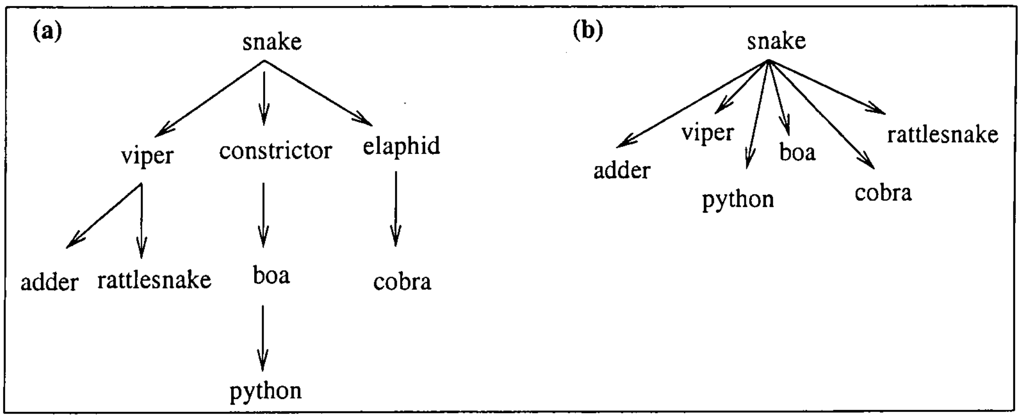


Figure 5.2: Fragments of hierarchies from WordNet and COBUILD rooted at the concept '*snake*'.

of COBUILD respectively, both of which are rooted at the concept '*snake*'. The appropriate definitions from COBUILD are:

boa — a large, strong *snake* that kills animals and birds by wrapping itself round their bodies and squeezing them to death. Boas are found mainly in South and Central America

python — a large very long *snake* that kills animals by squeezing them with its body

cobra — a large poisonous *snake* which can make the skin at the back of its head into a large hood

viper — a small poisonous *snake* found mainly in Britain and Europe. There are several kinds of viper most of which have zigzag patterns down their back

adder — a small poisonous *snake* which has a black zigzag pattern on its back. Adders are found in Europe and Asia

rattlesnake — a poisonous American *snake* with bony rings at the end of its tail which make a rattling sound when the tail is shaken

Given that concepts in (a) and (b) have properties derived from the definitions above, (*e.g. snakes have scales*, *vipers are poisonous*, *boas kill by squeezing*, *etc.*) then the derivable properties of (a) and (b) are identical only if the knowledge in (b) is supplemented thus:

1. properties of *vipers* are added explicitly to *adder* and *rattlesnake*, and,
2. properties of *boas* are added explicitly to *python*.

The problem of spatial inefficiency (resulting from using a flat IS-A hierarchy) is compounded if a dictionary with a defining vocabulary is used to construct the hierarchy. Consider the two definitions of the verb, *sauté*:

sauté — to *cook* food over in oil or fat over heat, usually until it is brown

sauté — if you sauté food then you *fry* quickly in hot oil or butter

Since *frying* is a type of *cooking* the former definition from CIDE gives rise to a flatter IS-A hierarchy than the latter one from COBUILD. This is expected because dictionaries with defining vocabularies have the additional constraint of selecting from a smaller set of genus terms than those without the constraint.

Word Sense Division — The granularity of word senses in dictionaries (which are designed to be comprehensible by human readers) often conflicts with the granularity of senses ideal for NLP systems.

In general, NLP systems prefer a finer division of word senses than dictionary definitions contain, because they permit simple properties of a concept to be stated in a concise way. Consider the following noun definitions[3] from CIDE:

business — the activity of buying and selling goods and services, or a particular company that does this, or work in general rather than pleasure • The firm does a lot of business with overseas customers • Peter ended up in charge of the business • The visit was for business

[3] The format of dictionary definitions shown as examples in this thesis is illustrated by the template:

headword GUIDEWORD *grammar* — definition • example$_1$ • ... • example$_n$

The GUIDEWORD is used to indicate the sense of the headword which is being defined and the label '*gram*' refers to various types of grammatical information provided in CIDE, *e.g. obj* defines a transitive verb, (*obj*) defines verbs which can occur in both transitive and intransitive forms with the particular form specified by the labels [T] or [I] following the examples which illustrates its usage. The grammatical information is omitted (or simplified) in contexts when it is obvious or unimportant.

child — a boy or girl from the time of birth until he or she is an adult, <u>or</u> a son or daughter of any age • when she was a child she was always healthy • he's such a child when he doesn't get his way

which are too coarse for NLP purposes because the properties of the concepts '*business*' and '*child*' defined above, are simply a summation of the properties of each of the sub-parts of the definition[4]. This is clear if some of the intuitive properties of each sub-part of '*business*' are enumerated:

- *the activity of buying and selling goods and services* — takes place over a time period, involves negotiating a price, involves a transfer of goods, *etc.*
- *a particular company that buys and sells goods and services* — has employees, is located at a premises, often buys raw materials, *etc.*
- *work in general rather than pleasure* — is done in an official capacity, is done to earn money, *etc.*

because they show that the individual parts do not share many properties. The benefit of having each sub-part grouped in the same dictionary entry is to indicate semantic relatedness. However, the particular semantic relationships are not explicitly stated.

Consequently, from an NLP perspective, there is no reason why the sub-parts of coarse-grained definitions such as '*business*' should from a single word sense instead of the three distinct senses mentioned above. The fact that the sub-concepts are semantically related can be captured in other ways, *e.g.* by introducing a set of subject codes (as in LDOCE), or, by detailed processing of the definition.

This position is a pragmatic one, given that a goal of an NLP system is to infer as many specific properties as possible. An occurrence of the word

[4] the sub-part of a definition in this context refers to the segments of the definition which are separated by the underlined '*or*'.

'*business*' in a text is uninformative if the only derivable consequence is a disjunctive structure. A better approach is to use local context to eliminate the disjunction. This can be achieved by creating different word senses for each of the disjuncts above and by utilising a WSD algorithm (*e.g.* see [Hawkins and Nettleton, forthcoming]) when the word '*business*' is encountered within a text.

The same problem is also found in the definitions of many verbs in CIDE. Consider the following definitions of verbs which define the concept of *joining* something:

join CONNECT (*obj*) — to connect or fasten things together • the suspension bridge joins the two islands • I joined my car battery to a friend's with leads

connect JOIN (obj) — to join or be joined with something else • He connected his printer to the computer [T] • Where does the cooker connect up to? [I]

fasten (*obj*) — to make or become firmly fixed together or in position, or closed • he checked his seat belt was securely fastened [T] • The shirt fastens at the back [I]

The definition of the transitive verb, *join*, poses the same problem as the nouns discussed previously. It represents a generalisation of two different aspects of the concept of *joining*: *e.g.* two possible views of a situation where an agent, John, glues pieces of paper together, which the definition of the verb *connect* is trying to capture, are:

a. the glue connects the pieces of paper

b. John connects the pieces of paper

The reason why an NLP system prefers a separation of the two shades of the verb *connect* is that the simple properties of the two views above are unconnected:

a. *(inanimate entity) join (inanimate entities)* — then a part of the subject touches each entity that is the object of the verb during the time when the joining holds.

b. *(agent) join (inanimate entities)* — then the agent does a series of actions which result in a state in which the situation [a] holds.

The use of the conjunction '*or*' by lexicographers, to combine shades of different senses of a word, can be dangerous. For example, consider the definitions of the verbs which can occur in both transitive [T] or intransitive [I] forms[5]:

worship — to have or show a strong feeling of respect and admiration for God or a god • They worship in the same mosque [I] • In the various regions of India, Hindus worship different gods and observe different religious festivals [T]

drop — to fall intentionally or unintentionally or to let something fall • The flag dropped and the race started [I] • She dropped the tray with a crash [T]

The two verbs above are implicitly transitive in very different ways. The former example, *worship*, **always** takes an object, which can be implicit (*i.e.* not mentioned) when the verb is used. This is not the case in the latter example because the verb *drop* in "*the flag dropped*" does not take an object. The latter example indicates that the definition of *drop* consists of two distinct senses (one transitive and the other intransitive) of the verb. This is an example of a definition which must be split in the knowledge base of an NLP system for the reasons mentioned previously.

The problems above can be resolved by using SemNet as the framework of the knowledge base because it has a finer granularity of word senses than any dictionary. This is as a direct result of the aim of each of the knowledge bases: while dictionaries are intended to be read by humans, WordNet lexicographers clearly have more computational and representational issues in mind.

[5] which in this thesis are called *transitive implicit* verbs.

Coverage — The analysis of definitions in CIDE would result in a hierarchy which is not as comprehensive as the coverage of WordNet.

The first reason for this is that CIDE is a learners' dictionary not containing many of the specialised words which are included in WordNet.

Secondly, each entry in CIDE contains one main definition with a number of sub-definitions which do not necessarily have explicit defining strings, *e.g.* the nouns '*manufacture*', '*manufacturer*' and '*manufactures*' in Figure 5.1. The lack of knowledge would mean that few inferences could be made regarding the concept of '*manufacturer*', *e.g.* it is synonymous with '*maker*' and that a *manufacturer* is a '*business person*', *etc.*

An approach which enriches the existing KR would mean that these two problems are dealt with automatically.

There is a price to pay for the approach in which the semantic relationships from a MRD are integrated into the existing KB. These are considered below.

Mapping Task — the integration of knowledge extracted from CIDE will require the task of identifying the correct sense of the equivalent SemNet node. This process is an inevitable source of errors which would not exist if the hierarchy was constructed from CIDE alone.

Knowledge Loss — there are often many ways of representing the same concept in a hierarchy because:

- the formalisation of word meanings into semantic hierarchies requires decisions to be made concerning the precise meaning of a word sense
- even if the precise meaning of a word sense is clear, there are many different ways of dividing objects in the world and consequently representing the same concept.

Although, in theory, the IS-A hierarchy between two dictionaries could be vastly different, there are reasons to believe that a hierarchy extracted from an MRD and the WordNet hierarchy will have many sub-hierarchies (and consequently many concepts) in common. This is because both types of dictionary aim to divide the world of entities and relations at the level of single words.

Given that the framework for the integration of knowledge has been laid, the remainder of the chapter expands upon the role of LOLITA within the acquisition process.

5.3 A Knowledge Acquisition Framework

There are three major stages to the knowledge acquisition process:

1. *mapping* from the CIDE defining vocabulary to SemNet.
2. the *extraction* of semantic relationships from CIDE
3. the *representation* and *integration* of the semantic relationships into SemNet

The initial mapping of the CIDE defining vocabulary needs to be done only once at the beginning of the acquisition process. The extraction and integration phases form a cycle which is repeated as each CIDE definition is analysed. The three stages are outlined below.

5.3.1 Mapping from the CIDE Defining Vocabulary to SemNet

This stage involves finding the SemNet concepts which 'correspond' to those defined by words which exist in the CIDE defining vocabulary. This mapping can be used by LOLITA to limit the number of interpretations of the text of each definition.

The mapping process is not entirely straightforward because of the possible variations in the formalisation and representation of word senses as illustrated in Section 5.2. The mapping of a word sense from CIDE to WordNet can be categorised as belonging to one of the types below.

5.3.1.1 CIDE concept maps to a single SemNet concept

This is the case in which there is a direct *correspondence* between a CIDE word sense and a WordNet node. For example, consider the following definitions of words which are listed in the CIDE defining vocabulary:

equip $_v$ — to *provide* a person or a place with objects that are necessary for a particular purpose

bird $_n$ — a *creature* with feathers and wings usually able to fly

dish $_n$ — a *container*, flatter than a bowl and sometimes with a lid, from which food can be served or which can be used for cooking

The concepts defined above have mappings to single WordNet synsets[6]. Figure 5.3 shows three IS-A hierarchies extracted from WordNet. Each one is rooted at the WordNet node which corresponds to the concepts defined above. Each line lists a WordNet synset which represents a lexicalised concept. Successive lines (indented ones beginning with '=>') indicate a hyponym link to the lexicalised concept directly below, *i.e.* the first example illustrates that the concept represented by the synset {equip, appoint, fit,...} IS-A-KIND-OF {supply, provide,...}. WordNet concepts are generally followed by an informal description of the concept which is called a *gloss*. Many of the glosses have been removed from Figure 5.3 (and subsequent WordNet hierarchies) for clarity.

[6] a synset consists of a set of lexical items which are synonymous in some context.

The hierarchies illustrate an equivalent mapping between a CIDE concept (c_1) and a WordNet concept (w_1) as consisting of one the following cases:

1. the superordinate terms of c_1 and w_1 are equivalent.

   This is illustrated with the definition of the verb, '*equip*', given previously, which has *provide* as its superordinate genus term. Figure 5.3 shows that the WordNet synset which contains the word '*equip*' also has the word '*provide*' in the superordinate synset. Hence the concept defined by the word *equip* in CIDE can be directly mapped to this WordNet concept.

2. the superordinate term of c_1 is more specialised than the superordinate of w_1.

   This case is illustrated with the definition of the noun, '*bird*', which has '*animal*' as its genus term. This concept can be mapped to the WordNet synset which contains the single element *bird* (shown in Figure 5.3) because the synset is linked to the concept of '*animal*' via a number of intermediate superordinate concepts which include '*vertebrate*' and '*chordate*'.

   This often arises because of the requirement that CIDE only uses a limited defining vocabulary, the effect of which only permits very general concepts to be referenced when more specific ones are available. Since WordNet is not restricted in this way, its hierarchy of concepts tends to be deeper.

5.3.1.2 CIDE concept does not map to a SemNet concept

There are cases when a CIDE word sense does not correspond to any WordNet concept because the particular view of the former deviates too far from the latter. The situation can be identified by the lack of a common superordinate concept. Consider the CIDE definitions:

party$_n$ — a social *event* where a group of people meet to talk, eat, drink, dance etc., often in order to celebrate a special occasion

chew $_v$ — to *crush* food into smaller, softer pieces with the teeth so that it is easier to swallow

crush $_v$ — to *press* something very hard so that it is broken or its shape is destroyed

Figure 5.4 shows the intuitively closest WordNet concepts to the first two CIDE definitions above.

Discrepancies arise because of the imprecise nature of language. WordNet views a '*party*' as the group of people who participate in an event (gathering for pleasure) while CIDE views a party as the event itself. A similar problem exists in the different views of the verb '*chew*'. While WordNet regards the *chewing* of an object as the process of *fragmenting* it, CIDE allows the possibility that *chewing* may simply change the shape (shown in the definition of '*crush*') of the object.

It is clear that the context will dictate the particular view that is taken, and if this is not the case, then these fine distinctions between word senses are unimportant. The latter view seems more plausible given that lexicographers cannot agree on the precise meaning. For this task, the CIDE and SemNet concepts above must be treated as different senses of a word given that both views are equally plausible. Hence new SemNet nodes corresponding to the CIDE meanings of '*party*' and '*chew*' need to be created. While these newly created nodes need to be given the correct grammatical information, they do not need to be linked into the correct place in the SemNet hierarchy. This can be done later by analysis of the definition of each word illustrated in the next chapter.

5.3.1.3 CIDE concept maps to multiple SemNet concepts

It was noted previously that the granularity of CIDE word senses is far coarser than that in WordNet. This is achieved by the use of the conjunction '*or*' within a definition. For example, consider the definitions of the following words taken from the CIDE defining vocabulary:

```
equip, appoint, fit, kit out, kit up, kit, fit out -- (supply with equipment)
 => supply, provide, render, furnish
  => give -- (transfer possession of something concrete or abstract)
   => transfer -- (cause to change ownership)


bird
 => vertebrate, craniate
  => chordate
   => animal, animate being, beast, creature, fauna
    => life form, organism, being, living thing -- (any living entity)
     => entity -- (something having concrete existence; living or nonliving)


dish -- (a piece of dishware normally used for holding or serving food)
 => crockery, dishware -- (eating and serving dishes collectively)
  => tableware -- (utensils for use at the table)
   => utensil -- (for practical use esp. in a household)
    => implement -- (a piece of equipment or tool used to effect an end)
     => instrumentality
      => artifact, article, artefact -- (a man-made object)
       => object, inanimate object, physical object -- (a nonliving entity)
        => entity -- (something having concrete existence; living or nonliving)
```

Figure 5.3: WordNet hierarchies rooted at concepts which correspond to CIDE definition.

```
party -- (a social gathering for pleasure)
 => gathering, assemblage, assembly, body, confluence
  => social group -- (people sharing some social relation)
   => people -- (any persons collectively; "old people")
    => group, grouping


chew, masticate, jaw -- ("He jawed his bubble gum")
 => grate, grind
  => break up, fragment, break into fragments
   => break, separate, split up, fall apart, come apart
    => change integrity -- (change in physical make up)
     => change, undergo a change, become different
```

Figure 5.4: Various WordNet taxonomies rooted at concepts which do not correspond to CIDE definitions.

connect $_v$ — to join or be joined with something else • He connected the printer to the computer • The lead connects the printer to the computer

degree $_n$ — a *course* of study at a college or university, or the *qualification* given to a student who has completed this • What degree did you do? • She's got a degree in Physics from Oxford

The examples accompanying each definition illustrate the different contexts in which the finer word senses (those on either side of the underlined '*or*') occur, *e.g.* `connect-as-activity` (a human connecting objects together) vs. `connect-as-span` (an inanimate entity connecting objects together) respectively. WordNet senses are finer grained and generally have separate concepts representing each of these senses.

```
connect, interconnect, interlink, link, connect together, communicate

connect, link, join, unite -- (act as a link between, be a connector)

academic degree, degree
 => qualification
  => fitness, fittingness
   => suitability, suitableness
    => quality
     => attribute -- (abstraction belonging to or characteristic of an entity)
      => abstraction
```

Figure 5.5: Various WordNet taxonomies rooted at concepts which correspond to parts of CIDE definitions.

The first and second synsets in Figure 5.5 correspond to the `connect-as-activity` and `connect-as-span` cases respectively. Hence the CIDE meaning of the verb '*connect*' is mapped to the two separate WordNet concepts.

Similarly the CIDE meaning of '*degree*' would map to two meanings in WordNet: `degree-as-activity` and `degree-as-attribute` respectively. However, WordNet only contains the `degree-as-attribute` concept illustrated in Figure 5.5. Therefore a new SemNet node (corresponding to `degree-as-activity`) is created in the manner outlined in the previous section.

5.3.2 The Extraction of Semantic Knowledge

The aim of the extraction process is to utilise LOLITA's current language processing capabilities to analyse and consequently extract the semantic relationships encoded within the text of a CIDE definition.

5.3.2.1 Transforming the Definition

Two related problems exist if LOLITA is given the definition of a headword as input text:

the language of definitions — LOLITA expects to analyse texts (such as newspaper articles) which consist of one or more sentences. However, a definition contained in CIDE is a typically a noun or verb phrase which is only a fragment of a sentence.

the association between headword and definition — some method of associating the headword which is being defined to the text of the definition is required.

Both of the problems are solved by appealing to a property of the structure of definitions that the head of the first noun or verb phrase is generally the IS-A related genus term of the headword. This makes it possible to transform a CIDE entry into a sentence which LOLITA is able to analyse as normal text. The transformations of CIDE entries (to the left of ⇝) into sentences (to the right of ⇝) which can be analysed by LOLITA, are as follows:

[noun — definition] ⇝ "**each** noun **is** definition"
[verb$_I$ — definition] ⇝ "**to** verb$_I$ **is** definition"
[verb$_T$ — definition] ⇝ "**to** verb$_T$ **something is** definition"

where verb$_I$ and verb$_T$ represent entries for transitive and intransitive verbs respectively[7]. The keyword '*each*' is used in the noun case because although definitions are intended to capture general classes of objects, they are written so that each one refers an arbitrary individual of that class. In addition, it should be noted that these templates are intended to capture the majority of cases. Some types of definitions will cause problems, *e.g.* those that start with an example. Currently there is no specific machinery to deal with these types of non-standard definitions.

The translation process is illustrated on the definitions below:

jack $_n$ — a piece of equipment which can be opened slowly to allow heavy weights to be raised

$\leadsto$

"each jack is a piece of equipment which can be opened slowly to allow heavy weights to be raised"

carp$_I$ — to complain continually about unimportant matters

$\leadsto$

"to carp is to complain continually about unimportant matters"

banish$_T$ — to send someone away from their country and forbid them to come back

$\leadsto$

"to banish something is to send someone away from their country and forbid them to come back"

The effect of the insertion of the word "*is*" between headword and definition ensures that LOLITA will interpret the relation between the headword and the head of the first NP of the definition to be an IS-A. The insertion of an indefinite pronoun

[7] verbs that are marked as object optional (tagged as *(obj)*) in CIDE are viewed as a subclass of transitive verbs.

"*something*" in the case of a transitive verb tells LOLITA about the existence of an entity which is the object of the verb being defined. The specific class of objects which will form the referent of the indefinite pronoun are inferable at a later stage because of the existence of the hierarchical relationship between the headword and the first verb phrase (*e.g.* between "*banish*" and "*send someone away*") of the definition (see Section 6.13 for further details).

5.3.2.2 Ambiguity in Dictionary Definitions

In the normal mode of operation, LOLITA attempts to interpret the meaning of text by making explicit the various entities and semantic relationships which are contained within it. The interpretation process involves resolving a number of ambiguities. From our experience the following types of ambiguity are common when processing dictionary definitions:

Structural Ambiguity gives rise to multiple parse trees. There are a number of types of structural ambiguity (see [Hirst, 1987]) the most important of which are:

Prepositional Phrase (PP) Attachment — definitions typically consist of a noun or verb phrase possibly, followed by a number of relative clauses and prepositional phrases. The ambiguity of PP attachment is illustrated by considering the following CIDE definition with PPs underlined:

hangman$_n$ a person whose job is to operate the device which kills criminals by hanging them from a rope by their necks

Assuming the rule that the sequential ordering of PP attachments respects the order in which the PPs occur in the text (in other words there are no crossed attachments in the parse tree), and that attachments can be made to verbs[8] or NPs preceding the PP in the sentence (*i.e.* the pos-

[8] The distinction between the attachment of PPs to verbs or to verb phrases is not considered

sibilities for attachment are, '*person*', '*job*', '*operate*', '*device*' and '*kill*') there still exist 35 different combinations of attachments of the three PPs.

Referential Ambiguity arises from the use of anaphoric expressions (*e.g.* amongst others, pronouns, the temporal and locative expressions '*when*' and '*where*', and so on) in the text. They are illustrated in the following examples:

coerce$_v$ — to persuade someone forcefully to do something which they are unwilling to do

drive$_n$ — a planned effort to achieve something

imprint$_n$ — the name of a publisher as it appears on a particular set of books

abattoir$_n$ — a place where animals are killed for their meat

hideaway$_n$ — a place where someone goes when they want to relax and get away from their usual surroundings

The current solution, in LOLITA, to finding the correct referents for a class of anaphora is based upon a number of heuristics detailed in [Urbanowicz, 1999].

In a recent NLP evaluation (MUC-7 [DAR, 1998]) which consisted of resolving a class of anaphora in unseen newspaper articles from the Wall Street Journal (WSJ), LOLITA scored 46.9% recall and 57% precision[9]. The complexity of the task of anaphora resolution is illustrated by the following definitions:

banish$_v$ to send someone away from their country and forbid them to come back

candle$_n$ a usually cylindrical piece of wax with a piece of string in the middle of it which produces light as it slowly burns

in much of the research in the field [Franz, 1996]. The distinction is addressed in Section 6.9.

[9] The number of correct co-reference links made by the system divided by the number of all links in an answer key constitutes the *Recall* measure.

The number of correct links made by the system divided by the number of all links made by the system (i.e. the correct links plus the spurious links) constitutes the *Precision* measure.

thalidomide$_n$ a drug which was once used to help people relax or sleep, and which was found to cause damage to babies inside the womb, esp. by stopping the development of their arms and legs, when it was taken by their mothers

The first case is difficult to solve because the correct antecedent of the plural '*them*' is the singular '*someone*'. The relaxing of feature matching rules will result in a larger number of potential referents.

In the definition of '*candle*' the first pronoun '*it*' could be resolved to a number of different preceding NPs: piece of string, string, wax, piece of wax, *etc.* The latter '*it*' contains the same plausible referents as the former augmented with '*light*'. A naive strategy such as preferring the last mentioned NP would resolve the former '*it*' to '*a piece of string*', and the latter one to '*light*'.

In the final example, plausible candidates for the possessive determiner '*their*' are: arms, legs, arms and legs, babies and humans.

There are two types of rules which can be used to resolve the anaphora correctly:

structural rules — use information derived from the grammatical structure of the sentence together with a set of heuristic rules to rank and subsequently to select the most plausible candidate. Examples of these rules would be to prefer referents that are in the dominating position of a clause, prefer referents closest to the anaphor, prefer the referents of other pronouns in adjacent clauses, and so on. Hobbs [Hobbs, 1986] describes a structural algorithm to resolve pronouns.

semantic rules — try to use semantic knowledge together with reasoning machinery to understand the meaning of the text and consequently resolve the anaphora occurring within it. For example, Schank's script theory [Schank and Abelson, 1977] attempts to resolves anaphora, as a by-product of the understanding process.

The semantic knowledge needed to resolve many anaphora is precisely the type of knowledge that is being acquired. For example, to correctly resolve the possessive determiner '*their*', in the compound '*their babies*', requires the knowledge that, from the set of all NP's in the text, only *people* or *babies* have mothers. In addition, *babies* are more closely associated with a *mother*.

Most NLP systems (including LOLITA) will attempt to make use of both types of knowledge to find the correct antecedent of the anaphoric expression (see [Urbanowicz, 1999]).

Word Sense Ambiguity arises because some words which occur in the text of definitions are used in more than a single sense. Two types of word sense ambiguities (WSA), that need to be resolved, are:

content words — which are nouns, verbs, adjectives and adverbs whose meanings are restricted by the defining vocabulary. Hence the WSD problem using CIDE is relatively smaller than in dictionaries which have no defining vocabulary.

function words — which can only be disambiguated in the context of the constituents which they relate, *e.g.* prepositions represent a binary relation between the object being modified, and the prepositional object. The senses of prepositions have not been restricted in the CIDE defining vocabulary.

It is clear that any attempt to extract useful knowledge from definitions needs to be able to resolve the types of ambiguity mentioned above.

5.3.3 Knowledge Integration

The semantic knowledge that is to be extracted from the definitions needs to be integrated into SemNet in a consistent way. In addition, the relationships must be represented in a form which will allow inferences to be drawn in an efficient manner.

There are two classes of badly written definitions:

- those in which an anomaly can be detected automatically. For example, in the definition:

  bend $_v$ — to cause to curve

  it is incorrect to view the relationship 'bend IS-A cause'. The definition is intended to capture that bending is the event of "doing something which results in the curvature of the object".

- those in which an anomaly can only be detected by complex reasoning about the definition, *e.g.*,

  bite $_v$ — to use your teeth to cut into something

  To illustrate that this definition is badly worded, consider the following chain of reasoning. If '*bite*' IS-A '*use*', then event, "X bite Y" implies the event, "X use Y" (in much the same way that, if the verb "*fry* IS-A *cook*" then the event, "John fried eggs", implies the event, "John cooked eggs"). Now, the event, "I bit my tongue", does not imply "I used my tongue", except in a very counter-intuitive definition of the verb '*use*'. On the other hand, it could imply "I cut my tongue" or at least "I could have cut my tongue" depending on the intensity of the bite. Consequently, a better definition of the verb '*bite*' would be:

  bite $_v$ — to cut into something using your teeth

The detection of badly worded definitions is important because the incorrect semantic relationships would be inherited to many different levels of the hierarchy as mentioned in Section 2.3.

5.4 Summary

This chapter illustrated that the market currently targeted for dictionaries makes them less than ideal for NLP purposes. Several problems were explicitly considered. A strategy for solving these problems was presented. It involved integrating the knowledge in CIDE with an existing lexical resource, SemNet.

A framework for the acquisition process was introduced. The framework requires a one-off mapping between the CIDE defining vocabulary and SemNet, followed by an analysis and knowledge integration cycle for each of the definitions in CIDE. Finally, the range of ambiguities which would need to be resolved during each cycle was outlined.

Chapter 6

A Semi-Automated Approach to Lexical Acquisition

In the previous chapter it was illustrated that the extraction of semantic knowledge from dictionary definitions requires solving complex NLP tasks such as attachment problems, anaphora resolution, and word-sense disambiguation. The strategy in this research is to use a semi-automated approach in which the LOLITA system will process as much of the definition as possible and present the points of ambiguity to the operator[1] who will subsequently disambiguate them. This approach has been chosen for the following reasons:

1. The solution of many of the tasks, to a level of accuracy comparable to that achievable by humans, is far beyond the state-of-the-art of current NLP systems. For example, the best parsers for LDOCE definitions are approximately 90% accurate [Wilks, 1996], with the successful parses being subjected to further errors during semantic analysis. The major problem is that there is no immediate way of detecting when errors in parsing or analysis have occurred. Even worse, their effect within a taxonomy of concepts is multiplied because the resulting structures are *inherited* to other levels.

[1] the word 'operator' is used to refer to the human disambiguator.

2. Humans lack the consistency that computers possess when formalising complex logical constructs, *e.g.* the task of representing semantic relationships, by hand, in an unambiguous KRL (*e.g.* semantic network, conceptual graph, first-order logic, *etc.*) would cause difficulty because of the complexities caused by the sheer number and inter-relatedness of the semantic relationships.

On the other hand it is unrealistic to expect that linguistic experts will manually process an entire dictionary of definitions (a problem of time and money in addition to those above). Therefore an additional aim of the acquisition task is to ensure that people with a minimal amount of linguistic training are able to understand and carry out the entry process.

This requirement has many consequences in the way questions are posed to the operator. These consequences are highlighted at the appropriate points in the remainder of this chapter.

The setup for the acquisition process is shown in Figure 6.1. It shows three major components:

A Machine Readable Dictionary Database is used (in this thesis) to refer to an MRD which allows easy access to the particular fields of information associated with each dictionary entry, *e.g.* headword, phonological information, definition, examples, *etc.* The production of an MRDD often involves the processing of typeset dictionary files. This task is not necessarily straightforward [Alshawi, 1989].

The LOLITA System is the key component in the acquisition process. It is given definitions of headwords which are processed (following the transformations described in Section 5.3.2) as input text. The points of ambiguity are extracted from the analysis process and output as a series of questions. The semantic relationships which are extracted from the definition are represented as knowledge structures and are stored in SemNet.

The User Interface is responsible for co-ordination of the acquisition process. It gets dictionary entries from the MRDD, and passes the definitions to LOLITA for processing. Subsequently the interface co-ordinates the disambiguation process by receiving disambiguation questions from LOLITA, presenting them to the operator, and communicating the answers back to the LOLITA system.

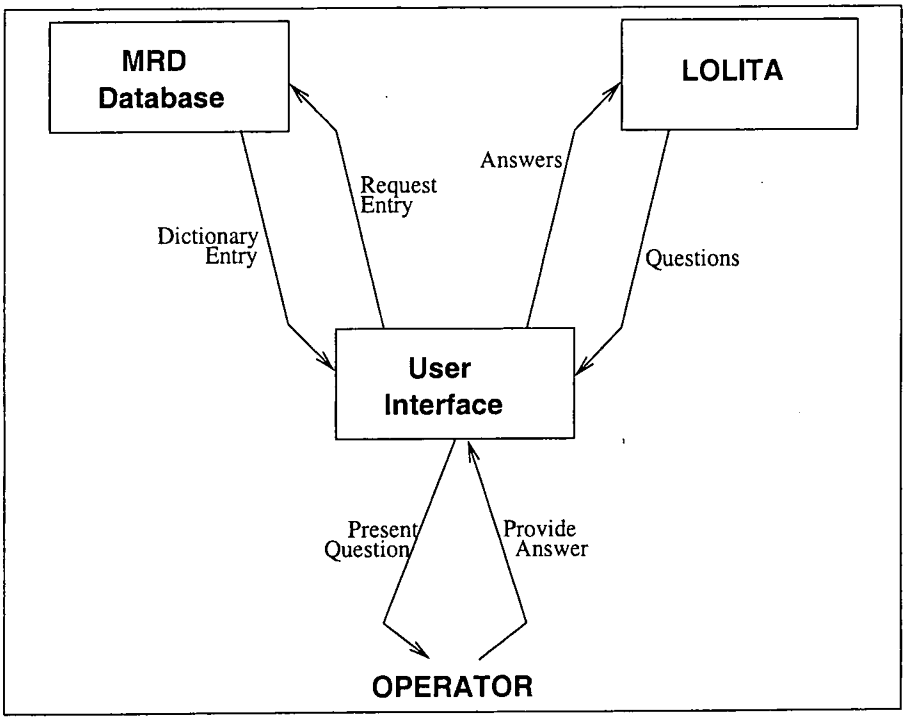


Figure 6.1: The setup for the lexical acquisition process.

The next sections introduce details of the interactive analysis process between LOLITA and the operator. This interaction is divided into a number of stages which are shown in Figure 6.2. For each stage, the sections below introduce the type of knowledge that is extracted from the definition, and the assistance that needs to be provided by the operator.

6.1 Picking the SemNet Meaning

For each CIDE definition to be entered, the operator will need to find the SemNet node which represents the concept corresponding to the headword being defined.

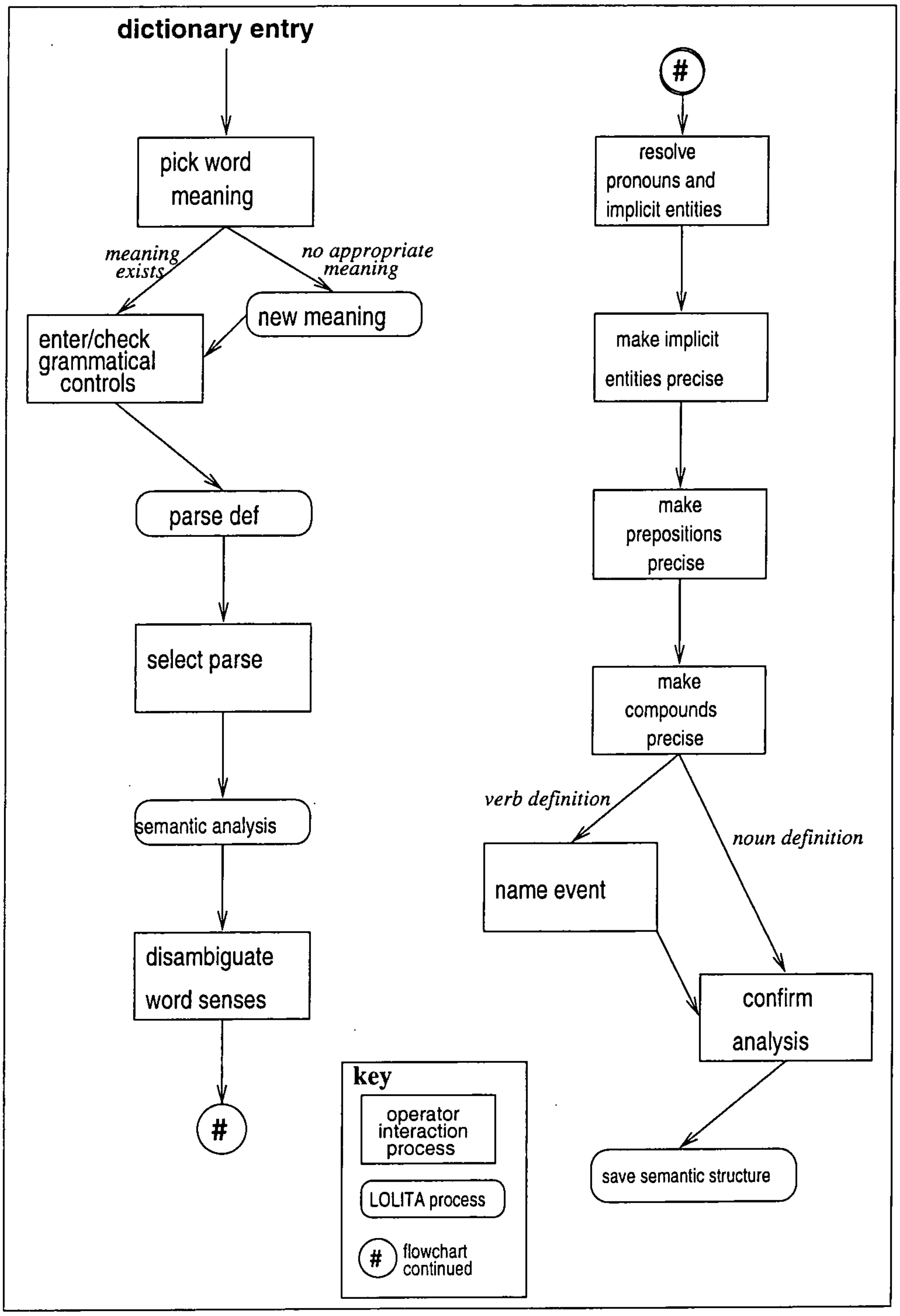


Figure 6.2: A flowchart showing the major stages in the acquisition process.

The following information will aid this mapping process:

1. Knowledge derived from the original WordNet source. This information includes the original WordNet glosses, together with any synonym/antonym and hyponym/hypernym links. As an example, consider the CIDE definition:

   crack $_n$ — a pure and powerful form of the drug cocaine

   for which the operator will be expected to pick the corresponding SemNet meaning, amongst those shown in Figure 6.3.

   The glosses following each node are the most informative source of information in this case. It is easy to see that the last meaning of '*crack*' is compatible with the definition. The first three meanings in Figure 6.3 do not have a descriptive gloss[2]. However, their meanings are intuitively clear from the list of associated concepts, the most informative of which identify the various senses as `crack-as-cranny`, `crack-as-slit` and `crack-as-wisecrack` respectively.

2. Knowledge which has been subsequently added to each concept as control values (see Section 4.2). The most informative of these is the *family* of the concept.

There may be a number of definitions which have no corresponding mapping to SemNet. If this is the case then new SemNet nodes will need to be created for them. These newly created nodes will be linked to existing SemNet concepts, when their definitions are processed by LOLITA.

[2] [Miller, 1998] reports that 60% of WordNet 1.4 synsets have a descriptive gloss.

```
Cracks (=> Depressions
       = Scissures , Fissures , Crevices , Crannies , Chaps)
 family: shape
 emotional value: indifferent
 level of language: common level


Cracks  (=> Apertures
        = Slits , Breaches , Rifts , Clefts
       <= Fractures , Crevices , Chasms)
 family: inanimate
 emotional value: indifferent
 level of language: common level


Cracks (=> Comments
       = Wisecracks , Sallies , Quips)
 family: communication
 emotional value: indifferent
 level of language: common level


Cracks (=> Defects
       = Checks , Chips)
 family: attributes
 emotional value: indifferent
 level of language: common level

 a mark left after a small piece has been chopped or broken off


Cracks (=> Openings
       = Gaps
      <= Blanks , Breaches)
 family: inanimate manmade
 emotional value: indifferent
 level of language: common level

 a narrow opening; "he opened the window a crack"


Cracks (=> Cocaines)
 family: inanimate manmade
 emotional value: indifferent
 level of language: common level

 a purified and potent form of cocaine that is smoked rather than snorted
```

Figure 6.3: The different SemNet meanings for the entity ‘crack’.

6.2 Capturing Control Information

Although each CIDE entry contains explicit grammatical information useful to a parser, there is a vast amount of grammatical knowledge that is not explicitly encoded within the definition of an entry. This information, which is generally about verbs, can be extracted from the usage examples which accompany dictionary definitions. Examples of grammatical knowledge and its use in an NLP system is illustrated below:

reflexive verbs are a class of transitive verbs in which the object and the subject are always the same agent, *e.g. perjure* is defined as:

perjure $_v$ — to cause yourself to tell a lie in a law court, after promising formally to tell the truth

This class of reflexive verbs are identified in CIDE by the personal pronoun in the definition. However there is a superset of transitive implicit verbs which are reflexive, but only in the intransitive case, *e.g.*

dress $_v$ — to put clothes on someone else, esp. a child or yourself

shave $_v$ — to remove hair from the body, esp. a man's face by cutting it close to the skin with a razor or shaver, so that the skin feels smooth

then the phrases "*John shaved*" and "*John dressed*" imply that "*John shaved himself*" and "*John dressed himself*" respectively. These types of inferences (about the actors involved in events) are essential for NLP systems. Since the latter class of verbs are not explicitly marked in CIDE, the information needs to be provided by the operator.

personal verbs are transitive verbs which denote private states which can only be subjectively verified. They hold in the mind of the speaker. There are of several classes of personal verbs (see [Quirk *et al.*, 1985]), examples of which are:

perception *e.g.* hear, feel, see

emotion/attitude *e.g.* intend, wish, want, like

Knowledge of this class of verbs is useful to an NLP system because they prohibit erroneous inferences. For example, consider the sentences, "*Mary died in the accident*", and "*John wished Mary was dead*". While the former sentence necessarily implies the occurrence of the event of *Mary's death*, the latter sentence does not.

symmetric verbs are transitive implicit verbs which have the same meaning if their subject and object are reversed, for example,

touch $_v$ — (of two or more things) to be so close together that there is no space between; to be in contact

fight $_v$ — to use force against esp. another person or group of people

These verbs form a useful class because their intransitive versions are formed by combining the subject and object, *e.g.* "*she pushed the bookcases until they touched*", means *until the two bookcases touched each other.*

The full list of controls which are considered important for NLP purposes is listed in Appendix C.

6.3 Parsing the Definition

The next stage is for LOLITA to parse the definition. The parser builds a parse forest (of all possible parses) of the input and uses a set of penalties to order them according to syntactic and grammatical plausibility. This results in a list of sets of parses, each set occurring at a particular penalty level. Even choosing between parses at the top penalty level presents a number of problems:

1. structural ambiguity is prevalent in dictionary definitions. Definitions are written so that they typically consist of a noun or verb phrase followed by a number of relative and prepositional clauses. The resolution of structural ambiguity generally requires semantic knowledge.

2. understanding the interpretation represented by a parse tree requires an amount of expertise. For example, Figure 6.4 shows the two simplified parses[3] of the sentence:

   "Each abattoir is a place where animals are killed for their meat"

```
1
sen
detph
  det EACH
  comnoun ABATOIR
is_a
  is_a BE
  copula_detph
    undefinedDet A
    relcl
      moved_relcl
        comnoun PLACE
      prep_relclause
        prep_and_implicit2
          literal WHERE
         sen
           moved_obj2
             missing_det
               comnoun ANIMAL
           linkph
             auxBe BE
             auxphrase_advprepph
               verb_and_implicit
                 verb KILL
               prepp
                 prepNormMode FOR
                 poss_detph
                   possessiveDet THEIR
                   comnoun MEAT
```

```
2
princ
sen
  detph
    det EACH
    comnoun ABATOIR
  is_a
    is_a BE
    copula_detph
      undefinedDet A
      comnoun PLACE
subsen
  subconj WHERE
  sen
    moved_obj2
      missing_det
        comnoun ANIMAL
    linkph
      auxBe BE
      auxphrase_advprepph
        verb_and_implicit
          verb KILL
        prepp
          prepNormMode FOR
          poss_detph
            possessiveDet THEIR
            comnoun MEAT
```

Figure 6.4: The two parses trees produced by LOLITA for the definition of '*abattoir*'.

The problem with parse (2) is that the attachment of the WHERE is too far to the left. The interpretation is that the "*place where animals are killed*", and

[3] the parses are simplified because feature information has been removed.

the place where "*each abattoir is a place*", are the same. The attachment of WHERE in (2) makes sense in a sentence such as "*The police stopped pedestrians where the accident happened*", in which the place where "*the police stopped pedestrians*", and, "*the accident happened*" are the same. A parse is often best interpreted by finding a structurally similar sentence in which the construct is more coherent, in the manner shown.

It is unrealistic to expect untrained operators to select between a number of parse trees because of the difficulties mentioned above. However, the disambiguation process can be simplified because:

- although the interpretation of a parse tree requires a degree of expertise, our experience shows that humans find it easier to identify parses which violate the intuitive interpretation of a piece of text.
- multiple parses of a sentence exist because of structural ambiguity which results mainly from the interpretation of one or more of three types of construct:

  1. prepositional phrases
  2. relative clauses
  3. conjunctions

  This suggests that a certain amount of detail from the parse tree can be hidden from the operator. A better strategy is to extract points of uncertainty in the parse trees and present them in a clear way.

A strategy in which an operator can reject incorrect parses has been developed. The parses are presented in a simplified linear form with brackets being used to illustrate the differences between them.

Each bracketed unit is composed of a single *main item* and a number of remaining *subordinate items*. The main item is the dominating term in the bracket. Consider the following examples:

1. each hangman is a person whose job is to operate (the device which kills criminals by hanging them from a rope by their necks)
2. to vomit something is to (empty the contents of the stomach) through the mouth
3. (each abattoir is a place) where animals are killed for their meat

For interpretation (1), the entity '*the device*' is the main item (or head) of the bracketed segment, and the remaining entities, the verbs '*kill*' and '*hang*', together with the nouns '*the criminals*', '*a rope*' and '*their necks*' are the subordinate items. In (2) and (3), the verbs '*empty*' and '*is*' are the main items, respectively.

This information should be used together with a number of rules given below in order to make a decision regarding the validity of a parse. An operator should:

- reject parses in which the concept formed by the bracketed unit does not exist in an intuitive understanding of the input.

  This is the simplest rule which can be used to identify parses which have incorrect PP attachments. For example, in the interpretations[4]:

  1* to vomit something is to empty the contents of the (stomach through the mouth)

  2. to vomit something is to empty the contents of (the stomach) through the mouth

  the entity in the first bracket '*stomach through the mouth*' is inconsistent with an intuitive understanding of the sentence, since no such entity exists. Consequently, interpretation (1) can be rejected.

- reject parses which violate knowledge concerning the modification of entities.

[4] incorrect interpretations are specified by a *

The operator can make use of the rule that a PP or relative clause following a bracketed segment should be assumed to be modifying the main item in the segment, only if it is of the correct type[5].

If the phrase immediately following the bracketed unit should modify a subordinate entity within the unit, then the interpretation can be rejected.

The section below illustrates how the rules above are applied in the disambiguation process.

6.3.1 An Example of Parse Tree Selection

The interpretation process for parse tree selection is best illustrated by considering the transformed version of the definition of '*hangman*':

> "each hangman is a person whose job is to operate the device which kills criminals by hanging them from a rope by their necks"

The parse trees can be represented in a linear form by simply bracketing each subtree in the parse. Bracketed versions of the 9 parses produced by LOLITA are:

1. each hangman is a person whose job is to operate (the (device (which (kills criminals ((by (hanging them from a rope)) by their necks)))))
2. each hangman is a person whose job is to operate (the (device (which (kills criminals ((by hanging them) (from a rope by their necks))))))
3. each hangman is a person whose job is to operate (the (device (which (kills criminals (by (hanging them (from a rope by their necks)))))))

[5] Some types of construct can be thought to modify (or have as an argument) other types of grammatical constructs. Prepositional phrases modify nouns or verbs, and cannot modify other grammatical constructs, *e.g.* an infinitive phrase or another PP. Conjunctions modify verbs (or verb phrases), nouns and sentences, and relative pronouns only modify nouns and clauses.

4. each hangman is a person whose job is to (operate (the (device (which (kills criminals ((by hanging them) from a rope)))))) by their necks

5. each hangman is a person whose job is to (operate (the (device (which (kills criminals (by (hanging them from a rope))))))) by their necks

6. each hangman is a person whose job is to (operate (the (device (which (kills criminals (by hanging them)))))) (from a rope by their necks)

7. each hangman is a person whose job is to (operate (the (device (which kills criminals)))) ((by (hanging them from a rope)) by their necks)

8. each hangman is a person whose job is to (operate (the (device (which kills criminals)))) ((by hanging them) (from a rope by their necks))

9. each hangman is a person whose job is to (operate (the (device (which kills criminals)))) (by (hanging them (from a rope by their necks)))

The linear form is clearer because of the ease of identifying a group of words which has been parsed as a unit, or, more informatively, a group of words which has not been considered to form a unit, *e.g.* in interpretations (4) and (5), the PPs "*from a rope*" and "*by their necks*" are separated. The hypothesis is that humans with limited linguistic expertise could identify that both PPs should form some sort of unit because they modify the same object, *i.e.* the event of *hanging.*

This illustrates the basic strategy which is being adopted. The idea is that incorrect parses of the original sentence[6] can be easily identified by finding a group of words which do not (or cannot) form a unit in the bracketed version, when they clearly should.

Showing the full bracketed version to the operator and asking them to reject incorrect ones may be a little daunting because some parses may be deeply nested. The problem is eliminated by asking the operator to consider only a single pair

[6] a parse is considered incorrect if any information (or structure), contained within it, violates an intuitive understanding of the original text.

of brackets at a time. The strategy adopted here is to show each bracket in turn using a left-most outer-most algorithm. Instead of presenting the operator with the 9 interpretations above, they are initially asked to reject violations of their understanding with:

a. each hangman is a person whose job is to operate (the device which kills criminals by hanging them from a rope by their necks)

b*. each hangman is a person whose job is to (operate the device which kills criminals) by hanging them from a rope by their necks

c*. each hangman is a person whose job is to (operate the device which kills criminals by hanging them) from a rope by their necks

d*. each hangman is a person whose job is to (operate the device which kills criminals by hanging them from a rope) by their necks

Option (a) corresponds to the fully bracketed versions (1—3), option (b) to (7—9), option (c) to (5), and option (d) to versions (4—5).

Given that the correct interpretation of the sentence is that the verb '*kill*' is done '*by hanging*', and that, the '*hanging*' is done '*from a rope*' and '*by their necks*', then the correct answer is to reject interpretations (b),(c) and (d).

The reasoning for this is that, in (b), the PP '*by hanging them*' cannot modify the verb '*kill*', in (c) the phrase '*from a rope*' cannot modify the '*hanging*', and in (d) the phrase '*by their necks*' cannot modify the "*hanging*". In each case the particular verbs cannot be modified because they are not the main items within each of the bracketed units respectively.

If the interpretations (b—d) are correctly rejected then the following options remain:

e. each hangman is a person whose job is to operate the device which kills criminals by (hanging them from a rope by their necks) f*. each hangman is a person whose job is to operate the device which kills criminals (by hanging them) from a rope by their necks

g*. each hangman is a person whose job is to operate the device which kills criminals (by hanging them from a rope) by their necks

these correspond to the next left-most outer-most difference in the fully bracketed versions of the original parses (3), (2) and (1) respectively. The correct answer at this point is to reject interpretations (f) and (g) because the phrases '*from a rope*' and '*by their necks*' cannot modify the event of *hanging* in each interpretation respectively.

The result of the two interactions above would result in the effective selection of the interpretation (1), which contains the correct attachment of PPs in the original definition.

6.3.2 A Strategy of Rejecting Parses

A process by which the operator rejects parses, by using their knowledge about which interpretations violate their intuitive understanding of the text, does not necessarily conclude with the single remaining correct parse. The following alternative situations can result:

all parses rejected — there will be cases in which LOLITA may not produce the correct parse. This may be caused by a number of reasons: grammar coverage (the grammar may not be able to handle a particular construct), bad knowledge (there may be errors in the lexical data), *etc.* In these cases the system will simply cease further processing of the particular definition.

more than one parse remains — this is the case in which more than a single correct parse remains after the filtering process outlined in the previous section. For example, in the analysis of:

> "a jack is a piece of equipment which can be opened slowly to allow heavy weights to be raised"

the operator is asked to reject incorrect interpretations from:

a. each jack is a piece of (equipment which can be opened slowly to allow heavy weights to be raised)

b. each jack is a (piece of equipment) which can be opened slowly to allow heavy weights to be raised

Neither of the interpretations above violate any rule given in the previous section. Since no remaining ambiguities exist, then one of the interpretations will be picked at random. These cases reflect structural ambiguities which map to a single meaning when they are semantically interpreted.

incorrect parse selected — the last section implicitly assumed that the parses, which are not rejected by the end of the analysis process, represent the correct interpretation of the input.

However, the process outlined above does not guarantee this because it only enables the operator to consider differences between the parses at various points. One can imagine a situation where all the parses produced are structurally incorrect in the same way. Consequently no question would be asked about the construct. The simplest example is the case where only a single, incorrect parse is produced and is automatically assumed to be the correct one.

The approach presented above in which humans are able to reject interpretations has a number of benefits compared to a process in which the operator examines and locates the correct parse tree. Firstly, only one point of violation needs to be found to reject the parse and, secondly, it provides a convenient way to ignore the details which are common to all interpretations.

The price of this approach is that the end product may still result in an incorrect interpretation being chosen. It is the fact that such cases can be detected and eliminated by further processing which makes the approach feasible.

6.4 Semantic Analysis

The role of LOLITA's semantic analysis module is to map a parse tree to an internal representation of nodes and links between the concepts which occur in the text. These nodes and links are stored in SemNet (see Section 4.2), the KR of LOLITA. Semantic analysis makes no decisions regarding ambiguity (*e.g.* word sense ambiguity, pronoun ambiguity, *etc.*) in the original parse. This ambiguity is preserved within the SemNet representation of the parse. Figure 6.5 shows the fragment of SemNet resulting from semantic analysis of the first parse tree (the definition of the noun '*abattoir*') in Figure 6.4.

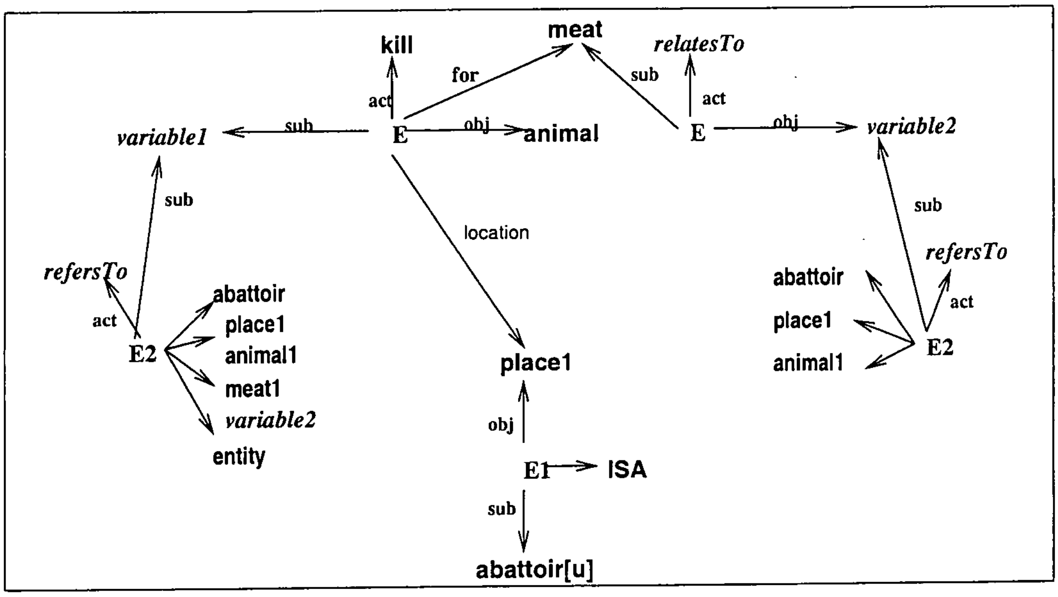


Figure 6.5: A simplified fragment of SemNet which results from semantic analysis of the parsed definition of the noun, '*abattoir*'.

The event E1 represents the assertion of the original sentence, "*each abattoir is a place...*", since it is the outermost event in the input. Events labelled E2 are internal events[7] which are used to represent word sense or referential ambiguity in the input. The two internal events in Figure 6.5 which have the action `refers_to`, are used to represent possible referents for the implicit entity which is the subject of the verb phrase '*use meat*' and the pronoun '*their*' respectively in the input. Although no word sense ambiguity exists in this case (due to the use of the CIDE defining vocabulary), it would be represented in much the same way, *i.e.* with internal events representing each of the possible senses.

In the normal mode of operation, the semantic analysis phase of LOLITA is followed by pragmatic analysis (see Figure 4.1) which aims to resolve the ambiguity (represented by the internal events) in the semantic representation of the input. It applies a set of heuristic rules of the form, "*prefer last mentioned entity*", and, "*prefer a subject over an object*", in order to rank the plausibility of each referent of a pronoun (see [Urbanowicz, 1999] for a detailed description). The most plausible referent is found and recorded in SemNet.

The approach in this research is to replace the heuristic rules in the pragmatics module of LOLITA with a semi-automated approach in which an operator will resolve the ambiguities which are present in the text. The two stages of operator intervention in the normal course of analysis are illustrated in Figure 6.6.

The next sections outline the various sources of ambiguity which can arise and the way in which they are solved.

6.5 Word Sense Disambiguation

Although the problem of WSA is reduced by restricting the possible senses of words to those in the CIDE defining vocabulary, there are still many which have been used in more than a single sense.

[7] other types of internal events which are created during semantic analysis include the words used events which represent the input text corresponding to each dynamically created concept. See Section 4.3.3 for further details.

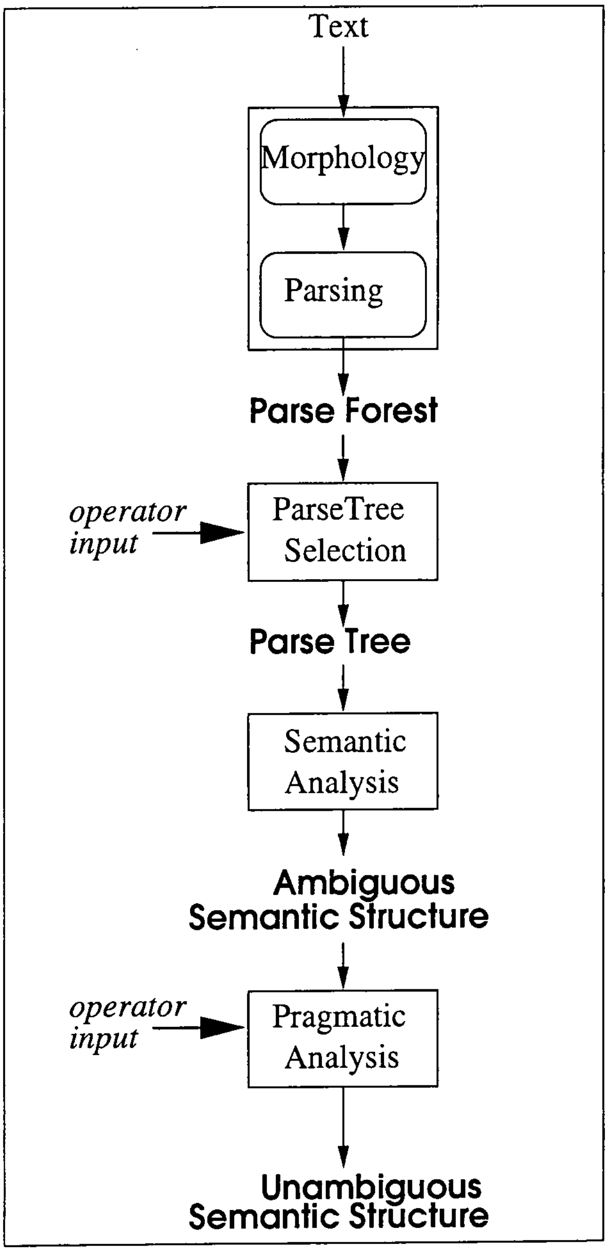


Figure 6.6: The core components of LOLITA showing intermediate structures and illustrating the location of operator intervention.

The problem is to present the information to the operator in a simple form, from which they are able to select the correct sense of the ambiguous word.

The most natural way is to simply highlight the ambiguous word in the input definition and let the operator select between the various senses which are presented as SemNet nodes in the format shown in Figure 6.3. For example the definition of the noun, '*hide*' in Section 6.1 contains ambiguous senses for the words '*make*' and '*thick*'. Figure 6.7 shows the type of question that the operator is expected to answer.

A problem of presenting word senses, in the form shown in Figure 6.3 mentioned earlier, was that not all WordNet concepts have explanatory glosses. If any sense is missing, an explanatory gloss can be added during the mapping process illustrated in Section 5.3.1. The informal description will be derived from the original CIDE entry.

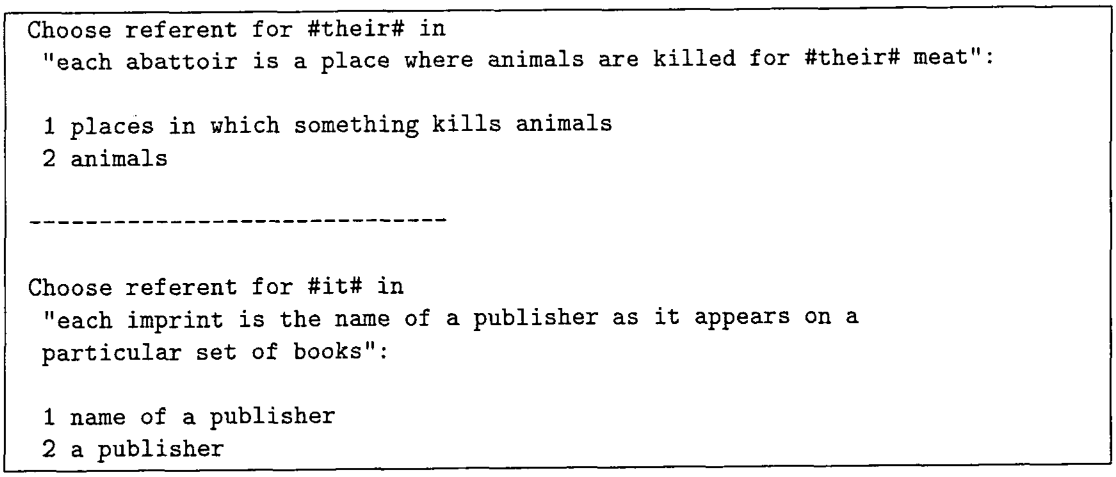
6.6 Resolving Pronouns

LOLITA's semantic analysis restricts the set of referents for a pronoun to those previously mentioned entities (in the text) with matching features. The problem is to present the question to the operator in a simple form. The original pronoun to which the question corresponds can be identified using the `words_used` events as shown in Section 4.3.3. The LOLITA generator can be used to describe any entities which are plausible referents for the pronoun. For example, the question the operator can expect to be asked in order to resolve the possessive determiner '*their*' in the definition of '*abattoir*' (given on page 97) is shown in Figure 6.8.

The referents should be described so that they are simple to identify in the original definition. In general, it is easy to identify the referent '*Animals*' than the longer form, '*Animals that are killed by something for something's meat*'.

```
Choose the meaning for the verb #make# in
 "something makes leather":

1)
 make (=> create
      = produce
     <= return , print , preassemble , reproduce , smelt , extrude ,
generate , generate , bootleg , laminate , elaborate , overproduce ,
machine ,  redo ,  breed)
 relation type: transitive

 create a product: "We produce more cars than we can sell"


2)
 make (=> accomplish
      =  carry , effect , do , execute, perform
     <=  exaggerate , complete , back-date , apply , enforce)
 emotional value: positive
 relation type: transitive

 carry into effect;  e.g., "make an effort"; "do research"; "carry too far"

---------------------

Choose the meaning for the adjective #thick# in the original definition:

1)
 Thick ()
 family: generic
 emotional value: indifferent
 level of language: common level

 relatively thick in consistency

2)
 Thick ()
 family: concrete
 emotional value: indifferent
 level of language: common level

 not thin; of relatively great extent from one surface to the opposite
 usu (sic) in the smallest solid dimension: "a thick board"; "a thick
 sandwich"; or of a specific thickness: "an inch thick"
```

Figure 6.7: The resolution of WSA during the analysis of the noun '*hide*'.



```
Choose referent for #their# in
 "each abattoir is a place where animals are killed for #their# meat":

 1 places in which something kills animals
 2 animals

------------------------------

Choose referent for #it# in
 "each imprint is the name of a publisher as it appears on a
 particular set of books":

 1 name of a publisher
 2 a publisher
```

Figure 6.8: The resolution of a pronoun, in the definitions of '*abattoir*' and '*imprint*' respectively.

6.7 Finding Referents for Implicit Entities

There are many entities which only exist implicitly within definitions and need to be made explicit because they reveal extra structure in the definition which can be used by NLP systems. There are two types of implicit entities:

1. entities which are implicit because of the nature of a definition — since dictionaries define verbs and not the events of the verb's occurrence, the participants of the events (*i.e.* the subject and object of the verb being defined) are implicit.

   Knowledge about the types of these participants is important because they are useful for a number of NLP tasks such as anaphora resolution and word sense disambiguation. Wilks [Wilks, 1975a][Wilks, 1975b] discusses an NLP system which is based entirely on knowledge provided by these selectional restrictions.

2. entities which do not occur in the main text of the definition but are inferable by humans — In the definitions:

   abattoir$_n$ — a place where animals are killed for their meat

   hangman$_n$ — a person whose job is to operate the device which kills criminals by hanging them from a rope by their necks the subject of the verb '*kill*' in the former, and the subjects of '*operating the device*' and '*hanging*' in the latter, are all implicit. The former group of agents (those which participate in the event of '*killing animal*') are termed '*butchers*' while in the latter both implicit entities are *the person* being defined.

The mechanism which makes the type of these entities precise provides a great deal of background and world knowledge.

The examples above illustrate a wide range of requirements for capturing the knowledge contained in implicit entities. The possible cases include:

1. the implicit entity may refer to one which has been explicitly mentioned in some other part of the definition, *e.g.* the definition of '*hangman*'.

2. the implicit entity may not refer to an explicitly mentioned entity, *e.g.* the definition of '*abattoir*'.

3. an implicit entity may be referential to another implicit entity in the definition, neither of which are referential to any explicit entity, *e.g.* in the definition of the noun '*hide*':

   hide $_n$ — the strong thick skin of an animal which is used for making leather

   the implicit agent which '*uses the skin*' and the one which '*makes the leather*' are identical, but are not explicitly mentioned in any other segment of the definition.

The knowledge identified above can be made explicit by having a two stage question answering process:

1. The first stage will make implicit entities in the definition explicit, and treat them as if they were neutral pronouns.

Consequently, the operator will be asked to pick a referent from the list of previously occurring entities in the definition. This will mean that implicits of type (1) and (3) in the list above will be dealt with.

In order to deal with the case that implicit entities do not necessarily have a explicit referent in the definition (case (2) above) the list of possible referents for each implicit entity is augmented with a default.

2. In the cases where the default referent is chosen to represent the implicit entity, further questions can be asked regarding its type, *e.g.* to extract the name '*butcher*' for the implicit entity in the definition of '*abattoir*'. The questions which capture this information are covered in Section 6.9.

Figure 6.9 shows the first stage of the questioning for an implicit entity in the definitions of '*abattoir*' and '*hangman*' respectively. It shows that the set of antecedents consist of explicitly mentioned referents together with a default case.

```
Choose referent for "Something that kills animals in places, abattoir":

 1 Places, abattoirs, in which something kills animals
 2 Animals
 3 Animals's meat
 4 some other entity

--------------------------------

Choose referent for "Something that hangs something":

 1 persons who operate a device
 2 person's job that is operate a device
 3 the device that kills criminals
 4 criminals that a device kills
 5 something that something hangs
 6 a rope
 7 criminals' necks
 8 some other entity
```

Figure 6.9: The extraction of knowledge about implicit entities which occur in the definitions of '*abattoir*' and '*hangman*' respectively.

The implicit entities are described by using the LOLITA generator because, unlike pronouns, they do not correspond to any particular word used in the original definition.

6.8 Making Entities Precise

This section describes procedures which gather knowledge about the types of entities that participate in the relationships which exist within definitions. There are two sources of these entities:

implicit entities are the entities identified in Section 6.7 which have no explicit referent in the definition.

indefinite pronouns such as '*someone*' or '*something*' are used within definitions to specify general classes of objects, *e.g.* in the verb definitions:

banish — to send someone away from their country ...
arrange — to put something in a particular order
coerce — to persuade someone forcefully to do something which...

they are used to specify the types of objects which participate in particular relationships. Knowledge of these selectional restrictions is important in tasks such as WSD. The restriction is determined by examining the definitions of the pronouns below:

something — an object, situation, quality or action which is not exactly known or stated
someone — a single person

Here, these indicate that it is people that are banished. In general, the more specific the restriction, the greater its utility. However, dictionaries tend to encode knowledge at a very coarse level (see Section 5.2) and this holds for selectional restrictions as well. Examples can be seen above: it is a subgroup of people (known as '*residents*') that are banished, and only '*objects*' that are arranged.

Often the type of entities which can participate in particular relationships will correspond to generic objects, such as *humans* (*e.g.* read, buy, record), *animals* (*e.g.* breed, breathe, move), or *inanimate objects* (*e.g.* break, flow, decompose). These groups (or families) provide convenient restrictions for the arguments of verbs, precisely because they exist at levels of abstractions which partition verbs into different conceptual groups.

Although the selectional restrictions of many verbs can be described with coarse grained types (*e.g. humans* or *animates*), they often have more specific names, *e.g.* "*murderers murder victims*", "*cooks cook food*", "*governments banish residents*". The operator should specify the restriction which is as precise as possible. An example of the type of question that is asked to the operator is shown in Figure 6.10.

```
Choose meaning for #someone# in
 "to banish something is to send #someone# away from their country and
  forbid them to come back"

 1 human          - any human or group of humans
 2 organisation - human organisations
 3 animal         - all kinds of animals, except humans
 4 animate        - all animates, including humans and non human creatures
 5 inanimate      - all inanimate entities, both organic and inorganic
 6 entity         - generic label for all objects
 7 event          - generic label for situations, states, phenomenons etc.
 8 other          - enter more specific entity
```

Figure 6.10: The acquisition of selectional restrictions

The operator has the choice of picking an extremely general class of entities, or is able to enter the name of a concept (*e.g.* '*resident*') by selecting the last option. If the name which is chosen is ambiguous (*i.e.* `resident-as-physician` or `resident-as-inhabitant`) then the operator will select the appropriate meaning through the procedure illustrated in Section 6.1.

The names which are provided by the operator are used to make inferences regarding the types of objects that **typically** participate in a particular relationship, *e.g.* upon being told that "`X banished Y`", it is reasonable to infer, in the absence of other knowledge, that `X` is a government and that `Y` is a resident. Used in this way, the violation of these restrictions by the many different 'fringe' meanings of *banish* does not pose a great problem to the logic of the underlying NLP system.

A further question needs to be asked to the operator concerning the status of a named entity because one of the two situations below:

1. the concept represents the largest group of entities which participate in the relationship.
2. the concept represents a larger group of entities than those that participate in the relationship. However, this group is smaller than the one represented by the coarse grained concepts such as '*humans*' and '*animate objects*'.

The distinction is best illustrated by the example, "*murderers murder victims*" where the status of '*murderers*' is of the former type, and '*victims*' of the latter. The concept of '*murderers*' represents the largest group of entities which *murder* because *all murderers* murder something. The concept of '*murderer*' can be said to be defined by the verb '*murder*' and vice versa. However, the status of '*victims*' is different because only *some victims* are murdered. A person can be a victim without being murdered, such as a victim of robbery or illness. For this reason the verb '*murder*' cannot be said to define the set of *victims.*

The advantage of using CIDE is that definitions for related concepts are stored under the same entry. If a verb is the major definition in a dictionary entry, the name of the concept representing its subject and object, if defined by the verb, are stored in the same entry. The operator is therefore able to look up the appropriate names quickly and accurately[8]. This is not the case with the layout of entries in other dictionaries.

[8] it is assumed that the operator has the particular dictionary entry available on-line while processing the definition.

6.9 Analysing Prepositions

The existence of a PP in an input sentence can result in two different types of ambiguity; structural and semantic. The reason for structural ambiguity is that a PP can modify[9] a number of objects in the preceding sentence. It is dealt with by the bracketing elimination algorithms introduced in Section 6.3. Semantic ambiguity exists because a preposition provides semantic knowledge about the nature of the relationship between the modified and prepositional objects. For example, the preposition '*with*' in the two sentences:

a. Rick shot the man with a gun.

b. He danced with joy.

in the most likely interpretation is used to express different relationships: INSTRUMENT and ATTRIBUTE respectively. The aim of the acquisition process introduced below is to extract the relationship being expressed by a preposition.

It is assumed (in much of the literature) that there are two kinds of grammatical constructs which can be modified by a PP:

verbs — *e.g.* the intuitive interpretation of sentence (a) in which the '*shooting*' is done '*with a gun*'. The PP '*with a gun*' is said to modify the verb '*shoot*'.

noun phrases — *e.g.* an alternative interpretation of (a) is that the *shooting* is at '*a man with a gun*'. The PP '*with a gun*' is said to modify the noun, '*a man*'.

The two types of structures, which result from the semantic analysis of parse trees which contain a verb and noun attachment of a PP, are shown in Figure 6.11.

[9] A prepositional phrase is said to attach to a particular object (generally nouns or verbs), or alternatively, modify that object. The two phrases are used interchangeably.

Whereas the PP is said to modify a verb, it may be more accurate to view it as modifying the event of the verb's performance, *i.e.* the PP '*with a gun*' more correctly modifies the event of '*Rick shooting the man*' than the verb '*shooting*'. Consequently, semantic analysis produces an arc from the main event labelled with the preposition as shown in the Figure 6.11 (a). A noun attachment of a PP results in the construction of an event having the preposition as action and the NP and prepositional object as subject and object respectively. This is shown in Figure 6.11 (b).

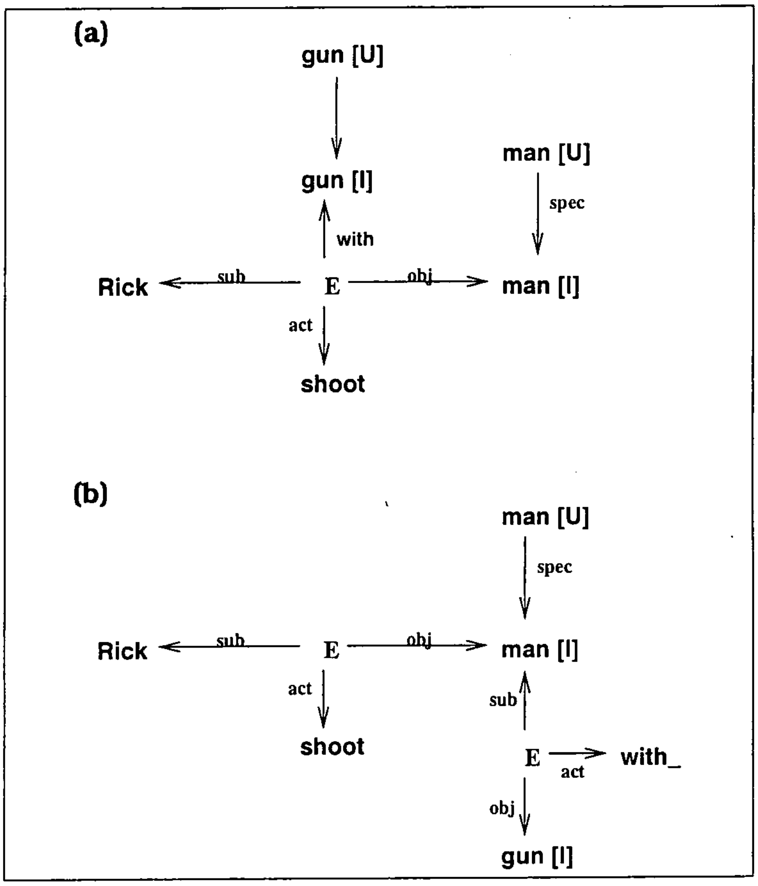


Figure 6.11: Semantic structures resulting from, (a) a verb attachment, and (b) a noun attachment, of a PP

The disambiguation procedure will transform the representation of the prepositions into meaningful semantic relationships. In the simplest case this will involve replacing the arc and action resulting from a noun and verb attachment of a PP respectively, by some more unambiguous relationship between the objects they relate.

The set of relationships indexed by a preposition can be found by examining its sense definitions within CIDE. In many cases the senses of prepositions have not been restricted in the CIDE defining vocabulary. The analysis involves examining the dictionary entries of approximately 190 sense definitions.

The information regarding the types of objects which can be modified by each sense of each preposition were recorded as a by-product of the analysis of their definitions. This information is gathered by examining the example usage sentences which accompany each definition. Several classes of definitions were identified.

6.9.1 Classes of Prepositions

Each class of preposition below is illustrated by providing a set of transformational rules which are used to interpret the meaning of the relationship identified by the particular preposition.

It is important to note that these rules (particularly the complex ones) are only heuristic in nature. The transformations are as complete as the examples which accompany the definition of each preposition. The strategy of applying transformational rules which attempt to make the semantic content of the preposition explicit is plausible only because the operator has subsequent opportunity (see Section 6.12) to reject incorrect analyses.

6.9.1.1 Prepositions Encoding A Simple Relationship

These are a class of prepositional senses whose meanings have a simple translation into the frame-like KRL of LOLITA. The particular sense translates into an action (in the case of noun attachments), or an arc [10] of an event (in the case of verb attachments). They are illustrated by the definitions below which include bulleted examples of their usage:

[10] arcs in SemNet correspond roughly to slots in a framed representation.

of$_{\text{THROUGH}}$ — through; having as the cause • He did it of his own free will • John died of cancer • He bought the car of his own free will

of$_{\text{CONTAINING}}$ — containing • a bottle of beer • a book of short stories • sacks of rubbish

The former sense of the preposition '*of*' only modifies verbs and asserts a CAUSE relationship between the prepositional object and the event of the verb's performance respectively, *i.e.* cancer *caused* John's death, his free will *caused* him to buy the car. The disambiguation process will entail replacing the arc labelled for which is produced by semantic analysis to an arc labelled cause_of[11] which is a primitive in the current semantic representation.

The latter sense of the preposition '*of*' only modifies nouns and asserts a CONTAIN relationship between the modified NP and the prepositional object, *i.e.* a bottle that *contains* beer, a book that *contains* short stories. The disambiguation process will entail replacing the action '*for*', which is produced by semantic analysis, by the correct meaning of the action CONTAIN (*i.e.* the concept of contain-to-hold and not contain-to-control).

The transformations above can be represented in a simplified notation by the rules:

R1. (X:event) of$_{\text{through}}$ (Y:entity) $\rightsquigarrow$ Y cause X

R2. (X:event) of$_{\text{containing}}$ (Y:entity) $\rightsquigarrow$ X contain$_{\text{hold}}$ Y

The rules provide transformations between the prepositional relationship represented on the LHS and its meaning on the RHS. The uppercase characters (*e.g.* X and Y) are meta variables denoting structures which are present in the semantic representation of an ambiguous preposition, *i.e.* X represents an event which is modified by the preposition '*of*'.

[11] the arc cause and cause_of are inverse of one another.

The first rule says that the preposition of$_{\text{through}}$ relates an event (of which X is a particular instance) and an entity (of which Y is a particular instance) which can be transformed into the event which has subject Y, object X and action *cause*.

6.9.1.2 Prepositions Encoding A Complex Relationship

The meaning of some prepositions cannot be represented by simple transformations which involve a direct replacement of the preposition with a specific semantic relationship. Examples of complex transformations are illustrated by the definitions:

for $_{\text{PAYMENT}}$ — getting in exchange • he paid $100 for the glasses • The mechanic repaired the car for a favour

for $_{\text{INTENDED FOR}}$ — intended to be given to • John bought a toy for the baby • There's a prize for the fastest runners in each category • Here's a romantic song for the ladies • Is the present for me?

whose semantic content may be extracted by the transformations shown in the following rules:

R3. (X:event) for$_{\text{payment}}$ (Y:entity)
$\rightsquigarrow$ X cause (S? get Y)

R4a. ((S A O):event) for$_{\text{intended for}}$ (Y:entity)
$\rightsquigarrow$ (S A O) has_goal ((S? give O) destination Y)

R4b. (N:entity) for$_{\text{intended for}}$ (M:entity)
$\rightsquigarrow$ A has_goal ((A give N) destination M)

These complex transformations are classified differently from the simple ones above because they often involve the construction of new events on the right hand sides of the rules. In addition, they are often complicated by the need to explicitly extract components (*e.g. subject*, *object* and *action*) of the modified event. The latter difference is contrasted in rules (R3) and (R4a). The structure X in (R3) represents an event whose components are hidden and the pattern (S A O) in rule (R4a) matches an event in which S is bound to its subject, A to its action and O to the object. These variables are used to build new structures on the RHS of rule (R4a).

Rules 4(a) corresponds to verb attachment of preposition for$_{\text{INTENDEDFOR}}$. The meaning of the preposition is best illustrated by considering an example, "*John bought a toy for the baby*" in which the concepts '*John*', '*buy*' and '*a toy*' are the subject S, action A and object O respectively. The interpretation of the PP '*for the baby*' is given by the instantiation of the corresponding concepts on the RHS of rule (4a):

(John buy a toy) for (the baby)
$\leadsto$ (John buy a toy) GOAL (John? give a toy to the baby)

The relationship specified by the preposition *for*, on the RHS of the rule, is to be understood as representing a GOAL relationship between the two events stated above. The question mark on the RHS of the rule means that it is only a plausible assumption (*i.e.* it is reasonable to assume) that '*John*' is also the agent of the '*giving*'.

Rule 4(b) corresponds to the noun attachment of for$_{\text{INTENDEDFOR}}$ where the variables N and M correspond to the modified object and the prepositional object respectively. It results in a statement such as "*the toy for the baby was lost*" to be interpreted as "*the toy which someone intends to give to the baby was lost*".

6.9.1.3 Prepositions Encoding A Number of Simple and Complex Relationships

It was noted in Section 5.2 that the definitions of words in dictionaries are often too coarse for NLP purposes. This holds for definitions of many prepositional senses, *e.g.*

for$_{\text{TO GET}}$ — in order to get or achieve • He ran for the bus • He's trying for a first in his exams • Simon applied for a job with another company

with$_{\text{AND}}$ — and or followed by • I'd like a steak with some salad • $200 is payable with a further $100 on delivery • He had steak and chips with cake for desert

The bulleted examples (taken from CIDE) show typical uses of each sense of the prepositions. Each definition is made from two distinct parts (or sub-definitions) separated by a disjunction. This disjunction does not reflect genuine uncertainty in the context in which the words sense is used because it is possible to disambiguate the particular sub-definition in all usages of the word. This can be illustrated with the the example sentences given above:

1. He ran in order to get the bus
2. Simon applied in order to get a job with another company
3. He's trying in order to achieve a first in his exams
4. $200 is payable followed by a further $100 on delivery
5. He had steak and chips followed by cake for desert
6. I'd like a steak and some salad

Although it is coherent to speak of '*getting a first in an exam*', the relationship of '*achieving*' is preferable because the latter is more specific, *i.e.* one can speak of 'getting a concrete entity' (*e.g.* the bus) but not of achieving it.

In order to extract the most informative semantic relationships from the senses of each of the definitions above it is necessary to split them into separate rules. The rule for the former example is:

R5. ((S A):event) for$_{\text{to get}}$ (Y:entity)
$\rightsquigarrow$ (S A) has_goal (S get Y) | (S A) has_goal (S achieve Y)

where the vertical bar | represents a splitting of the rule into the two cases on its left and right hand side.

6.9.1.4 Prepositions Whose Semantic Relationship Is Imprecise

Ideally, for NLP researchers, lexicographers would write definitions of prepositions such that for each definition, one or more easily identifiable relationships exist between the modified and the prepositional objects. Since this task would involve a great deal of resources, the definitions of many prepositions escape such precise formalisation. For example, the definition:

of$_{\text{THAT IS/ARE}}$ — that is/are • the skill of negotiating • the difficulty of raising twins • a rise of 2% • the pain of separation

encodes a number of different semantic relationships illustrated in the informal descriptions of just some of the examples above:

- the skill of negotiating is that skill required in order to achieve the goal 'negotiate'
- the difficulty of raising twins is the difficulty caused by the raising of twins
- a rise of 2% is a rise whose value is 2% of the original value

Generally, these specific relationships are not offered to the operator, since they would result in an explosion of the number of options which need to be considered.

6.9.2 An Example of Disambiguating Prepositions

An example of a preposition occurring in a definition was illustrated in Figure 6.5. The operator will be asked to disambiguate the relationship '*for*' between the event of *killing* and the noun *animal's meat*[12]. The question which is posed to the operator is shown in Figure 6.12.

```
Choose the meaning of the word "for" between "killing" and "meat" :

 1 showing the length of time
     eg, I'm going to sleep for an hour
 2 showing amount of distance
     eg, He drove for 10 miles
 3 towards; in the direction of
     eg, They followed signs for the town centre
 4 intended to be given to
     eg, Roberto bought a toy for the baby
 5 for the purpose of
     eg, The neighbours invited us for dinner
 6 in order to obtain
     eg, He sent off for the details.
 7 in order to go into and travel in
     eg, John ran for the bus; Roberto applied for a job
 8 in order to achieve
     eg, Kevin was trying for a first in his exams
 9 because of; as a result of
     eg, Bob was better for his weeks holiday
 10 compared to other similar things
     eg, Jane is very mature for her age
 11 getting in exchange
     eg, Roberto paid $100 for the glasses
 12 representing (a company, country, etc.)
     eg, John works for a charity; He swims for England
 13 in relation to
     eg, Sanjay has a great liking for spicy food
```

Figure 6.12: Disambiguating the sense of a preposition

The relationship should correctly be disambiguated to option (6), *i.e.* animals are killed *in order to get* their meat. The semantic transformation which is shown in

[12] it is assumed that the pronoun '*their*' occurring in the definition of '*abattoir*' has been correctly resolved to the '*animal*'.

Figure 6.13 for this interpretation is given by the rule (R5).

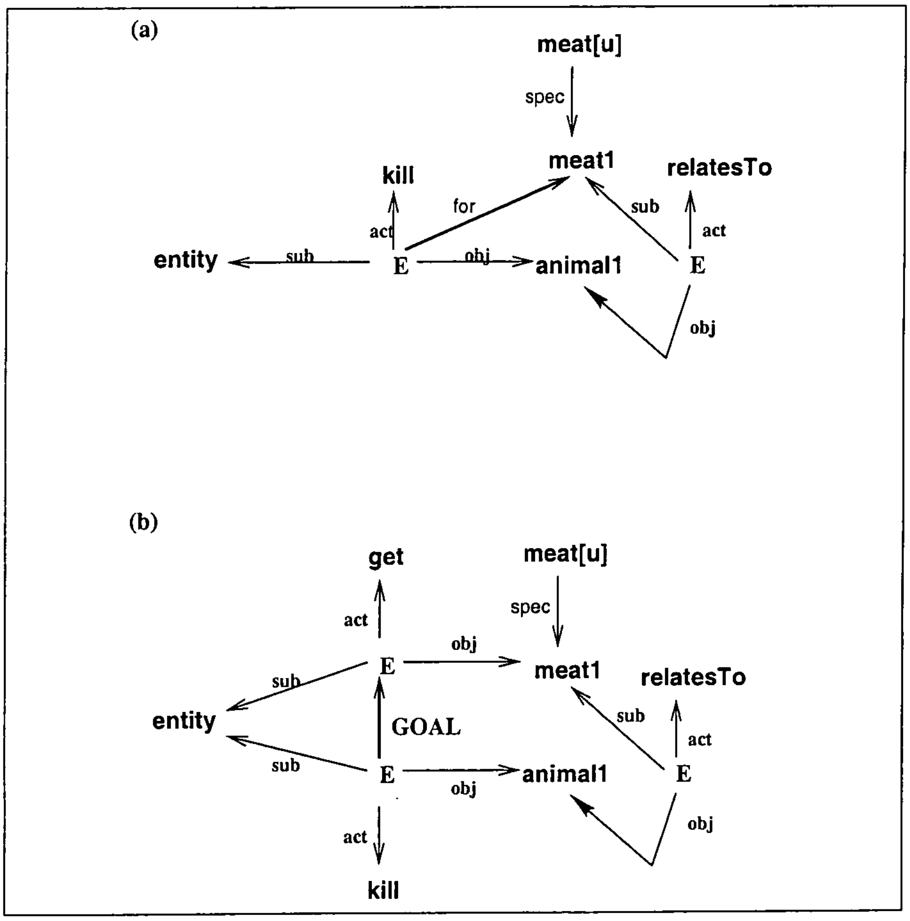


Figure 6.13: The semantic transformation following the disambiguation of the preposition '*for*' in the definition of '*abattoir*'

6.10 Compound Nouns

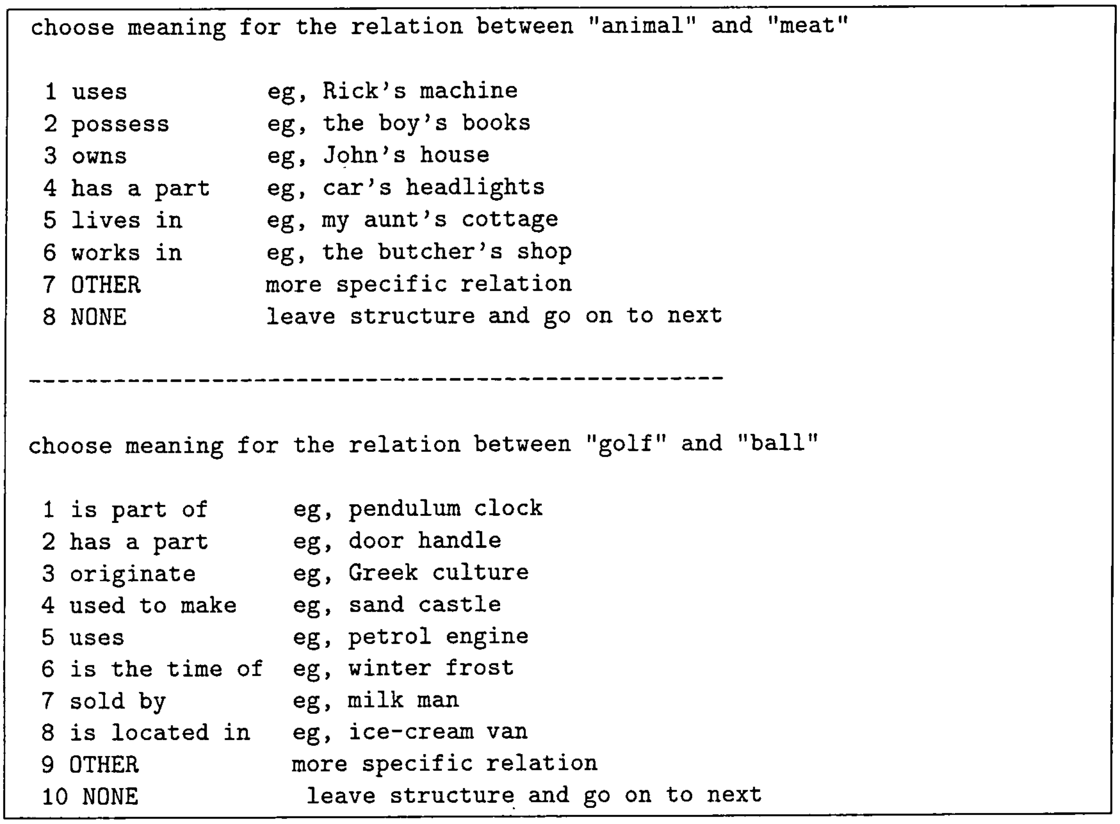
The final disambiguation procedure involves making the relationship between compound nouns occurring in a definition explicit. Two kinds of compounds are distinguished by semantic analysis:

possessives — *e.g.* '*John's house*', '*his books*' or '*Sarah's teeth*'

other compounds — *e.g.* '*car door*', '*stone furniture*' or '*baby oil*'

Possessives are distinguished from the general class of compounds because they encode a restricted set of semantic relationships, *e.g.* control, ownership, attribute *etc.*, between each noun.

Downing [Downing, 1977] provides evidence to show that one cannot derive a complete list of compounding relationships because novel compounds can always be created and coherently interpreted depending upon the surrounding context. Consequently, the best strategy is to offer the operator a list of common relationships with the option of choosing a more specific one by typing its name. Figure 6.14 shows an example of the question which the operator is expected to answer in the case of, (a) possessives, and, (b) other compounds.



```
choose meaning for the relation between "animal" and "meat"

 1 uses          eg, Rick's machine
 2 possess       eg, the boy's books
 3 owns          eg, John's house
 4 has a part    eg, car's headlights
 5 lives in      eg, my aunt's cottage
 6 works in      eg, the butcher's shop
 7 OTHER         more specific relation
 8 NONE          leave structure and go on to next

-------------------------------------------------

choose meaning for the relation between "golf" and "ball"

 1 is part of      eg, pendulum clock
 2 has a part      eg, door handle
 3 originate       eg, Greek culture
 4 used to make    eg, sand castle
 5 uses            eg, petrol engine
 6 is the time of  eg, winter frost
 7 sold by         eg, milk man
 8 is located in   eg, ice-cream van
 9 OTHER           more specific relation
 10 NONE            leave structure and go on to next
```

Figure 6.14: The formalisation of the relationship between (a) possessive nouns, and (b) other compounds.

The set of relationships offered in each case are taken from [Downing, 1977]. Often the relationship between the nouns cannot be stated by using a simple relationship. For example, the relationship between '*telephone*' and '*number*' in the compound '*telephone number*' would require a complex structure to represent it, *i.e.* "*the* *number which uniquely identifies a line to which a telephone can be attached etc.*". This complexity could be captured by a process in which the operator enters a description which would subsequently be disambiguated using the same process as outlined in this chapter.

6.11 Naming Events

A dictionary entry for a verb defines a prototypical event[13] of the verb's performance. There are two types of linguistic constructions which introduce events:

1. a verb together with its participants — *e.g.* "*John married Mary*", "*He ran*", *etc.*
2. an event noun — which is a name used in this thesis for a class of nouns with a temporal dimension, *e.g.* in the sentences

    - John was fired following the *investigation*
    - she suffered a *hangover* after the *party*
    - The *murder* occurred last Tuesday

    the nouns '*investigation*', '*hangover*', '*party*' and '*murder*' are events that happen or occur although no explicit verbs are mentioned.

Each event noun names an event. The event which is named must have a corresponding action. The action often corresponds to a verb which *may* have an obvious lexical connection to the original event noun, For example:

- an *investigation* is the name of the process where '*someone investigates something*'.

[13] an *event* is something that happens and has a temporal dimension to it. An event is distinguished from an *entity* which is simply assumed to exist.

- a *party* is the name of the process when '*people party*'.
- a *murder* is the name of the process when '*someone murders someone*'.

Some event nouns may have no obvious verb form. An example is the noun '*hangover*' which names the event of '*someone feeling ill after drinking too much*'. Conversely, many of the events which correspond to the performance of a verb defined will not have a corresponding event noun describing it.

The layout of entries in CIDE is advantageous when acquiring the type of knowledge which relates a verb and an event noun in a definition, because:

- the morphologically related event nouns are easily accessible. For example, the entry which defines the verb '*manufacture*' (see Figure 5.1) contains the sub-entry of the name of the semantically related event noun, also called '*manufacture*'. In other words, the process of '*manufacture*' happens when '*someone manufactures something*'. This is shown in Figure 5.1.
- the two words do not have separate definitions. Most other dictionaries would contain separate entries for the semantically related concepts. Consider the following definitions of the verb '*kill*' and event noun '*killing*' taken from COBUILD:

  kill$_v$ — to kill a person, animal, plant or other living thing means to cause the person or thing to die

  killing$_n$ — a killing is the act of deliberately killing a person

  which would require the analyses of two definitions in order to establish a relationship between the two concepts. In the case of CIDE, the same relationship could be established as a by-product of the analysis of only one definition (the verb) by exploiting the layout of entries.

The acquisition of knowledge which documents this correspondence between verbs and event nouns is important in an NLP system because it allows implicit structure to be revealed by linking semantically related concepts, which, otherwise, belong in very different hierarchies. For example, the sentence:

"Rick invited Mary for *dinner*"

can be understood in the expanded form as representing the collection of events:

E_1 = Rick invite Mary
E_2 = Mary eat dinner
E_3 = E_1 has_goal E_2

after disambiguating the preposition '*for*', whose interpretation relates the event of a '*dinner*' to its expanded form '*someone eat dinner*'. Note that the latter sense of the word *dinner* is the sense of `dinner-as-food`.

The operator will be expected to provide a name for the event of the performance of an action when it exists. During the processing of the definition of the verb '*kill*' the operator will be asked the question:

```
Is there a name for the process:
  "killers kill victims"
```

which can be deduced by examining the CIDE entry for the verb. The operator will need to pick the correct meaning of the semantic concept corresponding to the event noun in the case of ambiguity. The process for this is the same as outlined in Section 6.1.

The representation which links the semantic concepts corresponding to an event noun and the verbal form of the event is shown in Figure 6.15. It shows fragments from three IS-A hierarchies: the entity hierarchy, the event noun hierarchy and the action hierarchy. The node in the event hierarchy which represents the concept of *killing* has the action *kill* from the action hierarchy, the subject *killer* and object *victims* both from the entity hierarchy. Put another way, an event called *killing* consists of a *killer* performing the action *kill* on a *victim*. The representation enables the normal rules of inheritance to operate by simply moving up and down each of the three inheritance hierarchies shown in the diagram.

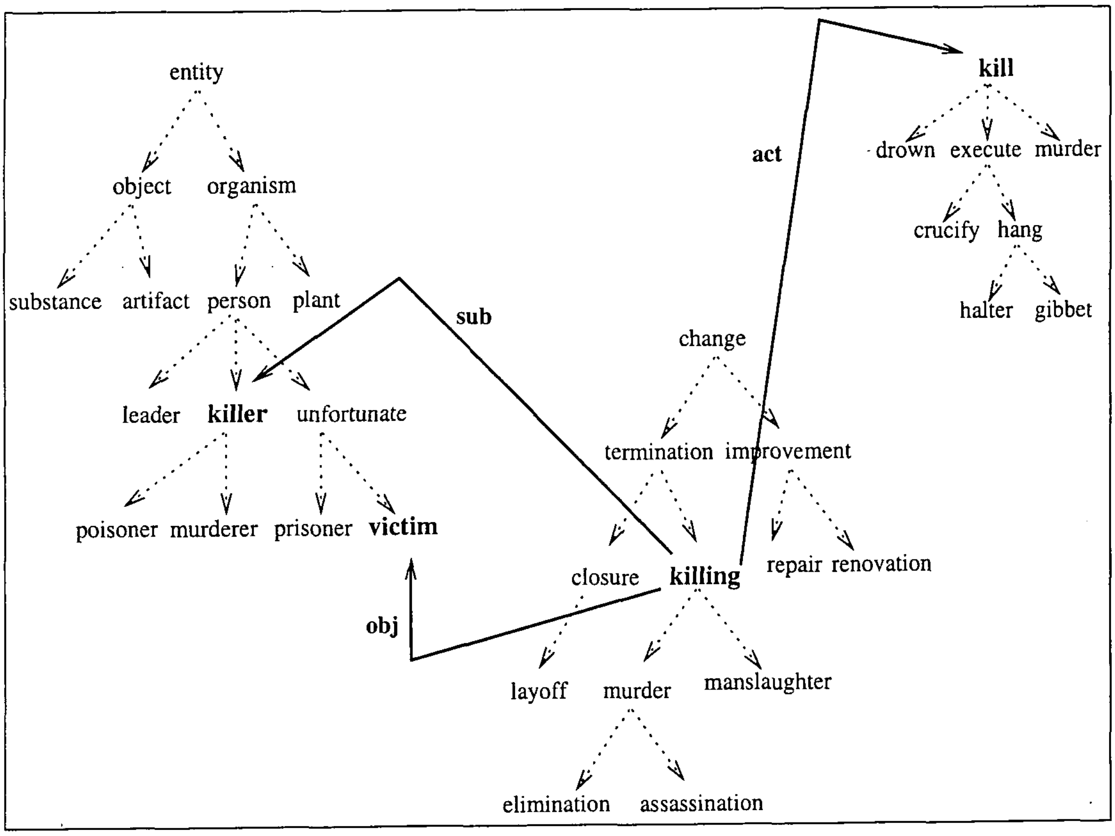


Figure 6.15: The representation of knowledge which links an event noun with its verbal form

6.12 Confirming the Analysis

During the normal course of analysis of a definition, an error may occur in one of the previous stages. Examples of situations where this might happen include:

- wrong parse — it was previously noted that the cost of a strategy whereby the operator rejects parses is that it does not guarantee that the correct one is selected.

- an error in semantic analysis — in cases where a definition is parsed correctly, the extraction of semantic relationships from the resulting parse is not necessarily trivial and may result in an incorrect structure being built.

The detection of incorrect semantic structures often requires a developer to examine the network of links and nodes (in textual format) which result from the semantic analysis of some input text. Understanding the semantic relationships which are created during the analysis of input text requires an understanding of the representational language of SemNet. Training many operators to understand this language is not a feasible approach given the constraints previously mentioned in Chapter 2.

A more feasible approach is to use the LOLITA generator [Smith, 1996] to generate a natural language description of the resulting semantic structures. In this way, the generator can be a powerful and useful debugging tool because it provides a natural interface to SemNet. The generator is able to provide NL descriptions of any concept (*i.e.* node) in SemNet. For example, during the analysis of the input:

> "each jack is a piece of equipment which can be opened slowly to allow heavy weights to be raised"

LOLITA's semantic analysis process generates a number of concepts which correspond to the entities and relationships present in the definition above. The generator is subsequently able to provide the following natural language descriptions of each dynamically created concept:

- "Pieces of equipment"
- "Some pieces of equipment"
- "Pieces that something could open slowly so that jacks could allow something to be raising heavy weights"

- "Something that raises heavy weights"
- "Something raises heavy weights"
- "Heavy weights that something raises"
- "Jacks could allow something to be raising heavy weights"
- "Something that could slowly open pieces so that jacks could allow something to be raising heavy weights"
- "Something could open slowly pieces so that jacks could allow something to be raising heavy weights"
- "Jacks are pieces that something could open slowly so that they could allow something to be raising heavy weights"

The word 'something' is used to replace the name of implicit entities that occur in the definition. The most important description is the final one because it describes the assertion of the outermost relationship in the input text (*i.e.* the event which asserts the '*is*' relationship between the headword and head of the first NP/VP of a definition). Consequently this node can be used to generate a description of the the complete set of semantic structures which have been extracted from the input. It is therefore sufficient for the operator to examine a single description:

```
Accept definition as:
  "Jacks are pieces that some humans could slowly open so
   that they could allow they to be raising heavy weights."
```

as a confirmation that the definition has been analysed correctly. This presentation of the analysis is advantageous because it takes advantage of the human's ability to comprehend language. Errors in the analysis are easily identified because all the implicit information in the original definition is made explicit in the descriptions provided by the LOLITA generator, *i.e.* PP's occur adjacent to the items they modify, the referents of implicit entities are made explicit, *etc.*

6.13 Representing Semantic Structures

It was previously mentioned that verb entries define the event of the verb's performance. This event is termed the *prototypical event.* For example, the protypical event of '*strangulation*' is defined by the verb '*strangle*':

strangle — to kill someone by pressing their throat so that they cannot breathe

Prototypical events are distinguished because their instances permit the inference of the semantic relationships specified in the definition. For example, a strangulation consisting of a particular strangler, John, strangling a victim, Mary, by definition permits the inferences:

E_1 = John kills Mary
E_2 = John presses Mary's throat
E_3 = Mary's throat is a part of Mary
E_4 = Mary cannot breathe
$E_5 = E_2$ causes E_4
$E_6 = E_5$ causes E_1

A flexible representation for prototypical events must enable inferences from instances of the prototype (*i.e.* John strangle Mary) to the properties which hold as a consequence of the definition of the prototype (*i.e.* E_{1-6}).

The required inferences can be made in SemNet by assuming that an instance of a prototypical event inherits properties from the corresponding arc of the prototype. This is illustrated in Figure 6.16.

The figure shows two prototypical events named '*killing*' and '*strangulation*' with the event "*John strangles Mary*" being an instance of the latter. This relationship enables the inference that "*John is a strangler*" and that "*Mary is a victim*". The

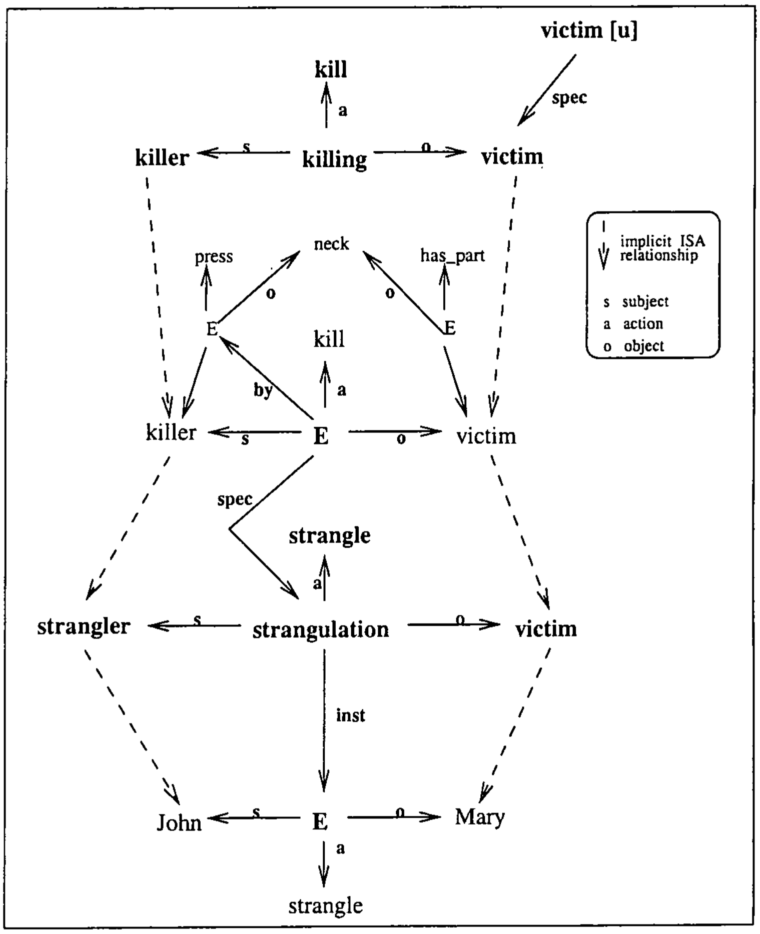


Figure 6.16: The representation of prototypes in SemNet.

properties specified by the events E_{1-6} hold by normal inheritance rules of the semantic representation [Long and Garigliano, 1988][Baring-Gould, forthcoming]. In the same way that the event "*John strangles Mary*" is an instance of the prototype of *strangulation*, the event "*John kills Mary by pressing her throat*" is an instance of the prototype of the event *killing.* The latter relationship permits the inference that "*John is a killer*" (because stranglers are killers) and that "*Mary will die*" (not shown on the figure) as a result of the *strangulating.* Mary's death is inferred by extracting the semantic relationships from the definition of the verb *kill.*

6.14 Summary

This chapter has outlined a semi-automated approach to the extraction of semantic relations from dictionary definitions using the LOLITA system. The input to LOLITA is a transformed version of a definition which the system can analyse. The system then extracts points of ambiguity from the definition, which are subsequently resolved by an operator. The format of the presentation of ambiguities, was considered at length because it is unreasonable to expect operators (possessing limited linguistic expertise) to analyse complex structures such as parse trees.

It was also shown that definitions contain a lot of implicit knowledge (*e.g.* selectional restrictions for verbs, implicit entities *etc.*) which is extremely useful to an NLP system. Procedures were given to extract this knowledge in a feasible way. The feasibility of the approach relies upon the existence of the appropriate NLP machinery (*i.e.* the LOLITA system) to extract and present the information to the operator in a clear and concise way.

Finally, a representation for the semantic relationships which are extracted from the dictionary definition was considered. This is done by introducing the notion of a prototypical event of an action's performance. It was shown how a hierarchy of prototypical events could be represented in a way which allows inheritance procedures to infer relationships from the generic prototype to a particular instance.

Chapter 7

Implementation

The previous chapters have introduced a procedure for the semi-automated acquisition of semantic relations from dictionary definitions. This chapter illustrates the implementation of the procedure by showing two fully worked examples. The examples consist of questions (in the form of output from the LOLITA system) that an operator is asked, together with commentary on the correct answers that they would be expected to provide.

The setup of the acquisition process in Figure 6.1 shows that the operator does not interact directly with the LOLITA system during the analysis of a definition. An easy to use, point and click, Windows-based user interface has been built. The main functionality of the interface is to communicate questions from LOLITA to the operator and answers from the operator back to the LOLITA system, during the question answering process. A screen shot of the interface for each different category of question is shown at the relevant stages in the walk-through examples below.

7.1 Walk-Through Examples

The walk-through examples below present fully worked question-answering sessions for the definitions:

hide $_n$ — the strong thick skin of an animal which is used for making leather

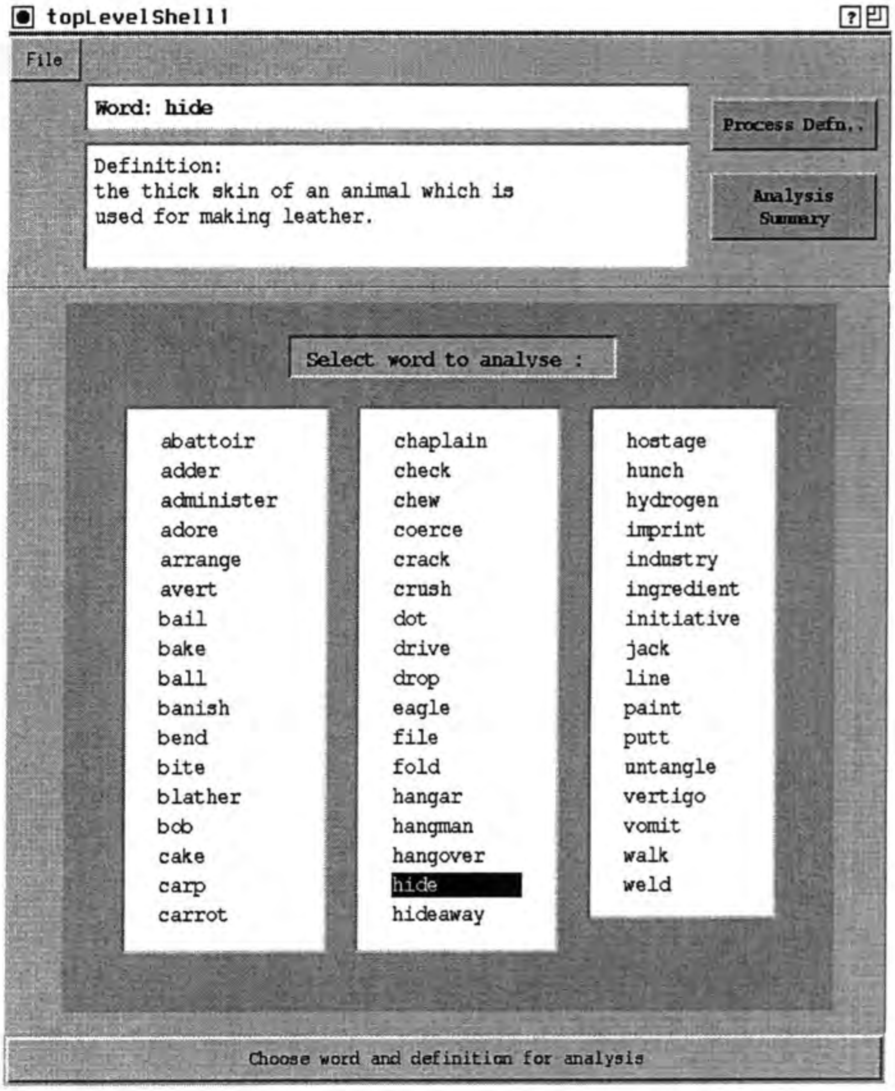
banish $_v$ — to send someone away from their country and forbid them to come back

Upon executing the interface, the operator is presented with several lists of words whose definitions they may browse by selecting the appropriate entry. The initial interaction is shown in Figure 7.1. Any particular definition can be selected for processing by the operator. The interface process will subsequently initiate processing by passing the headword and its definition to LOLITA. The long term aim of the project is to enable the operator to browse all the definitions in the dictionary. This requires parsing the lexicographers typeset files (a non trivial task [Alshawi, 1989]) to extract the main definition in each CIDE entry.

A web site has been created which contains a non technical description of the types of questions that are posed to the operator and the decisions which he/she should make when answering questions. The content of the user manual is shown in Appendix C. In addition, Appendix D contains a user manual describing the operation of the user interface.

The interaction below is shown in an entirely text-based format[1]. The questions and their possible answers are shown on indented lines beginning with a '>' symbol. These lines represent output from the LOLITA system which may subsequently be reformatted by the user interface. The purpose of the walk-through examples below is to explain the content of the questions and not the format of the actual

[1] The processing of definitions can be initiated not only through the graphical interface but also in command line mode through a text-based LOLITA interface. The traces shown are from this latter mode of operation.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

abattoir
adder
administer
adore
arrange
avert
bail
bake
ball
banish
bend
bite
blather
bob
cake
carp
carrot

chaplain
check
chew
coerce
crack
crush
dot
drive
drop
eagle
file
fold
hangar
hangman
hangover
hide
hideaway

hostage
hunch
hydrogen
imprint
industry
ingredient
initiative
jack
line
paint
putt
untangle
vertigo
vomit
walk
weld

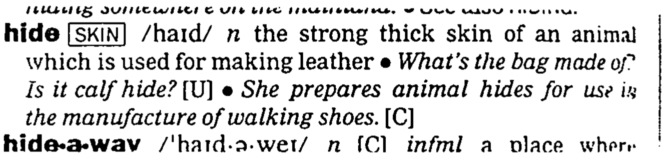
Choose word and definition for analysis

Figure 7.1: The initial interface screen showing the selection of a definition to analyse

interaction. The walk-through indicates the correct answer that should be selected, and also provides a description as to why that answer is the most appropriate.

7.1.1 The Noun '*hide*'

While working through the example below the operator will have the complete CIDE entry (shown in Figure 7.2) available for inspection. Although currently this is in textual form, the eventual aim is to present it on-line through the user interface. After selecting the word 'hide' (Figure 7.1) from those available and after initiating the processing, the first interaction between the system and the user is[2] the following:



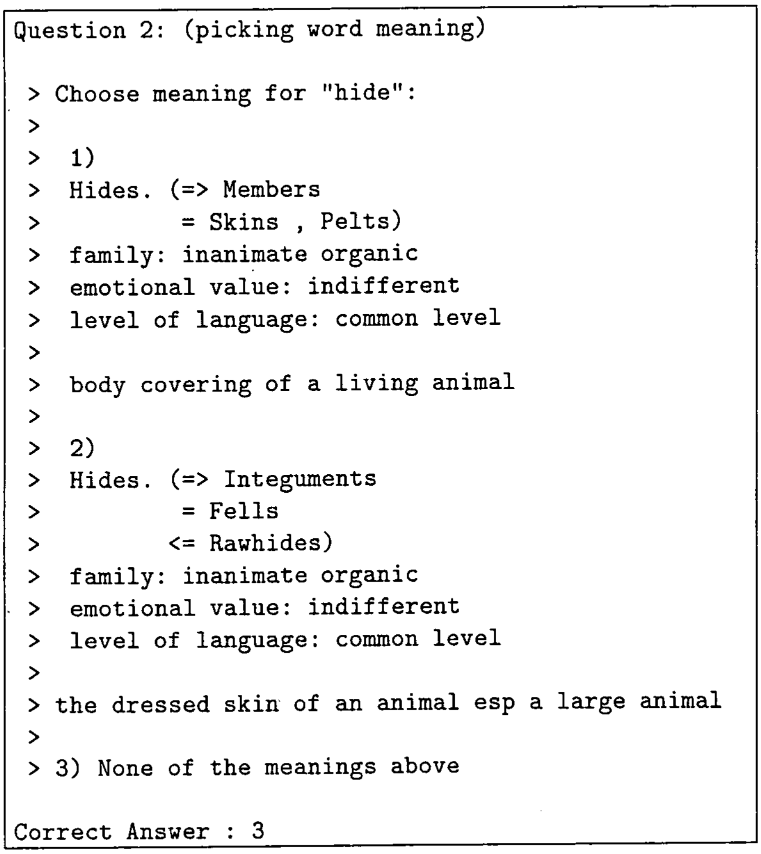
hide SKIN /haɪd/ *n* the strong thick skin of an animal which is used for making leather • *What's the bag made of? Is it calf hide?* [U] • *She prepares animal hides for use in the manufacture of walking shoes.* [C]
hide-a-way /'haɪd·ə·weɪ/ *n* [C] *infml* a place where

Figure 7.2: The CIDE entry for the noun '*hide*'.

```
Question 1: (choosing grammatical category)

 >  Choose grammatical category for "hide":
 >
 >   0 Noun representing an entity
 >   1 Noun representing an event
 >   2 Verb

Correct Answer : 0
```

[2] The headings following each question number refer to headings of sections in the user manual shown in Appendix C

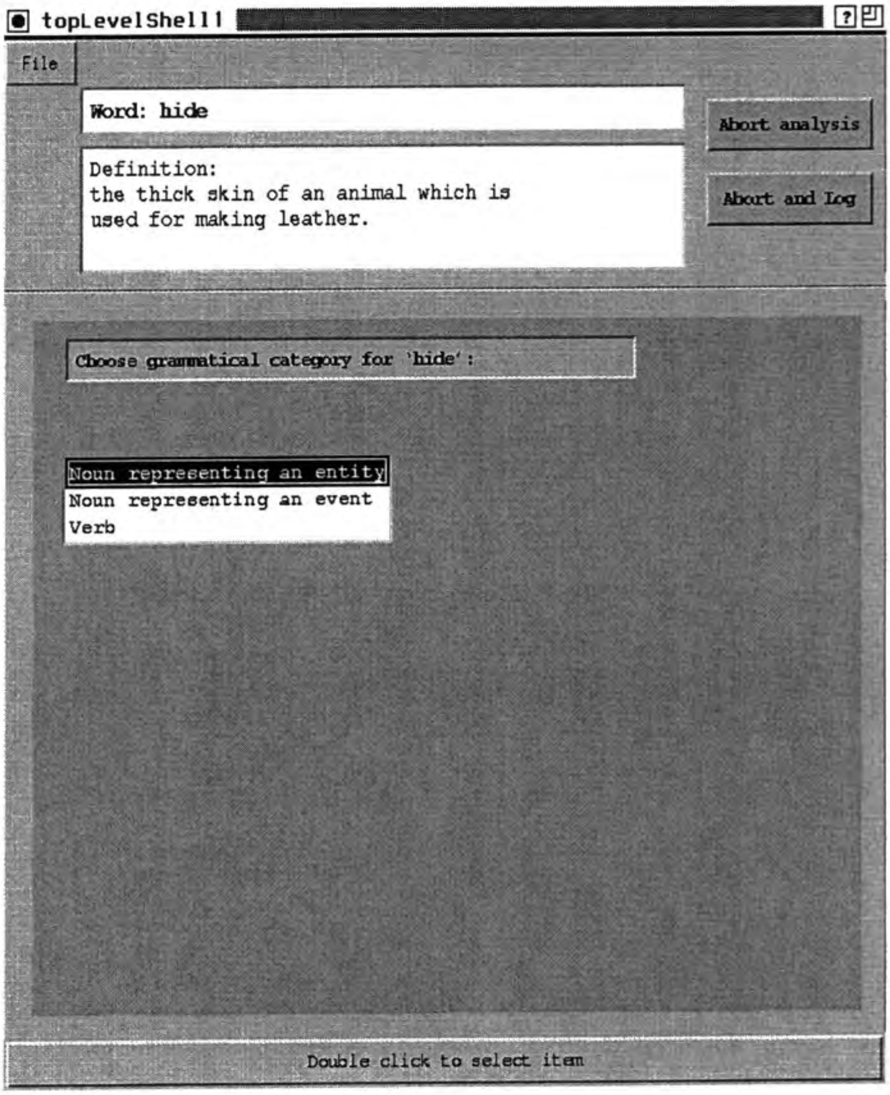
CIDE specifies the word to be a noun so the choice of answer is either 0 or 1. The question to ask oneself as to whether it is an event is: "is there a sense in which a '*hide*' can occur or happen?". Clearly not in this case and so the correct grammatical category is a noun representing an entity. As previously mentioned, the long term aim of the project is to extract the POS of a word automatically by parsing each dictionary entry therefore making part of the question redundant[3]. The advantage of using the information above is that it allows for the pruning of the number of word senses needed in the questions which follow. The interface screen shot for the question above is shown in Figure 7.3.



```
Question 2: (picking word meaning)

 > Choose meaning for "hide":
 >
 >  1)
 >  Hides. (=> Members
 >          = Skins , Pelts)
 >  family: inanimate organic
 >  emotional value: indifferent
 >  level of language: common level
 >
 >  body covering of a living animal
 >
 >  2)
 >  Hides. (=> Integuments
 >          = Fells
 >         <= Rawhides)
 >  family: inanimate organic
 >  emotional value: indifferent
 >  level of language: common level
 >
 > the dressed skin of an animal esp a large animal
 >
 > 3) None of the meanings above

Correct Answer : 3
```

The first sense of 'hide' is the skin of the animal which is actually on the animal, and the latter 'hide' is skin which is taken off the animal and used as a body covering, as indicated by the informal descriptions and list of semantically related concepts associated with each meaning. The CIDE definition is a generalisation of both

[3] the POS of a word cannot distinguish between an entity and an event.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

Choose grammatical category for 'hide':

Noun representing an entity
Noun representing an event
Verb

Double click to select item

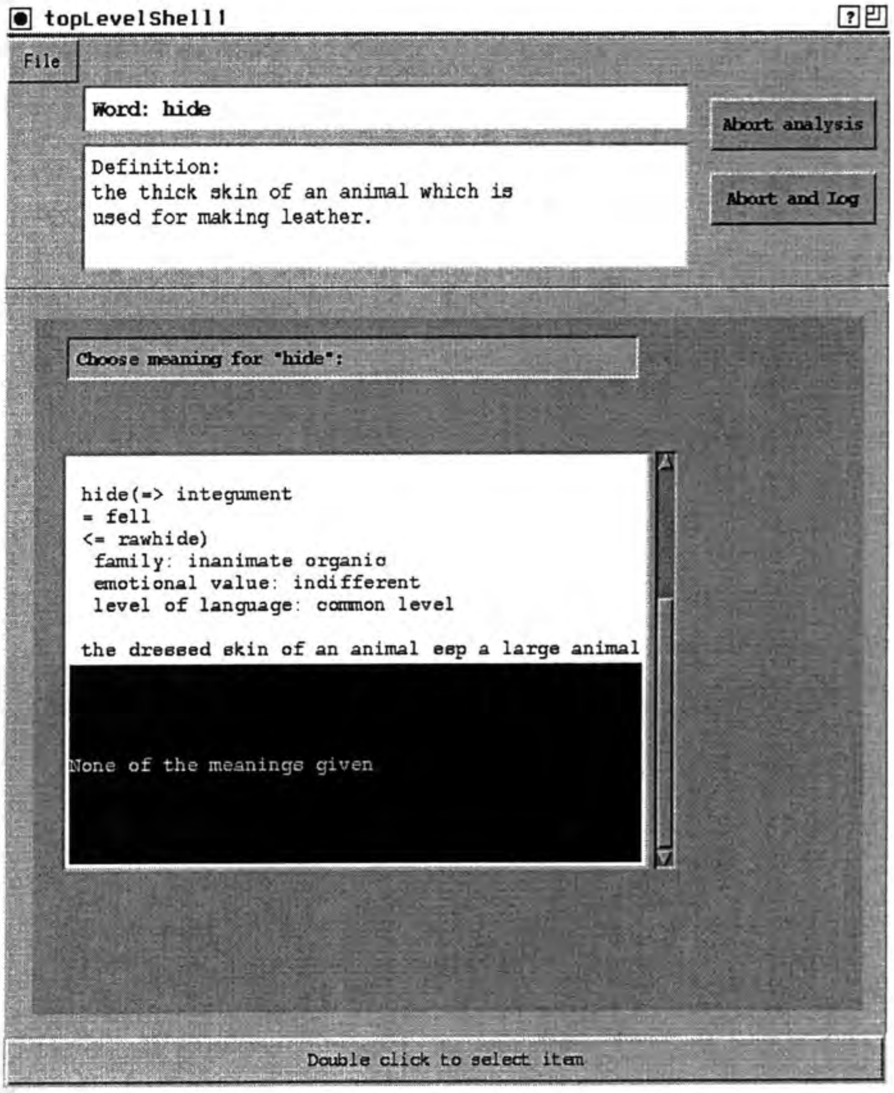
Figure 7.3: The graphical interface for selecting the word category

meanings because it does not make any such distinction regarding the temporal aspects of the process of removing and using the skin. The interface screen shot for picking between a number of word meanings is shown in Figure 7.4.

```
Question 3: (entering word information)

 > Do you wish to enter controls?

Correct answer : Yes
```

In the graphical version of the process, controls are displayed in a form which allows a simple selection of information. This is illustrated in Figure 7.5. It was mentioned in Section 6.2 that two types of controls exist: grammatical information explicitly mentioned in the dictionary entry, and other pragmatic knowledge which is implicit in the word usage. In a parsed version of the dictionary entry, the former class of control information can be extracted automatically. However, the latter type of knowledge needs to be explicitly entered by the operator. The two types of knowledge are separated in the graphical version of the interface for clarity.

In this case, the dictionary entry shows that the sense of '*hide*' which is being analysed can be used in an uncountable way, *e.g.* "*is the bag made of calf hide*". The operator should therefore select the appropriate option from the interface.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

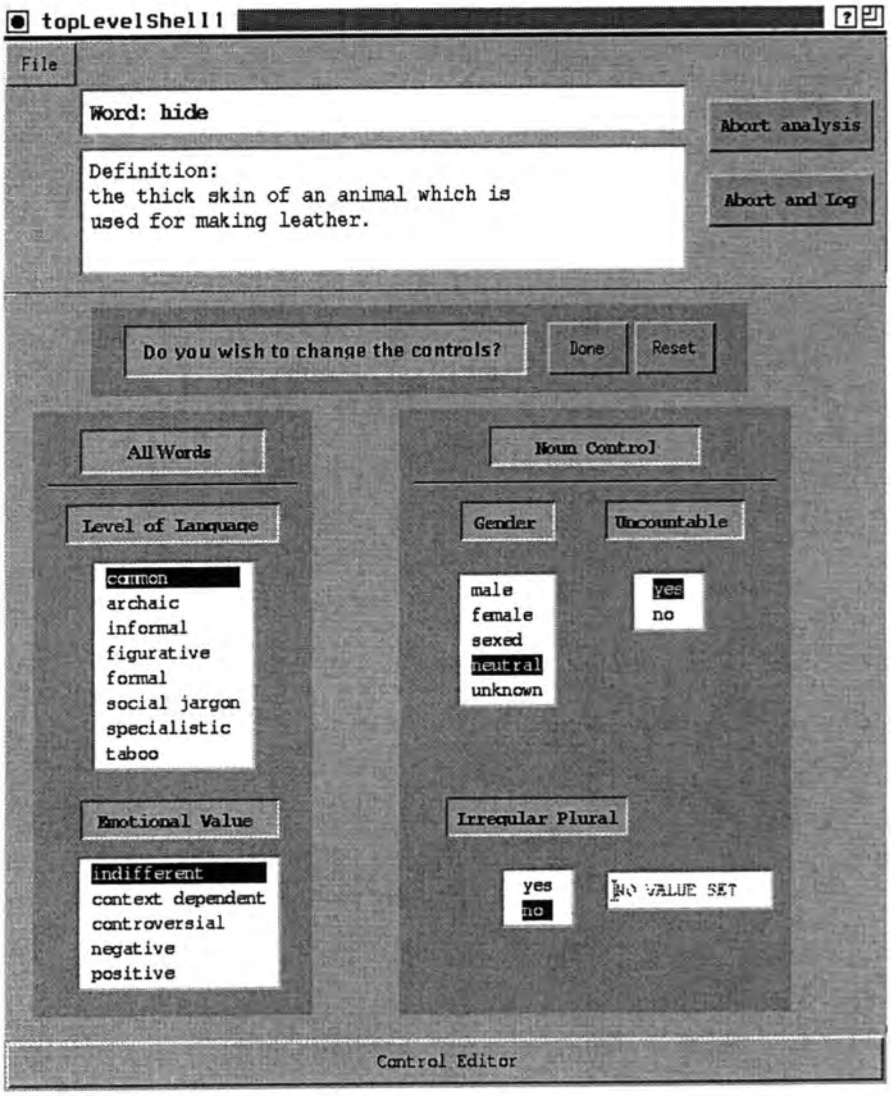
Choose meaning for "hide":

```
hide(=> integument
= fell
<= rawhide)
 family: inanimate organic
 emotional value: indifferent
 level of language: common level

the dressed skin of an animal eep a large animal
```

None of the meanings given

Double click to select item

Figure 7.4: The graphical interface for selecting word meanings



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

Do you wish to change the controls?

All Words

Level of Language: [✓ common] archaic informal figurative formal social jargon specialistic taboo

Emotional Value: [✓ indifferent] context dependent controversial negative positive

Noun Control

Gender: male female sexed [✓ neutral] unknown

Uncountable: [✓ yes] no

Irregular Plural: yes [✓ no] NO VALUE SET

Figure 7.5: The graphical interface for selecting controls of nouns

```
Question 4: (solving structural ambiguities)

 > Reject incorrect bracketing<s>:
 >
 > 1 each hide is the strong thick (skin of an animal which is used
 >   for making leather)
 > 2 each hide is the (strong thick skin of an animal) which is used
 >   for making leather

Correct answer : neither should be rejected.


Question 5: (solving structural ambiguities)

 > Reject incorrect bracketing<s>:
 >
 > 1 each hide is the strong thick skin (of an animal which is used
 >   for making leather)
 > 2 each hide is the strong thick (skin of an animal) which is used
 >   for making leather

Correct answer : neither should be rejected
```

The operator is given a number of rules (see Appendix C) which enable them to reject erroneous interpretations. The decisions made by the operator rely upon having an understanding of the basic entities (*i.e.* formed by noun phrases) and attachments (*e.g.* prepositional phrases) which exist in the text.

In the cases above, both bracketed units are coherent because the words contained within them do not describe an entity which does not exist in an intuitive understanding of the text, nor are any possible attachments violated.

```
Question 6: (solving structural ambiguities)

 > Reject incorrect bracketing<s>:
 >
 > 1 each hide is the strong thick skin of (an animal which is used
 >   for making leather)
 > 2 each hide is the strong thick skin (of an animal) which is used
 >   for making leather

Correct answer : reject (1)
```

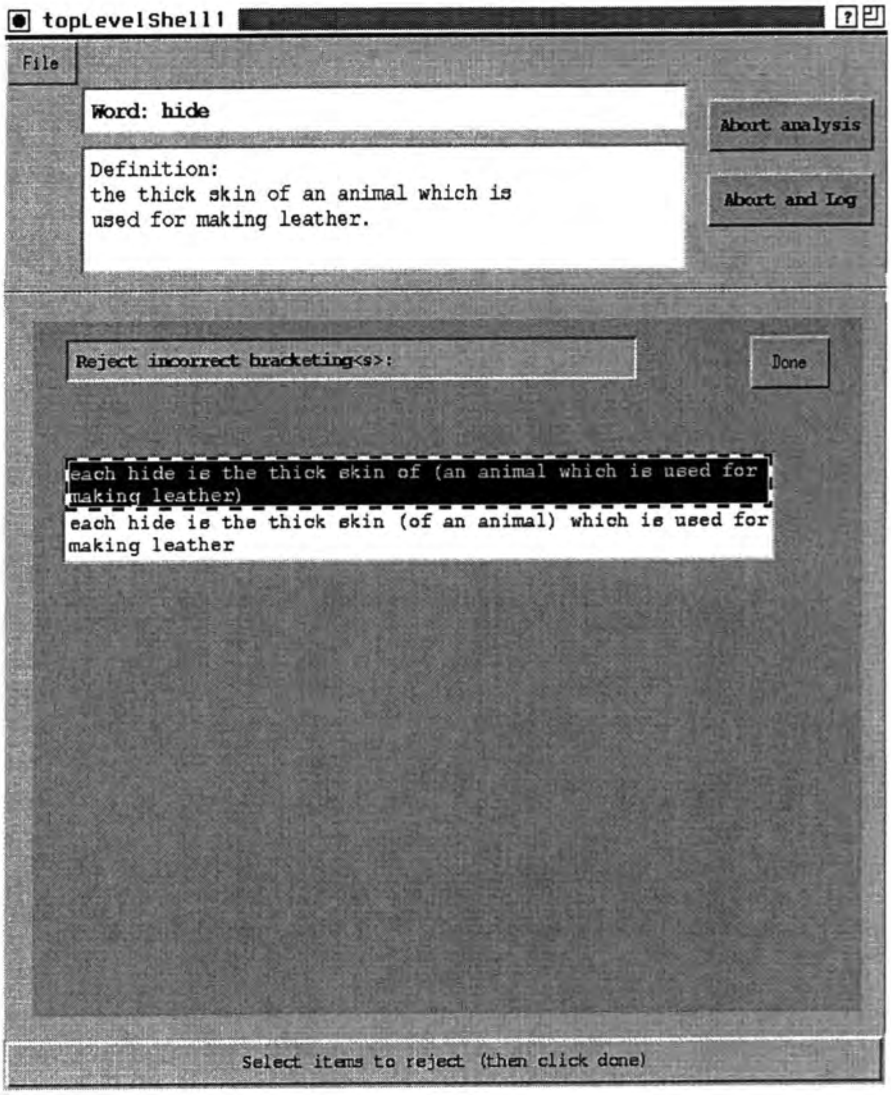
Option 1 violates the intuitive understanding of the text because there does not exist an entity described by '*an animal which is used for making leather*', *i.e.* it corresponds to the interpretation:

> each hide is the strong thick skin of an animal, and that animal is subsequently used for making leather

which is not intended by the definition. Hence option 4 is rejected at this stage. The screen shot for rejecting parses is shown in Figure 7.6.

```
Question 7: (picking word meanings)

 > Choose a referent for the verb #make# in:
 >  "something makes leathers"
 >
 >  1)
 >  To make  (=> To create
 >             = To produce
 >            <= To return , To print , To preassemble , To reproduce,
 >               To smelt  , To extrude , To generate , To generate ,
 >               To bootleg , To laminate , To elaborate ,
 >               To overproduce , To machine , To redo , To breed)
 >     relation type: transitive
 >
 > create a product: "We produce more cars than we can sell"
 >
 >  2)
 >  To make  (=> To accomplish
 >             = To carry , To effect , To do , To execute , To perform.
 >            <= To exaggerate , To complete , To back-date  , To apply ,
 >               To enforce)
 >      emotional value: positive
 >      relation type: transitive
 >
 > carry into effect;
 >     e.g., "make an effort"; "do research"; "carry too far"

Correct Answer : 1
```

From the informal description it is easy to see that the sense of '*make*' intended in the definition corresponds to the former one, *i.e.* the act of creating something. The graphical screen-shot for word sense disambiguation is shown in Figure 7.7.


topLevelShell1
File
Abort analysis
Abort and Log
Reject incorrect bracketing<s>:
Done
Select items to reject (then click done)

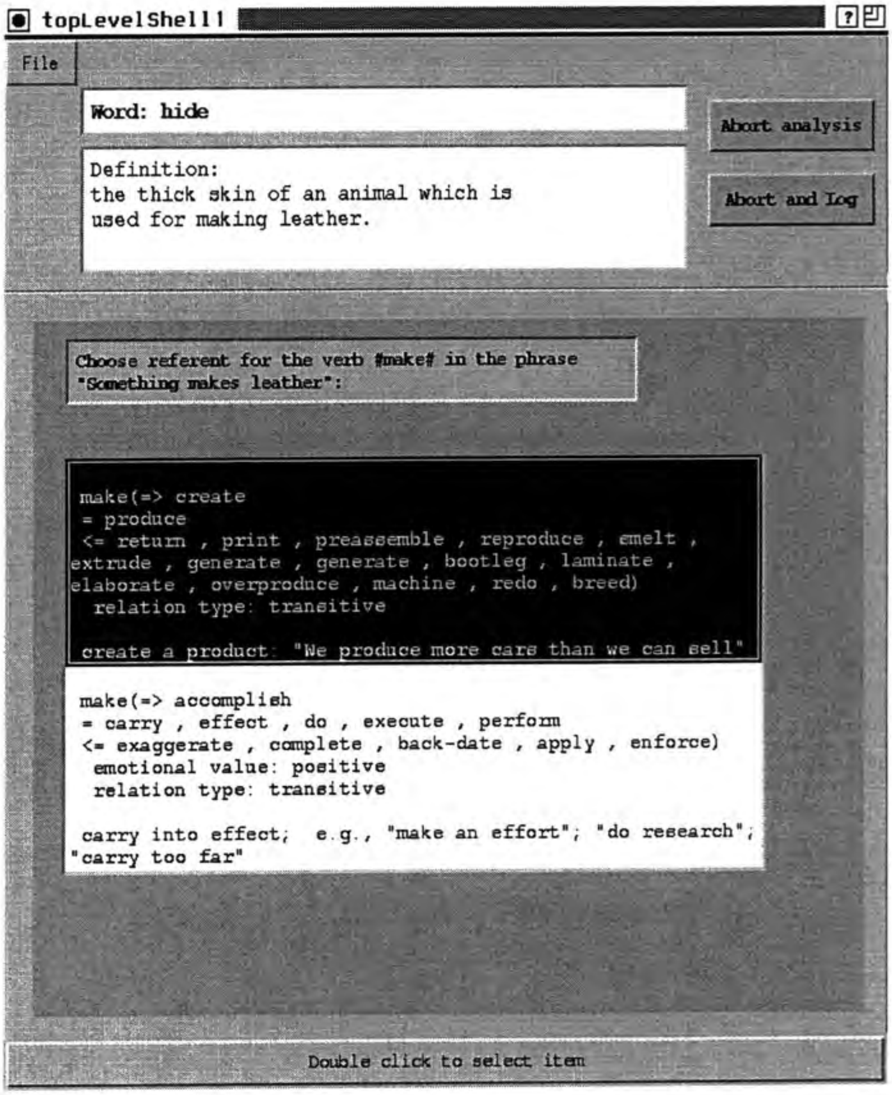
Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

each hide is the thick skin of (an animal which is used for making leather)

each hide is the thick skin (of an animal) which is used for making leather

Figure 7.6: The graphical interface for rejecting incorrect parses



Word: hide

```
Definition:
the thick skin of an animal which is
used for making leather.
```

Choose referent for the verb #make# in the phrase
"Something makes leather":

```
 make(=> create
 = produce
 <= return , print , preassemble , reproduce , smelt ,
extrude , generate , generate , bootleg , laminate ,
elaborate , overproduce , machine , redo , breed)
  relation type: transitive

 create a product: "We produce more cars than we can sell"
```

```
 make(=> accomplish
 = carry , effect , do , execute , perform
 <= exaggerate , complete , back-date , apply , enforce)
  emotional value: positive
  relation type: transitive

 carry into effect;  e.g., "make an effort"; "do research";
"carry too far"
```

Double click to select item

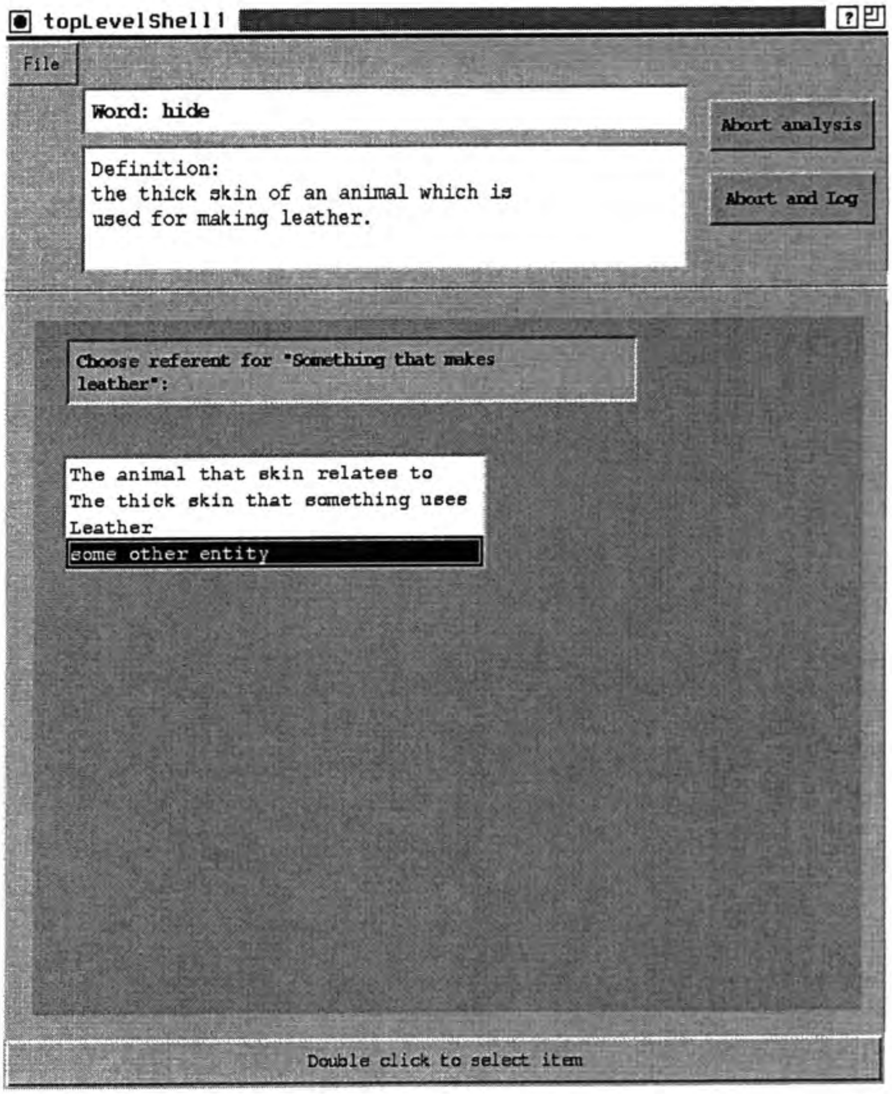
Figure 7.7: The graphical interface for disambiguating word senses

```
Question 8: (picking word meanings)

 > Choose a referent for the adjective #thick# in the original
 > definition:
 >
 > 1)
 > Thick things. ()
 >  family: generic
 >  emotional value: indifferent
 >  level of language: common level
 >
 > relatively thick in consistency
 >
 > 2)
 > Thick things. ()
 >  family: concrete
 >  emotional value: indifferent
 >  level of language: common level
 >
 > not thin; of relatively great extent from one surface to the
 > opposite usu in the smallest solid dimension: "a thick board";
 > "a thick sandwich"; or of a specific thickness: "an inch thick"

Correct answer : 2
```

From the descriptions given, it is clear that the latter sense of the adjective '*thick*' is the one used in the definition. The former applies to the viscosity of liquids.

```
Question 9: (choose referent for implicit object)

 > Choose referent for "Something that makes leathers":
 >
 >  1 The animal that skin relates to
 >  2 The thick strong skin, hides, that something uses
 >  3 Leathers
 >  4 some other entity

Correct answer : 4
```

The agent that makes leather is not any of the explicit objects 1, 2 or 3 that are mentioned in the sentence. The making of the leather is performed by something else that is not mentioned, *i.e.* some implicit entity. Hence option (4) should be chosen (as illustrated in Figure 7.8).



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

Choose referent for "Something that makes leather":

```
The animal that skin relates to
The thick skin that something uses
Leather
some other entity
```

Double click to select item

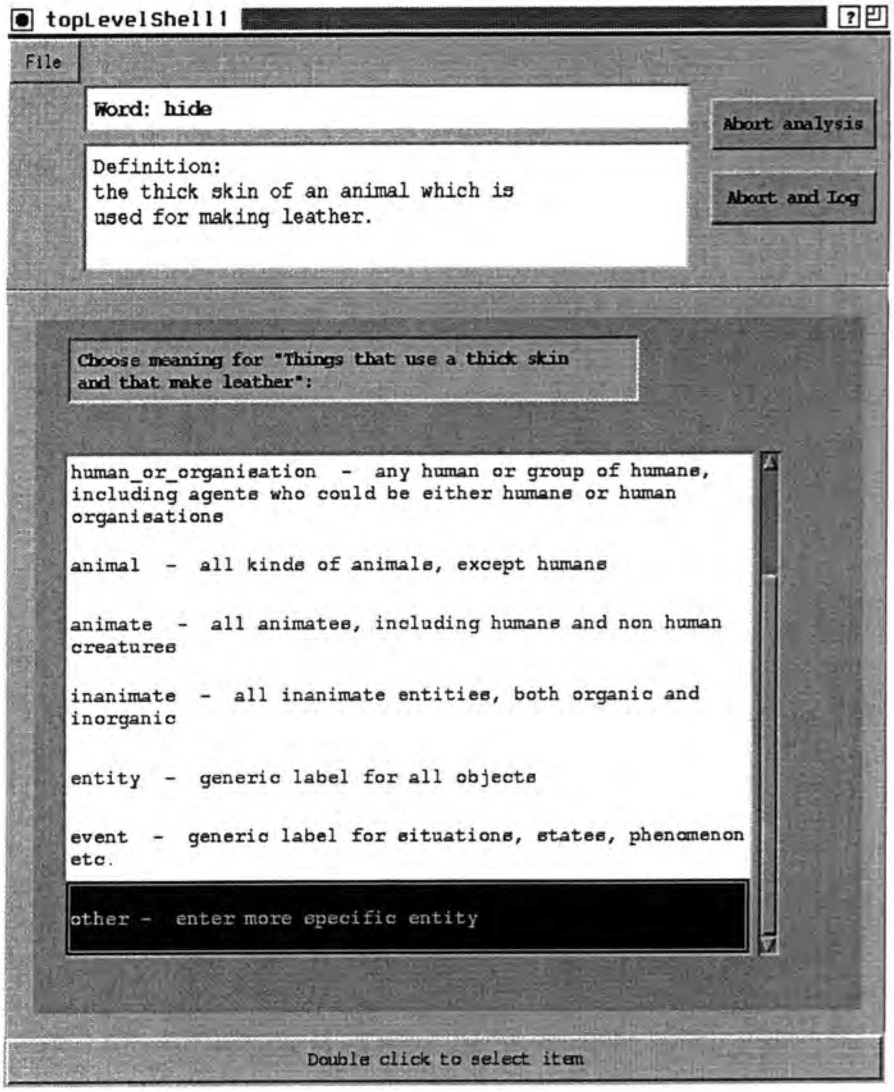
Figure 7.8: The graphical interface for selecting the referent of an implicit entity

```
Question 10: (choose referent for implicit object)

 > Choose referent for "Something that uses the thick strong skin":
 >
 >  1 The animal that skin relates to
 >  2 The thick strong skin that something uses
 >  3 Leathers
 >  4 Things that make leathers
 >  5 some other entity

Correct Answer: 4
```

The agent that uses the skin to make the leather is the same agent that makes the leather. This agent is also implicit and has not been previously mentioned in the definition. Therefore the algorithm enables the co-referencing of two implicit entities.

```
Question 11: (making objects more specific)

 > Choose meaning for
 >  "Things that use a thick strong skin and that make leathers":
 >
 >  1 human  -          any human or group of humans
 >  2 organisation -    human organisations
 >  3 animal -          all kinds of animals, except humans
 >  4 animate -         all animates, incl. humans and non human creatures
 >  5 inanimate -       all inanimate entities, both organic and inorganic
 >  6 entity -          generic label for all objects
 >  7 event -           generic label for situations, states, phenomenons
 >  8 other -           enter more specific entity

Correct answer : 8
```

The group of agents who use the skin and consequently make the leather (which involves a chemical process) are called '*tanners*'. Hence option (8) should be selected and the appropriate name entered. A snapshot of the interface which makes objects precise is shown in Figure 7.9.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

Choose meaning for "Things that use a thick skin
and that make leather":

human_or_organisation - any human or group of humans, including agents who could be either humans or human organisations

animal - all kinds of animals, except humans

animate - all animates, including humans and non human creatures

inanimate - all inanimate entities, both organic and inorganic

entity - generic label for all objects

event - generic label for situations, states, phenomenon etc.

other - enter more specific entity

Figure 7.9: The graphical interface for specifying selectional restrictions

```
Question 12:

 > Choose meaning for "tanner":
 >
 > 1)
 > Tanner (=> Forename)
 > family: propername human
 > emotional value: indifferent
 >
 > 2)
 > Tanners (=> Artisans)
 > family: job
 > emotional value: indifferent
 > level of language: common level
 >
 > 3) None of the meanings given

Correct answer: 2
```

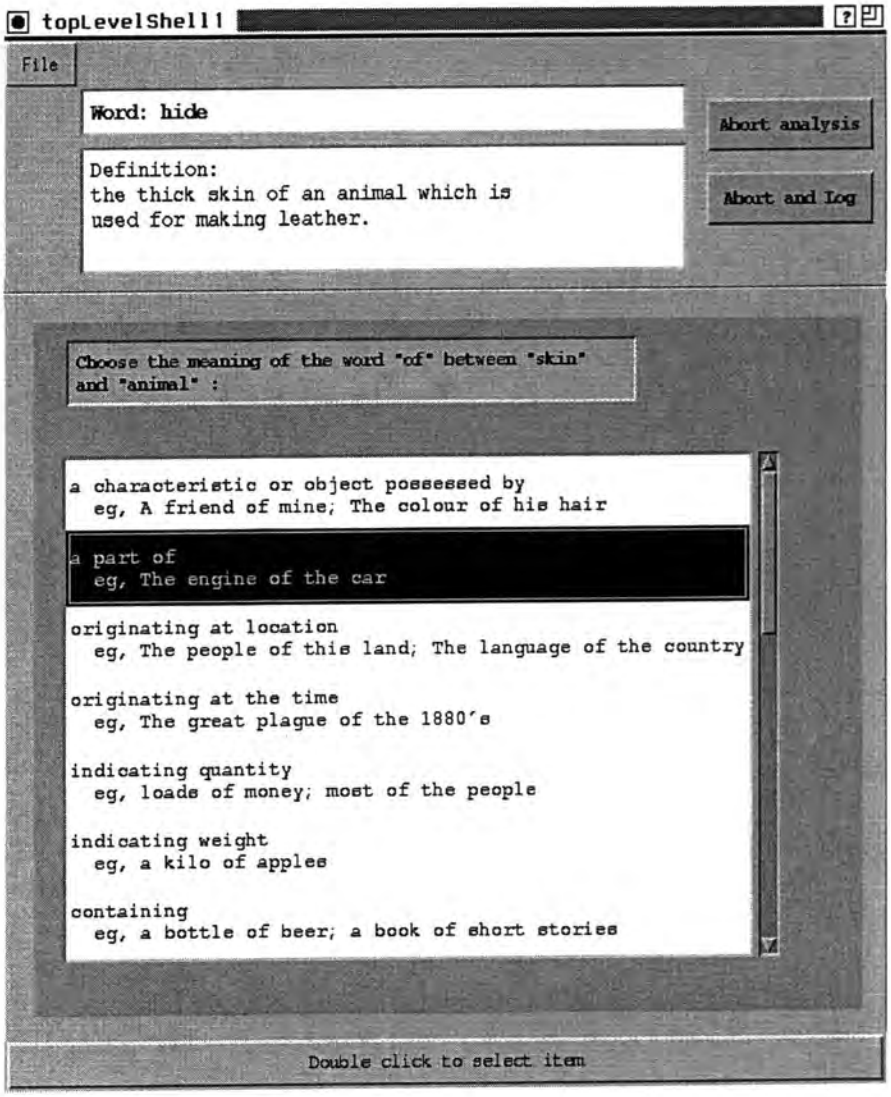
The first meaning refers to the name *Tanner*. (2) is the correct answer because it is a type of *job* which is illustrated by the semantic family of the word.

```
Question 13: (naming events and entities)

 > [a]ll tanner or [s]ome tanner?

Correct answer:  all
```

It is the job of making leather which defines a tanner. This can be checked by looking at the definition of a tanner. Hence '*all*' should be chosen.

```
Question 14: (naming relationships)

 > Choose the meaning of the word "of" between "skin" and "animal":
 >
 > 1 a characteristic or object possessed by
 >      eg, A friend of mine; The colour of his hair
 > 2 a part of
 >     eg, The engine of the car
 > 3 originating at location
 >     eg, The people of this land; The language of the country
 > 4 originating at the time
 >      eg, The great plague of the 1880's
 > 5 indicating quantity
 >      eg, loads of money; most of the people
 > 6 indicating weight
 >      eg, a kilo of apples
 > 7 containing
 >      eg, a bottle of beer; a book of short stories
 > 8 showing position
 >      eg, the top of his head; the North of England
 > 9 typical or characteristic of
 >      eg, She moves with the grace of a dancer
 > 10 describing a particular day
 >      eg, the tenth of March; first of the month
 > 11 getting in exchange
 >      eg, Roberto paid $100 for the glasses
 > 12 representing (a company, country, etc.)
 >      eg, John works for a charity; He swims for England
 > 13 in trouble
 >      eg, Gavin was in for it after that display
 > 14 NONE (leave structure and go on to next)

Correct answer : 2
```

The preposition ‘*of*’ which encodes relationship between the “*skin*” and the “*animal*” is a straightforward part-whole one, *i.e.* the skin is a part of the animal, which is indicated by option (2) above and illustrated in Figure 7.10.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

Choose the meaning of the word "of" between "skin" and "animal" :

```
a characteristic or object possessed by
  eg, A friend of mine; The colour of his hair

a part of
  eg, The engine of the car

originating at location
  eg, The people of this land; The language of the country

originating at the time
  eg, The great plague of the 1880's

indicating quantity
  eg, loads of money; most of the people

indicating weight
  eg, a kilo of apples

containing
  eg, a bottle of beer; a book of short stories
```

Double click to select item

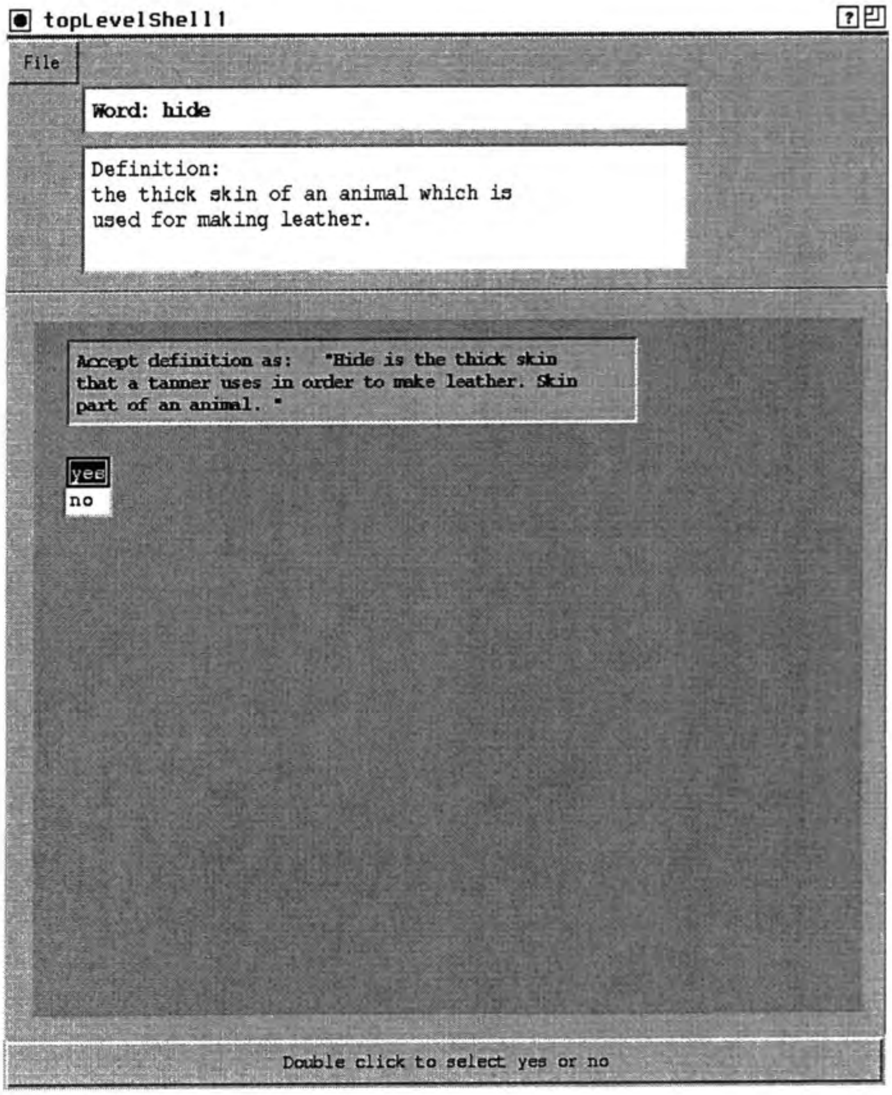
Figure 7.10: The graphical interface for disambiguating a preposition

```
Question 15: (naming relationships)

 > Choose the meaning of the word "for" between "using" and "making":
 >
 > 1 showing the length of time
 >      eg, I’m going to sleep for an hour
 > 2 showing amount of distance
 >      eg, He drove for 10 miles
 > 3 towards; in the direction of
 >      eg, They followed signs for the town centre
 > 4 intended to be given to
 >      eg, Roberto bought a toy for the baby
 > 5 for the purpose of
 >      eg, The neighbours invited us for dinner
 > 6 in order to obtain
 >      eg, He sent off for the details.
 > 7 in order to go into and travel in
 >      eg, John ran for the bus; Roberto applied for a job
 > 8 in order to achieve
 >      eg,  Kevin was trying for a first in his exams
 > 9 because of; as a result of
 >      eg, Bob was better for his weeks holiday
 > 10 compared to other similar things
 >      eg, Jane is very mature for her age

Correct answer : 5
```

The question is attempting to make the relationship between the event of "*using skin*" and "*making leather*" precise (events are often re-stated at this stage with an '-ing' ending, and with the objects omitted). The relationship is that skin is used so that leather can be made (*i.e.* making leather is a goal of using skin) which is specified by option (5) above.

```
Question 16: (confirming the analysis)

 > Accept definition as:
 >  "Hide is the thick strong skin that a tanner uses in
 >   order to make leather. Skin is part of an animal"

Correct answer : accept
```

The understanding of the definition above, although broken into two sentences, does not contain any inconsistencies with its intended interpretation and should consequently be accepted. The final screenshot for the acquisition process is shown in Figure 7.11.



Word: hide

Definition:
the thick skin of an animal which is
used for making leather.

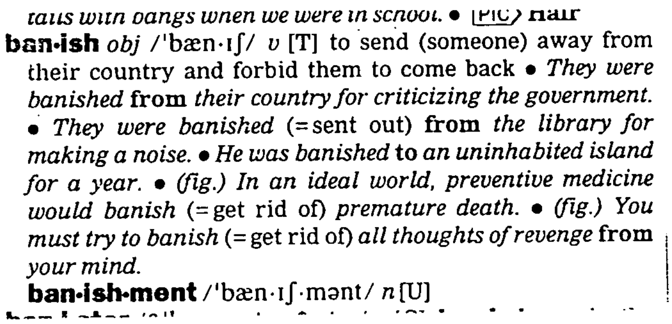
Accept definition as: "Hide is the thick skin that a tanner uses in order to make leather. Skin part of an animal. "

Double click to select yes or no

Figure 7.11: The graphical interface for confirming the analysis

7.1.2 The Verb '*banish*'

It is important that the operator has the entire dictionary entry availiable for inspection when processing the definition of a verb because the dictionary entry will often indicate (through examples or sub-definitions) the names of entities which act as preferred selectional restrictions for the verb, and also the name of the event of the verbs performance, if one exists. The CIDE entry for the verb '*banish*' is shown in Figure 7.12. Following the selection of the appropriate entry, the operator will be asked the questions shown below.



tails with bangs when we were in school. • [PIC] hair

ban·ish *obj* /'bæn·ɪʃ/ *v* [T] to send (someone) away from their country and forbid them to come back • *They were banished* **from** *their country for criticizing the government.* • *They were banished* (=sent out) **from** *the library for making a noise.* • *He was banished* **to** *an uninhabited island for a year.* • *(fig.) In an ideal world, preventive medicine would banish* (=get rid of) *premature death.* • *(fig.) You must try to banish* (=get rid of) *all thoughts of revenge* **from** *your mind.*

ban·ish·ment /'bæn·ɪʃ·mənt/ *n* [U]

Figure 7.12: The CIDE entry for the verb '*banish*'

```
Question 1: (choosing grammatical category)

 > Choose grammatical category for "banish":
 >
 > 1 Noun representing an entity
 > 2 Noun representing an event
 > 3 Verb

Correct Answer : 3
```

Similarly to the last walk-through example the question above is asked because it is a simple way to reduce the complexity of further questions. This is done by restricting the number of possible meanings of the word '*banish*'. In this case, the question is redundant because only verb meanings of '*banish*' exist.

```
Question 2: (picking word meaning)

 > Choose meaning for "banish":
 >
 > 1)
 > To banish. (=> To expel
 >              = To relegate , To bar
 >             <= To spike)
 >  emotional value: negative
 >  relation type: transitive
 >
 > 2)
 > To banish. (=> To expel
 >              = To shun , To ostracize , To ban)
 >  emotional value: negative
 >  relation type: transitive
 >
 > expel from a community or group
 >
 > 3)
 > To banish. (=> To expel
 >              = To ban
 >             <= To rusticate)
 >  emotional value: negative
 >  relation type: transitive
 >
 > ban from a place of residence, as for punishment

Correct Answer : 3
```

From the informal descriptions that accompany each meaning it is clear that the final one captures the concept in the definition closely enough. One point of contention could be the precise meaning of the phrase "their country"; is it the place where someone currently lives, or where they were born? Such fine grained distinctions are not regarded as important (because humans rarely make them) so long as the general concept of the word is captured.

```
Question 3: (entering word information)

 > To banish (=> To expel
 >            = To ban
 >           <= To rusticate)
 > emotional value: negative
 > relation type: transitive

 > Do you wish to change the controls?

Correct answer : No
```

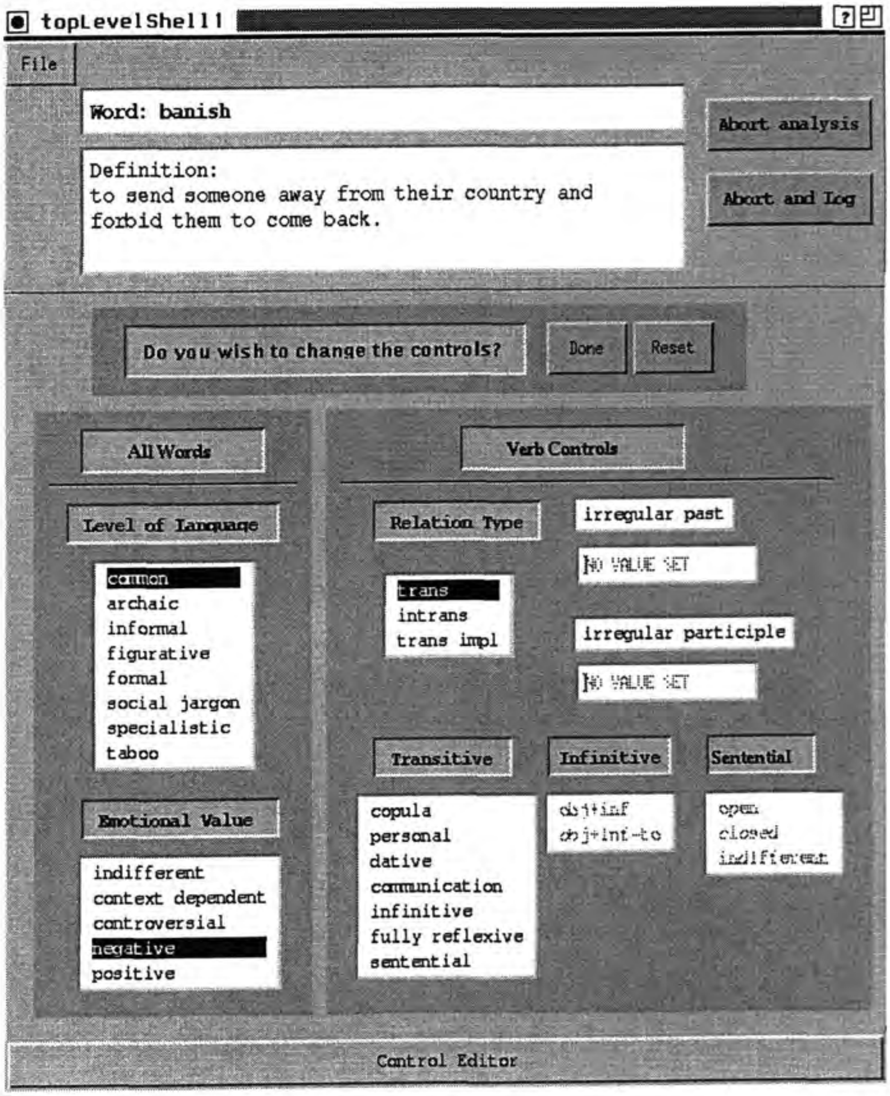
The values which are displayed are the correct ones and so the operator can simply go to the next question. The screenshot for changing controls of verbs is shown in Figure 7.13.

```
Question 4: (solving structural ambiguities)

 > Reject incorrect bracketing<s>:
 >
 > 1 to banish something (is to send someone away from their country
 >    and forbid them to come back)
 > 2 (to banish something is to send someone away from their country)
 >    and forbid them to come back

Correct answer : reject 2
```

The left hand side of the conjunction '*and*' in parse (2) is the entire fragment shown in the brackets. Consequently, interpretation (2) corresponds to the case where two independent pieces of information are being asserted. Hence this latter interpretation should be rejected.

```
Question 5: (solving structural ambiguities)

 > Reject incorrect bracketing<s>:
 >
 >  1 to banish something is to send someone away from their country
 >     and forbid (them to come back)
 >  2 to banish something is to send someone away from their country
 >     and (forbid them to come) back

Correct answer : reject 2
```

The latter case should be rejected because here the adverb '*back*' cannot modify



Word: banish

Definition:
to send someone away from their country and forbid them to come back.

Level of Language: common

Emotional Value: negative

Relation Type: trans

irregular past: NO VALUE SET

irregular participle: NO VALUE SET

Figure 7.13: The graphical interface for selecting verb controls

the verb '*come*' which is a subordinate item in the bracketed fragment.

```
Question 6: (picking word meanings)

 > Choose referent for the adverb #back# in the original definition:
 >
 >  1)
 >  Back. ()
 >  level of language: common level
 >
 >  in a previous place, condition or time;
 >  "put them back"; "she came back"
 >
 >  2)
 >  Back. ()
 >  father away in distance; "they pushed it back against the wall"; "

Correct answer : 1
```

The question requires the selection of the adverb '*back*' which modifies the verb '*come*' in the definition. The correct answer is the former sense as clearly illustrated by the informal descriptions.

```
Question 7: (finding referents for mentioned words)

 > Choose referent for #their#  in
 >  "to banish something is to send someone away
 >   from #their# country and forbid them to come back":
 >
 >  1 Something that sends away something
 >  2 Something that something sends away

Correct answer : 2
```

The former case refers to the entity that is sending someone away, and the latter to the entity that is being sent away. Intuitively (2) is the correct referent. It may be that the possessive determiner '*their*' refers to the combination of both the subject and object of banishing. This is an over-complication and since the combination of agents participating in a relationship implies that any single one must exist in the relationship, it can be ignored.

```
Question 8: (finding referents for mentioned words)

 >  Choose referent for #them#  in
 >   "to banish something is to send someone away
 >    from their country and forbid #them# to come back":
 >
 >  1 Something that away sends something
 >  2 Something that something sends away
 >  3 Something's country

Correct answer : 2
```

It is the people who are being sent away who are also forbidden to come back, and so (2) is the correct answer[4].

```
Question 9: (making objects specific)

 > Choose meaning for "Things that banish something":
 >
 >  1 human  -        any human or group of humans
 >  2 organisation -  human organisations
 >  3 animal -        all kinds of animals, except humans
 >  4 animate -       all animates, incl. humans and non human creatures
 >  5 inanimate -     all inanimate entities, both organic and inorganic
 >  6 entity -        generic label for all objects
 >  7 event -         generic label for situations, states, phenomenons
 >  8 other -         enter more specific entity

Correct answer : 8
```

It is only '*humans*' that banish people from a country, so intuitively option (1) could be chosen. However, in general, there is a more specific name for this group of people, and is called a '*government*'. Hence option (8) should be selected and the appropriate name entered.

[4] Note that the word order produced by the generator is often incorrect *e.g.* adverbs may appear in the wrong positions. This may happen because heuristics which can correct the problem do not exist, or the required knowledge in SemNet may be incorrect or missing. However, it is clear that humans are able to use their world knowledge to correctly interpret such utterances.

```
Question 10: (picking word meanings)

 > Choose meaning for "government" (return to abort):
 >
 >  1)
 >  Governments (=> Bodies
 >              = Administrations, Brasses , Establishments ,
 >                Governances , Organizations
 >             <= Governments that Prime Minister head ,
 >                White house ,
 >                The british government that meets in Cabinet Office,
 >                Managements , Officialdoms , Judiciaries , Executives)
 >  family: human organisation
 >  emotional value: context dependent
 >  level of language: common level
 >
 >  the body of persons who administer something
 >
 >  2) none of the meanings above

Correct answer : 1
```

Since there is only one meaning of '*government*' and it corresponds to the concept which is being defined (*i.e.* there are no associated words to suggest that it may be the wrong sense of '*government*'), it should be selected.

```
Question 12: (naming events and entities)

 > [a]ll government or [s]ome government?

Correct answer : some
```

Only some governments banish, because the verb *banish* does not define a government.

```
Question 13:  (making objects specific)

 > Choose meaning for #someone# in
 >   "to banish something is to send #someone# away
 >    from their country and forbid them to come back":
 >
 >  1 human  -          any human or group of humans
 >  2 organisation -    human organisations
 >  3 animal -          all kinds of animals, except humans
 >  4 animate -         all animates, incl. humans and non human creatures
 >  5 inanimate -       all inanimate entities, both organic and inorganic
 >  6 entity -          generic label for all objects
 >  7 event -           generic label for situations, states, phenomenons
 >  8 other -           enter more specific entity

Correct answer : 8
```

It is humans that are banished from a country. In particular, this group of humans is known as '*residents*' of the country (it makes no sense to banish someone that does not reside in that country). Option 8 is selected and the name "*resident*" entered as the more specific entity.

```
Question 14: (picking word meanings)

 > Choose meaning for "resident":
 >
 > 1)
 > Resident  (=> Physician)
 >  family: job
 >  emotional value: indifferent
 >  level of language: common level
 >
 > 2)
 > Residents  (=> Inhabitants
 >              =  Occupants , Occupiers
 >             <=  Spics , Townsmans , Tenants , Inmates , Welshmans ,
 >                 Ukranians  ,  Irelanders , Hindooes)
 >  family: human
 >  emotional value: indifferent
 >  level of language: common level
 >
 > 3) none of the meanings above.

Correct answer : 2
```

The first sense is the `resident-doctor` sense, and the second, the inhabitants of a place. Therefore (2) is the correct sense of the word '*resident*'.

```
Question 15: (naming events and entities)

 > [a]ll resident or [s]ome resident?

Correct answer : some
```

As with the *government* case, *residents* are not defined by being banished. Concquently, only *some* residents are banished by governments.

```
Question 16: (naming relationships)

 > Choose the meaning of the word "from" between "sending" and "country":
 >
 > 1 the place where someone or something starts
 >     eg, she sent me a postcard from Spain
 > 2 the cause of something or the reason why it happened
 >     eg, she made her fortune from investing in property
 > 3 the time when something starts, was made or first existed
 >     eg, drinks will be served from seven oclock
 > 4 showing the distance between two places
 >     eg, the hotel is 2 miles from the airport
 > 5 the origin of something or someone
 >     eg, my mother is from France and my father from Italy
 > 6 the material of which something is made
 >     eg, the desk is made from pine
 > 7 showing the level at which a range of things begins
 >      eg, tickets cost from $10 to $20
 > 8 show a change in the state of someone or something
 >       eg, things went from bad to worse
 > 9 the fact or opinions considered before making a decision
 >      eg, they found him guilty from the evidence presented
 > 10 showing someone or something has been removed, taken away or left
 >      eg, her handbag was snatched from her
 > 11 the difference between two people or things
 >       eg, his opinion is different from mine
 > 12 showing the position of something in comparison with other
 >    things or the point of view of someone when considering a matter
 >       eg, she talked from her experience;
 >           there was a nice view from the bar
 > 13 showing what someone is being protected against
 >       eg, they found shelter from the storm under a large tree
 > 14 showing what someone is not allowed to do or know,
 >    or what has stopped happening
 >       eg, he has been banned from driving;
 >           the loan saved them from bankruptcy
 > 15 NONE (leave structure and go on to next)

Correct answer : 5
```

The relationship in question is the one between the event of '*sending away*' and "*the person's country*" represented by the preposition '*from*'. The correct answer is that the country is the origin of the sending, which is given by selecting meaning (5).

```
Question 17:  (naming relationships)

 > choose meaning for the relation between "human" and "country"
 >
 >  1 uses        eg, Rick's machine
 >  2 possess     eg, the boy's books
 >  3 owns        eg, John's house
 >  4 has a part  eg, Sarah's teeth
 >  5 lives in    eg, my aunt's place
 >  6 works in    eg, the butcher's shop
 >  7 OTHER       more specific relation
 >  8 NONE        leave structure and go on to next

Correct answer : 5
```

The intended relationship between the '*someone*' (which is instantiated to a human) and the '*country*', in the definition, is that the person lives in the country. This relationship between the noun compound is specified by selecting option (5).

```
Question 18: (naming events and entities)

 > Is there a name for the process:
 >     "Some governments banish residents"?

Correct answer : yes
```

The name for the process of banishing someone is '*banishment*' which is listed in the CIDE definition (see Figure 7.12) under the main verb, '*banish*'.

```
Question 19: (picking word meanings)

 > Choose meaning for "banishment" (return to abort):
 >
 > 1)
 >  Banishments  (=> Rejections
 >                = Proscriptions
 >               <= Excommunications , Deportations , Exclusions)
 >  family: events
 >  emotional value: indifferent
 >  level of language: common level
 >
 > 2) none of the meanings above

Correct answer : 1
```

Since there is only one meaning of '*banishment*' and it corresponds to the concept which is being defined (*i.e.* there is no evidence to suggest the contrary), it should be selected.

```
Question 20: (confirming the analysis)

 > Accept definition as:
 >   "Some governments banishing residents are something
 >    sends away residents from a country."

Correct answer : accept
```

The analysis should be accepted because there are no inconsistencies in the interpretation of the definition, although many more relationships could be made explicit, *e.g.* the government governs the country mentioned in the definition. The missing information concerning the residents coming back[5] does not lead to inconsistencies and so does not affect the decision above.

7.2 Summary

This chapter has been used to illustrate the implemented knowledge acquisition process. It operates in two modes: a text mode which is initiated by direct inter-

[5] this may occur because LOLITA does not have appropriate semantics to deal with the particular constructs in the input definition.

action with the LOLITA system and, a graphical mode which is the normal mode of operation for the person participating in the question answering process.

Walk-through examples for the noun '*hide*' and the verb '*banish*' have been illustrated. The output shown is taken directly from the LOLITA system operating in text based mode. In addition, the screenshots of the GUI provide an illustration of the ease with which the system can be operated.

Chapter 8

Evaluation

The previous chapters have outlined a semi-automated approach to the analysis of dictionary definitions. The approach involves using the LOLITA NLP system to analyse the main text of a definition and extract the semantic relationships from within it. However, this text is often unstructured (particularly in the differentia of a definition) and requires the resolution of NL ambiguity.

In this thesis, the resolution of ambiguity within the definition is solved by a human operator. The use of human operators is attractive because implicit knowledge in the definition can be extracted and formalised as a by-product of the analysis process. Since the cost of using experienced linguists as disambiguators would be extremely high, the feasibility of the approach relies upon presenting the questions in a format suitable for someone with little linguistic expertise.

This chapter presents an evaluation of the approach introduced in the preceding chapters. The evaluation is divided into two parts: a qualitative part which outlines how the criteria in Chapter 2 have been given explicit consideration in the acquisition process, and, a quantitative part which evaluates the success of some of the more specific important goals of the project. The chapter concludes with a discussion of the results.

8.1 Evaluation Criteria

The long term goal of this research in relation to the parent project, LOLITA, is to construct a "rich lexicon" which can subsequently be used by LOLITA to solve particular NLP problems. The phrase "rich lexicon" means one which not only contains basic grammatical information, but also, semantic relationships between word meanings. The intention was that the semantic relationships recorded in the lexicon should be richer than basic taxonomic links such as those found in WordNet.

It was also noted (in Chapter 2) that dictionaries are a rich source of grammatical and semantic knowledge, the former type occurring in the regular fields of a dictionary entry, and, the latter type encoded within definitions. The particular sub-problem tackled by this thesis is the extraction of semantic relations from dictionary definitions. However the close coupling of syntactic and semantic analysis means that grammatical knowledge contained within a dictionary cannot be ignored.

A number of explicit criteria were listed in Chapter 2. These are reviewed in the sections below. Each section contains descriptions indicating how the particular criterion has been given consideration in the acquisition process presented in the preceding chapters.

8.1.1 Integration

Integration of a component into a large-scale NLP system should be done with a minimal amount of change to existing components. This research concerns the construction of a knowledge base for the LOLITA system.

The criterion of integration of the knowledge base has been given consideration because the representation of semantic relationships which have been extracted from the dictionary definition have been represented in SemNet, along with existing knowledge. The notion of a *prototype* has been introduced to indicate that an event in SemNet represents the total set of events of an action's performance. Consequently, any other event of the action's performance, must be an instance of the prototypical event.

This minor change to the inheritance algorithm (the creation of an instance link from prototype to instance) allows the semantic relationships extracted from the definition to be inherited from the generic to all instances of the event, by utilising LOLITA's standard inheritance rules.

This criterion is qualitative in that the degree of integration is difficult to measure. Perhaps the short time period the integration has taken is a good indicator of its success. One would expect to make changes to a large number of modules in a badly integrated system which was not the case here.

8.1.2 Correctness

In Chapter 2, correctness was seen as an important criterion because the effects of incorrect knowledge is multiplied across several levels of an inheritance hierarchy. The correctness of the analysis of a definition has been considered in various ways:

Errors During Analysis — In the worst case, the text of a dictionary definition will contain NL constructs whose analysis is beyond the current ability of LOLITA. This may result in errors during the processing of a definition. Possible sources of these errors were outlined in Section 6.12.

The incorrect analysis of a definition can be detected in one of two ways. Firstly, the operator will find that the descriptions of entities during the question answering process may be incomprehensible. If the question which the operator is asked is incomprehensible then the analysis must be rejected. Secondly, there may be cases in which an error during the analysis of a definition cannot be detected because the definition is subsequently unambiguous. The situation is salvaged by adding a final confirmation stage (see Section 6.12) to the analysis.

The detection of wrongly interpreted definitions relies upon the LOLITA generator [Smith, 1996] to provide a description of the semantic relationships which have been extracted from the original definition.

Dictionary Errors — another potential source of errors are inconsistencies in the dictionary entries. An example is:

administer$_v$ — to cause someone to receive something

The detection of these types of errors has been automated. This is possible because the semantic relationship '`X IS-A Y`' implies that the structure at node `X` can be inherited to `Y`. Since the types of these structures are specified as selectional restrictions during the acquisition process they can easily be checked.

In the example above, it is inconsistent to view "*someone administer something* IS-A *someone cause something*" because the object of the verb *cause* can only be an *event* while the object of the verb *administer* can only be an *entity*.

A mechanism has been added to the system which gives the operator the opportunity, at any stage in the question-answering a process, to reject the analysis of the definition. The assumption that the operator will be able to correctly detect cases of incorrect analysis is evaluated quantitatively in the next section.

8.1.3 Scale and Feasibility

Semi-automated acquisition procedures on a large scale could be seen to be unfeasible because of the cost associated with using paid disambiguators. Consequently, automated approaches to extracting definitions have been considered [Slator and Wilks, 1990][Alshawi, 1989]. However, these automated approaches can only have limited success because they rely upon the relationships in a definition being written in particular forms which are easy to analyse. Since lexicographers have no such constraints, it is not surprising to find that the results (which could consist of the relationships which have been extracted, or even, some evaluation of accuracy) are not readily available.

In our opinion, a semi-automated approach is the best that can be currently achieved, given the state of the art in NLP. The feasibility of a semi-automated approach can be increased in the following ways:

1. by utilising an NLP system to handle the details of the acquisition process. Humans are extremely good at interpreting and disambiguating language, but error prone when it comes to handling fine detail or large quantities of data. Computers are the opposite, and so, it is natural to use an NLP system to handle the fine details of the processing and utilise the operator as they are needed. In this case, the details of the acquisition process, handled by LOLITA include:

   - using SemNet to represent the extracted semantic relationships in an unambiguous form,
   - identification of implicit entities in the definition text,
   - handling of WSA by identifying and limiting the interpretation to those which occur in the CIDE defining vocabulary.

2. by presenting the ambiguities in a form which requires the minimal amount of linguistic expertise of the operator.

3. by using a dictionary which is suited to computational analysis. The advantageous features of CIDE include the use of a defining vocabulary, and the novel layout of entries:

   - The CIDE defining vocabulary is used to restrict the number of possible interpretations of a sentence. See Section 6.5 for further details.

- The layout of entries means that fewer definitions need to be analysed because the definitions of semantically related concepts can often be processed together. See Section 6.9 for further details.

If the approach is feasible then it implies that the knowledge can be acquired on a large scale from CIDE. The feasibility of the approach mainly hinges upon the degree of success of (2). This is evaluated quantitatively in the section below.

8.1.4 Testing

The acquisition process has been implemented and integrated into a fully working NLP system. There was no need to assume the existence of particular components. Therefore the testing criteria mentioned in Chapter 2 are not relevant to this project.

8.2 A Quantitative Evaluation

The quantitative part aims to evaluate the following:

Correctness — errors in the acquisition process can be presented in such a form as to allow an operator to reject incorrectly analysed definitions.

Feasibility — ambiguities are presented in a form that will allow humans with a minimal amount of linguistic training to participate successfully in the question-answering acquisition process.

An evaluation has been designed with these two requirements in mind. An outline of the set-up is presented below.

8.2.1 The Evaluation Set-Up

A tracing mechanism has been added to the GUI which records each session of analysis by an operator. The mechanism is based upon two abort buttons which are located on the GUI: the first allowing the operator to abort an analysis and try again if they feel they have made an error, and the second allowing them to abort the analysis of a definition which they feel has been analysed incorrectly by LOLITA. Analysis which has been rejected using the former abort button is not included in the evaluation.

In the former situation, only the answer given in the final analysis of the particular definition is evaluated.

The definitions of 50 words (30 nouns and 20 verbs) were entered into the GUI. The list can be seen in the initial GUI screen shown in Figure 7.1 (p 138). The definitions of these words are shown in Appendix B. The particular words have been chosen for the following reasons:

- the analysis of their definitions covers the full range of questions introduced in Chapter 7. For example, the list contains:
  - verbs, nouns classified semantically as entities, and nouns classified semantically as events,
  - definitions which have named selectional restrictions for the objects which occur within them,
  - verbs which are actions of named events,
  - definitions containing implicit entities which are co-referential,
- their analysis presents a wide range of problems for the analysis process, *i.e.* amongst the definitions that can be processed successfully (so as not to violate the criteria laid out in Chapter 2) the set includes:

- definitions which produce more than a single acceptable parse and definitions which fail to produce any acceptable parse,
- headwords which do not have an obvious WordNet mapping, and headwords which have no WordNet mapping,
- definitions which produce entirely incorrect parses which cannot be detected by the bracketing algorithm,
- definitions containing prepositions which clearly represent a given semantic relationship, and those that represent relationships not in the list given to the operator,

and so on. The evaluation presented below groups the answers into many of the fine-grained categories shown above.

The evaluation was conducted using a group of 10 operators each having little linguistic expertise. The operators were software engineers most of whom have a B.Sc. in Computer Science. The question-answering process was done in isolation through the GUI. Each operator was asked to read the user and interface manuals (shown in Appendix C & D) before they were tested. Each definition was analysed twice by different operators (an average of 10 definitions each) so that differences in their answers could be analysed. In addition, all the questions were answered by an expert (*i.e.* myself) so that the types of questions and expected answers could be classified in detail.

The idea is that each question which is posed to the operator represents a choice point in a tree of potential outcomes in the analysis of a single definition. At each point the operator can provide a correct or incorrect answer. The ratio between correct and incorrect answers provides a measure which can be used to judge the success of the question-answering process. However, the situation is a bit more complex because the notion of a correct answer is not so clear cut. In general, there is no guarantee that the set of potential answers is disjoint. In the extreme case a number of the potential answers may exist, each one being semantically correct but less informative than the next.

Given that correctness is an explicit goal of this evaluation, the notion of a correct answer shall be taken to be one that will not result in an incorrect semantic relationship, *i.e.* it does not belong to a set of answers deemed to be incorrect. Where possible, the ratios given below are divided into fine-grained categories which explain the various classes of answers.

8.2.2 Mapping the Original Word Meaning

This set of results reports upon the initial stage of the question-answering process in which an operator has to identify the SemNet concept (the target concept) which corresponds to the defined headword (the source concept). Although this task is a case of WSD it is treated separately from WSA occurring in the main text of the definition because of its importance; the target concept acts as the hook to which semantic relations from the definition are attached. These relationships will subsequently be inherited to all specialisations and instances of the target concept.

A summary of the results is shown in table 8.1. The table consists of two major categories of results: those mappings which have been classified as correct and those classified as incorrect. Each major category is divided into finer grained categories which indicate the precise meaning of "a correct mapping" and "an incorrect mapping" respectively.

CORRECTLY CLASSIFIED	
SemNet meaning identified	77%
no SemNet meaning identified	2%
more specific SemNet meaning identified	9%

INCORRRECTLY CLASSIFIED	
SemNet meaning not identified	4%
wrong grammatical category chosen	5%
incorrect SemNet meaning identified	3%

Table 8.1: A breakdown of results for the initial word sense selection task.

For each source concept, a correct decision can be classified as being one of three kinds:

SemNet meaning identified is the case in which an equivalent target concept exists in SemNet, and has been correctly identified by the operator.

no SemNet meaning identified is the case in which an equivalent target concept in SemNet does not exist, and has been identified as such by the operator.

more specific SemNet meaning identified is the case in which the target concept which has been selected by the operator, is more specific than the source concept. For example, the definition of the verb '*administer*' is:

administer — to cause someone to receive something

has been mapped to the target SemNet concept:

```
> To administer  (=> To doctor
>                  = To dispense
>                 <= To inject)
>   emotional value: context dependent
>   relation type: transitive
>
>  of medications
```

this target is more specific than that intended by the definition because of the restriction on the object of the verb.

The mapping is not classified as incorrect because the semantic relationships in the definition correctly apply to the restricted notion of the verb '*administer*' above, *i.e.* no inconsistencies will result the mapping above.

The figures in the top part of Table 8.1 are to be interpreted as follows: (1) the operator correctly mapped a source concept to an equivalent target 77% of the time they performed a mapping between a source and a list of potential targets, and, (2) the operator correctly identified the non existence of a target concept 2% of the time they were asked to perform the mapping between a source and a list of potential targets, and, so on.

In conclusion, when asked to perform a mapping between a source concept and a list of potential targets, an operator made a correct decision 88% of the time.

For each source concept, an incorrect decision in the mapping process can be classified as being one of three kinds:

SemNet meaning not identified is the case in which an equivalent target concept does not exist but the operator has nevertheless selected one.

wrong grammatical category chosen is the case in which the operator has selected the incorrect grammatical category for the headword, *e.g.* event noun instead of verb, verb instead of noun, *etc.*

incorrect SemNet meaning identified is the case in which an equivalent target concept exists, but the operator has selected a different one.

The figures in the bottom part of Table 8.1 are to be interpreted as follows: (1) the operator failed to identify the correct target concept 4% of the time they were asked to perform the mapping between a source and a list of potential targets, and, (2) the operator selected the incorrect grammatical category 5% of the time they were asked to identify the grammatical category of the source concept, and, so on.

In conclusion, the operator made an incorrect decision 12% of the time in which they had to identify a mapping between a source and target concept.

It is estimated that 25% of the operator errors will be eliminated when the grammatical category category of the dictionary entry (*i.e.* noun or verb) is processed automatically. This would mean that a realistic accuracy is 91% of questions answered correctly and 9% incorrectly.

8.2.3 Parse Tree Selection

This is the stage in which the operator is asked a series of questions. Each question involves the operator rejecting interpretations of the definition which violate their understanding of its meaning. The interpretations are indicated by the bracketing of different segments of the original definition.

The set of questions which comprise the interpretation of a single definition form a unit because all of them need to be answered correctly, if the correct interpretation (*i.e.* parse tree) is to be found. Consequently, one or more errors within a set of questions which comprise the analysis of a single definition carry the same penalty.

The results of the evaluation showed that:

1. 71.5% of the parse tree selection decisions were made correctly. This number represents a combination of the situations in which there exists a correct interpretation[1] and also cases in which all the interpretations are incorrect and consequently need to be rejected.
2. 28.5% of the parse tree selection decisions were made incorrectly by either rejecting the correct interpretation, and/or accepting one which should have been rejected.

Although the proportion of incorrectly answered questions may at first glance seem high, it should be noted that a single operator (out of the 10 that took part) was responsible for 40% of all the errors.

In addition, a source of confusion arises with interpretations of a common form of bracketing structure. Examples are given in the interpretation of '*abattoir*' shown in Section 6.4. The operator is expected to consider the bracketings (the interpretation of which is explained in Section 6.3):

[1] a correct interpretation is one in which none of the interpretations which are presented in the form of questions violate the correct understanding of the text. However, it does not necessarily guarantee that the resulting parse tree represents the correct interpretation.

- (each abattoir is a place) where animals are killed for their meat

However, bracketings such as the ones above can be automatically removed because the structure of the outermost term of a parse is predictable, *i.e.* the verb '*is*' must represent the outermost verb in the sentence.

It is estimated that the elimination of answers from the operator who performed particularly badly, together with the filtering of the problem parses shown above will result in an improvement up to 86.2% correct and 13.8% incorrect answers.

8.2.4 Word Sense Disambiguation

The CIDE defining vocabulary contains 2000 words. However many of these words are used in more than a single sense. The operator will therefore need to disambiguate the senses of ambiguous words used in the definitions. This task is easier than the mapping task evaluated in Section 8.2.2 because all the alternative word meanings are guaranteed to have a descriptive gloss (see Section 6.5).

The results of the evaluation showed that:

- 88.5% of the decisions correctly matched the choices that were expected.
- 9.5% of the decisions were in a category in which the correct word sense was judged to be difficult to decide. An example is the two senses of the verb '*can*' which are listed in the CIDE defining vocabulary:

  can ABILITY — to be able to • She can speak four languages • Can you read the sign from this distance? • The doctors are doing all they can

  can POSSIBLITY — used to express possibility in the present, although not in the future • You can get stamps from the newsagent • Smoking can cause cancer • He can be really annoying

The boundary between the two senses is less than clear. Consider the example "*you can get stamps from the newsagents*" does this imply that "*you have the ability to get stamps from the newsagents*"? If so, then the two senses are clearly related in some cases. If not, then each of the definitions is too vague.

- 2.1% of the decisions were incorrect.

It is clearly unreasonable to expect a human with little lexicographic experience to distinguish between word senses that an experienced linguist would have difficulty with. Therefore the category of difficult cases above are not seen to be as important as those with a clear cut division between word meanings. Therefore 97.7% of the answers that were given are considered to be correct.

8.3 Anaphora Resolution

This is the stage in which the operator is asked to choose the correct antecedent from a list of potential referents of a definite pronoun.

It was found that in 85.7% of pronouns were solved correctly. Out of the remaining cases, 12.5% were cases where the antecedent could be correctly viewed to be one of a number. These cases represent genuine ambiguity in the definition. An example was illustrated in the definition:

banish$_v$ — to send someone away from their country and forbid them to come back

in which the referent of the possessive determiner '*their*' could be the thing that is banishing something, the thing that is being banished, or both. Therefore it is reasonable to view the 12.5% of the cases in this grey area to be correct. This would mean that appropriate antecedents were found in 98.2% of the cases.

8.3.1 Identifying Implicit Entities

The handling of implicit entities is similar to that of definite pronouns. The difference is that a default is added to the list because the entity may not have been mentioned in the previous context.

The results of the evaluation showed that:

- 81% of the questions were answered correctly.
- 2.4% of the questions were given referents that were not as precise as they could be. In other words the operator selected the default when an alternative correct entity was present in the list of referents extracted from the definition.
- 16.6% of the questions were answered wrongly.

The category of correct answers above contains questions which may have more than one correct answer in the list of referents. For example, in the definition:

jack $_n$ — a piece of equipment which can be opened slowly to allow heavy weights to be raised

the implicit entity which is raising the heavy weight could be correctly viewed to be '*humans*' or '*the piece of equipment*' depending upon the granularity of the view that is taken. Different views in this way commonly occur when two agents co-operate to achieve some goal, *e.g.* if troops were ordered by the commander to kill the civilians, then the troops or the commander may be viewed as the killer. Plausible shifts in granularity such as these are discussed in [Poria and Garigliano, 1997][Poria and Garigliano, 1998]. Suffice it to say that either answer is viewed to be reasonable.

8.3.2 Specifying Restrictions

This is the class of questions which aims to make the types of objects precise. The source of questions in this category are implicit entities (which have no definite referent in the definition) and indefinite pronouns. The procedure consists of choosing from a list of standard referents. In the case of a named restriction (see Section 6.8), a further question is asked to the operator which prompts them for the name.

The results of the evaluation showed that:

- 81.5% of the questions were answered correctly.
- 11.1% picked restrictions which were more general than than they could have been. For example, picking '*human*' instead of '*butcher*' in the definition of abattoir, and, '*entity*' as the object of the verb '*bake*' instead of '*food*', *etc.*
- 7.4% of the questions were answered incorrectly.

Restrictions which are too general can sometimes be eliminated by adding a simple checking mechanism. For example, in the definition:

crush$_v$ — to defeat someone completely

'*animals*' were given as a restriction for the object of the verb. It is clear that the pronoun *someone* occurring in the definition implies that this restriction must be '*human*'. Consequently the list of referents offered to the operator must be dependent upon the category of the pronoun which has been used in the definition.

8.3.3 Disambiguating Prepositions

The operator is presented with a list of potential meanings for each preposition in the definition. The results of the evaluation showed that:

- 55.7% of prepositions were correctly disambiguated to a concrete relationship chosen from the list that was offered to the operator.

- 12.8% of prepositions were left ambiguous. This might occur because the operator simply overlooked the correct relation, did not understand the relationship being specified, or simply because the correct relationship was missing. Examples of the latter category include the PPs underlined below:

  jack$_n$ — a piece of equipment which can be opened slowly ...

  industry$_n$ — the people and activities involved in one type of business

  in which the relationship specified by the preposition ‘*of*’ is not identified in CIDE.

- 4.3% of the prepositions were disambiguated to relationships which were less precise than they could have been. For example, in the definition:

  hydrogen$_n$ — the lightest gas with no colour, taste or smell, that combines with oxygen to form water

  the relationship between the ‘*combining*’ and the ‘*oxygen*’ could be correctly disambiguated to the relationship ‘*combining in the presence of oxygen*’ or more precisely, the relationship ‘*combining using oxygen*’.

- 8.6% fell into a category in which it was unclear that the relationship specified was correct. In many cases, the relationship will depend on the precise word meanings used to express the relationship. For example, in the definition:

  chaplain$_n$ — a Christian official who is responsible for the religious needs of an organisation

  file$_n$ — to walk as a long line of people, one behind another

  is it correct to say that ‘an organisation possesses needs’, or that ‘a long line has a part people’? The answer depends upon the particular semantics of the relationship ‘*possess*’ and ‘*has a part*’ respectively.

- 18.6% were disambiguated to incorrect semantic relationships.

The first three categories of answers are all viewed as being acceptable. One would expect many relationships to be unclear because of the approximate nature of word meanings.

The numbers of errors could be reduced by introducing a semantic classification of prepositional objects. For example, consider two possible relationships for the meaning of the preposition '*for*' in the phrase '*animals are killed for their meat*' which occurs in the definition of an '*abattoir*':

```
>  Choose the meaning of the word "for" between "killing" and "meat" :
>
> 2) ....
>      ....
> 3) for the purpose of
>      eg, The neighbours invited us for dinner
> 4) in order to obtain
>      eg, He sent off for the details.
> 5) ....
```

Some operators chose relationship (3) (*i.e.* killing for the purpose of meat) instead of the correct answer (4) (*i.e.*killing in order to obtain meat). The confusion may arise because both relationships incorporate a GOAL relationship. However, interpretation (3) is coherent only if the prepositional object is an event, and the latter relationship is coherent only if the prepositional object is an entity. The checking of types of prepositional objects in this way can help to eliminate errors because irrelevant prepositional meanings (*i.e.* in the case of type mismatches) can be hidden from the operator.

The specification of a set of semantic categories which can act as restrictions for prepositional objects would involve a considerable amount of analysis due to the large number of prepositional meanings in CIDE[2]. The categories would need to be

[2] Slator [Slator *et al.*, 1990a][Slator *et al.*, 1990b] describes just such an endeavour with LDOCE using a entire team of linguists.

detailed enough to distinguish as many prepositional meanings as possible. This would require further empirical investigation.

8.3.4 Disambiguating Compounds

These questions involve finding the relationship between a number of adjacent NPs in the input, or a possessive determiner followed by an NP. The evaluation showed that:

- 71.4% of the questions were answered correctly.
- 14.3% of questions were given answers which were more specific than they needed to be, *e.g.* the relationship between the possessive determiner '*their*' and the NP '*usual surroundings*' in the definition:

  hideaway $_n$ — a place where someone goes when they want to relax and get away from their usual surroundings

  was disambiguated to be the relationship '*lives in*'. This is more specific than the correct answer because a person may want to get away from the place at which they work.
- 14.3% of the questions were answered incorrectly.

8.3.5 Confirming the Analysis

This is the final stage in the acquisition process. It is intended as a check in order for the operator to identify cases in which the definition has been incorrectly processed. Since the operator has the opportunity to reject the analysis at any previous stage, the set of figures in Table 8.2 presents a more general evaluation than the processing of just the final question.

Table 8.2 contains two major categories of results: those analyses which have been handled correctly and incorrectly respectively. The categories are subdivided to give a more detailed picture.

CORRECTLY CLASSIFIED	
rejected during analysis	13%
rejected at confirmation	32.5%
accepted at confirmation	36.4%

INCORRRECTLY CLASSIFIED	
rejected at final confirmation	2.6%
accepted at final confirmation	15.5%

Table 8.2: A breakdown of results for accepting or rejecting the analysis of a definition.

A correct classification is viewed to belong to one of three categories:

rejected during analysis is the case in which an operator has correctly decided that the analysis of a definition is wrong before the final confirmation stage.

The incorrect analysis of a definition is made apparent in a number of ways during the question-answering process. Firstly the description of entities provided by the generator may be incomprehensible, or secondly, the question may ask the operator to disambiguate a relationship (*e.g.* as specified by a preposition) between two unrelated entities.

rejected at confirmation is the case in which the analysis of a definition is correctly rejected because an inconsistency is present in the NL description of the extracted semantic relationships.

accepted at confirmation is the case in which the definition is correctly accepted because no inconsistencies are present in the NL description of the extracted semantic relationships.

An incorrect classification is viewed as belonging to one of two categories:

rejected at final confirmation is the case when a correct analysis of a definition has been rejected at the final confirmation stage.

accepted at final confirmation is the case when an incorrect analysis has been accepted at the final confirmation stage.

The figures in top part of Table 8.2 are to be interpreted as follows: (1) the operator correctly rejected the analysis of a definition during the main part of the acquisition process in 13% of the cases they were asked to analyse a definition, and, (2) the operator correctly rejected the analysis of a definition, at the confirmation stage, 32.5% of the time they were asked to analyse a definition, and, so on.

In conclusion, an operator made a correct decision regarding whether to accept or reject the analysis of a definition in 81.9% of the cases.

The figures in the bottom part of Table 8.2 are to be interpreted as follows: (1) an operator incorrectly rejected the analysis of a correctly analysed definition 2.6% of the time they were asked to analyse a definition, and, (2) an operator incorrectly accepted the incorrect analysis of a definition 15.5% of the time they were asked to analyse a definition.

In conclusion, an operator made an incorrect decision regarding whether to accept or reject the analysis of a definition in 15.5% of the cases. Although 2.6% of the questions were answered incorrectly they do not result in inconsistent knowledge being acquired although the semantic relationships will be lost.

8.4 Discussion

A summary of the results is shown in Table 8.3. The column of projected results is an estimate of the increase in precision following some of the simple changes mentioned in the respective sections. The tasks with a '—' in the latter column indicates that it represents a figure which is optimal or near optimal. A '??' symbol means that projected results for the task are difficult to calculate.

TASK	ACTUAL RESULT	PROJECTED RESULTS
Mapping Word Meanings	88%	91%
Parse Tree Selection	71%	86.2%
Word Sense Disambiguation	97.9%	—
Pronoun Resolution	100%	—
Identifying Implicit Entities	83.4%	??
Specifying Restrictions	92.6%	94.4%
Disambiguating Prepositions	82.4%	??
Disambiguating Compounds	85.7%	??
Confirming the Analysis	81.9%	??

Table 8.3: A summary of correct results for the various tasks.

The results achieved in the evaluation reinforce the view that the approach is feasible with questions being presented in a form capable of being answered by untrained operators. An average of 87% of all questions were answered correctly. In addition, the explicit requirement that the detrimental effects of incorrectly analysed definitions be minimised has also been demonstrated with 82% of processed definitions being correctly accepted or rejected.

There are a number of reasons to believe that, in reality, the results would be far better:

1. the operators chosen for the final evaluation would be more motivated at trying to understand the process as they would be paid for the task.

   There was plenty of evidence to suggest that the operators taking part in the evaluation above were not as motivated as they could be. One example is that many of the questions which were used as examples in the user manual were answered incorrectly (even though the correct answer was given).

2. the final acquisition will be conducted using only selected operators. Those that consistently produce wrong answers can be filtered out during an initial evaluation period.

3. the operators would receive guidance from a supervisor in the case of a particularly difficult question.

   We envisage that the setup would consist of a number of operators who are processing definitions in parallel with the aid of a linguistic expert acting as a supervisor.

4. the operators who took part in the evaluation presented above did not receive any form of active training. In reality, it is feasible to provide operators with a minimal amount of training to ensure that they have a basic understanding of the decision processes involved.

5. The acquisition process could be organised so that the definitions of semantically related items (*e.g.* particular sub-domains) are processed at the same time because it is likely that these definitions would share particular structure and style. Consequently, dealing with similar phenomena may lead to increased accuracy.

The only true evaluation for NLE products (see Chapter 1) is the market place. There must be a demand for the product. Consequently, perhaps the strongest evidence that the research outlined in thesis is successful is that it will be extended in the near future by a company specialising in NLP products.

8.5 Summary

This chapter has shown that the criteria of success laid out in Chapter 2 have been met by the approach presented in this thesis. The evaluation consisted of two parts:

a qualitative part which checked that the acquisition process met the methodological criteria laid out in Chapter 2, and, a quantitative part which evaluated the degree of success of the key criteria.

The results of the evaluation fully support the view that operators with minimal linguistic experience could successfully participate in the acquisition process.

Chapter 9

Conclusions and Future Work

9.1 A Summary of the Aims and Approaches

In order to solve many non-trivial language comprehension tasks (*e.g.* co-reference resolution, PP attachment, *etc.*) an NLP system must possess many different kinds of knowledge: grammatical knowledge, syntactic knowledge, encyclopedic knowledge, *etc.* The component of an NLP system which contains knowledge about individual words is known as the *lexicon* of the system. The lexicon is seen to be an important component of a KB because it forms the foundation upon which layers of richer knowledge can be added.

Dictionaries have long been seen as potential sources of lexical and real world knowledge. Not only do they provide the diverse range of knowledge required within a lexicon of an NLP system, but they are also well structured for taxonomic organisation. A lot of the research in computational lexicography aims at extracting (and representing) the grammatical knowledge contained in MRDs. The largely implicit but potentially richer knowledge present in the definitions of words has proved more challenging.

The problem with extracting semantic relationships from the main text of a definition is that it requires the resolution of NL ambiguity. Two alternative approaches are:

- to use human linguists to extract the semantic relationships contained within the text and represent it in an unambiguous KRL, such as a semantic network.

  Amongst other things, this task would involve disambiguating word senses, finding referents for pronouns, and require an understanding of the intricacies of the KRL.

- to use an automated approach. This would require designing and implementing algorithms to resolve the ambiguities outlined above.

The problem with the former approach is that humans lack the consistency that is required of the acquisition process. In addition, the cost of analysing 50,000 definitions by hand would be far too great.

The great strength of an automated approach is that the consistency which is lacking in the manual approach is acquired for free. However, the decisions regarding ambiguities in the input, which a human is able to resolve so easily, pose a major obstacle in an automated framework. The resolution of most types of NL ambiguity, such as pronoun resolution and WSD, requires semantic knowledge. However this semantic knowledge is precisely what is being acquired and is consequently unavailable when it is required, *i.e.* when the definition is being processed.

The solution presented in this thesis is to use a semi-automated approach. The approach combines the strengths of each of the strategies: the ability of a human to use their store of world knowledge in order to determine the correct interpretation of the input, and, the consistency resulting from using the computer in order to build representations of complex and inter-related semantic relationships.

The cost of a semi-automated approach can be overwhelming when 50,000 definitions need to be processed. It can be minimised if operators with minimal linguistic expertise could participate in the question-answering process. This requirement has been a major consideration in the design process of the algorithm which forms the core part of the research in this thesis.

9.2 Contributions of The Research

The research in this thesis presents a semi-automated approach which enables the extraction of unambiguous semantic relationships from dictionary definitions in a feasible way. The algorithm involves a number of stages, each of which require the intervention of human operators to resolve the ambiguities and formalise the implicit knowledge contained within a definition. During the design of the algorithm various issues have been resolved. They form the core of the contribution of this research:

- A feasible approach to the extraction of correct semantic relationships was introduced.

  In our opinion, previous approaches to the extraction of semantic relationships fail to address important issues. Firstly the amount of resources required to analyse an entire dictionary of definitions by hand would be far too great. Therefore, the approach of Amsler and White (Section 3.4) would, according to our criteria, fail to address the issue of feasibility. In addition, Slator and Wilks (Section 3.4.1.2) and Alshawi (Section 3.4.1.1) do not provide any evidence to suggest that incorrect interpretations of definitions can be prevented or detected.

  The evaluation of the approach introduced in this thesis supported our view that a semi-automated approach to knowledge acquisition is a feasible compromise between a manual approach, which would require trained linguists, and a fully automated approach, which would undoubtedly result in many incorrect interpretations during the course of the acquisition process.

  It is hardly surprising to find that the weakest results occur when a computer is asked to perform human-like tasks (*i.e.* to generate a NL description of semantic relationships), and when humans are asked to perform tasks well suited for computer programs (*i.e.* picking from long lists of meanings as with the disambiguation of prepositions). These extreme cases stretch the paradigm of this thesis which is to let both participants in the acquisition process use the abilities to which they are best suited.

- The problems of using dictionaries as knowledge sources for NLP systems were identified and solutions to the problems were incorporated into the framework.

  Although there is little doubt that dictionaries contain a wealth of knowledge, they are ultimately written for human readers.

  Although previous approaches have extracted semantic relationships from dictionary definitions (see Section 3.3) they have failed to consider many of the important requirements that NLP systems place on their knowledge bases.

  The features of dictionary definitions which make them less than ideal for NLP purposes (*e.g.* the encoding of shallow hierarchies) have been identified. In addition, it was shown how these problems can be solved as a by-product of the semi-automated acquisition process. This solution provides the framework upon which subsequent work is based.

- To our knowledge this research presents the first attempt to extract semantic relationships from within CIDE.

  Until now it was mainly LDOCE that has been used in computational lexicography and NLP research. However, the recently published CIDE has many innovative features which can aid the knowledge acquisition process. It has been shown throughout the course of the thesis how the innovative features of CIDE (in particular the novel layout of entries) can be exploited during the acquisition process.

- It was shown how to transform the different types of dictionary entries allowing them to be analysed by an NLP system.

Dictionary entries are not in a form which allows direct manipulation (*i.e.* the analysis and extraction of semantic relationships) by an NLP system. The approaches introduced in Chapter 2 implement specific algorithms and incorporate domain specific rules to process headwords and their definitions. This is unnecessary because the dictionary entry can be transformed into an utterance which can be analysed by an NLP system. The transformations are such that they enable implicit knowledge contained within the entry to be extracted as a by-product of the acquisition process.

9.3 Future Work

There is still a good deal of work to be done before this research is at a stage where an entire dictionary, together with the implemented system, could be given to a group of operators in order to extract and formalise the knowledge within it. Some open problems and future work in the context of this research are listed below:

- The semantics and the representation of some constructs which occur frequently within definitions need to be formalised. For example, the phrases "especially X" and "such as X" are used to indicate restrictions on a set.

- The number of prepositional relationships offered to the operator needs to be kept to a minimum. This can be achieved by analysing the types of restriction on prepositional objects. Since there are many prepositional senses, tools may be required to permit this investigation.

- We envisage a number of operators processing definitions in parallel. There are a number of large-scale engineering issues to be resolved.

  For example, ideally, a definition should be processed in the context of previously entered definitions because selectional restrictions are imposed by existing knowledge. This will simplify question answering because some candidates for the antecedents of various questions may be eliminated.

However, this process is not easy to handle in the case of multiple operators, not only because knowledge will be entered in parallel, but also because of the need to maintain consistency. One must consider a case where a mistake is found in the acquisition of a definition and needs to be undone. If the process is such that subsequent definitions have been entered in the context of the mistake, then there may be a need (depending upon the nature of the dependency) to delete all subsequently entered knowledge.

- There are a number of possibilities regarding the order in which the definitions contained within a dictionary should be acquired. Some issues associated with the various options are:

  Generic abstract definitions first — processing the definitions of concepts which exist at the top of each IS-A hierarchy is a reasonable strategy because the semantic relationships which have been acquired can be used as consistency checks during the processing of more specific definitions.

  Specific concrete definitions first — another plausible strategy is to process the definitions of concepts belonging to particular sub-domains, *e.g.* the hierarchy whose top node is the concept of *vehicle*, *furniture*, *food etc.* These hierarchies will pose fewer problems than more abstract concepts whose definitions are less precise.

  Defining Vocabulary first — the last strategy is to initially process the definitions of words contained in the defining vocabulary because they are used to provide selectional restrictions which can be used in subsequent analysis.

It is clear that the research in this thesis provides the framework for the eventual acquisition process. Many global issues remain to be solved. They will be the subject of future investigation.

References

[Alshawi, 1987] H. Alshawi, "Processing Dictionary Definitions With Phrasal Pattern Hierarchies", *Computational Linguistics*, 13, 1987.

[Alshawi, 1989] H. Alshawi, "Analysing the Dictionary Definitions", in Boguraev and Briscoe [1989a], pages 153–169.

[Amsler and White, 1979] R. A. Amsler and J. White, "Development of a Computational Methodology for deriving natural language semantic structures via analysis of machine-readable dictionaries", Technical Report MCSS77-01315, NSF, 1979.

[Baring-Gould, forthcoming] S. Baring-Gould, *SemNet: The Knowledge Representation of LOLITA*, PhD thesis, Durham University, forthcoming.

[Boguraev and Briscoe, 1989a] B. Boguraev and T. Briscoe, editors, *Computational Lexicography for Natural Language Processing*, Longman, 1989.

[Boguraev and Briscoe, 1989b] B. Boguraev and T. Briscoe, "Introduction", in *Computational Lexicography for Natural Language Processing* [1989a], pages 1–39.

[Boguraev *et al.*, 1995] B. Boguraev, R. Garigliano, and J. Tait, "Editorial", *Journal of Natural Language Engineering*, 1, 1995.

[Bokma and Garigliano, 1992] A. F. Bokma and R. Garigliano, "Uncertainty Management through Source Control: A Heuristic Approach", in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Mallorca, Spain, July 1992.

[Briscoe *et al.*, 1993] T. Briscoe, V. de Paiva, and A. Copesteak, editors, *Inheritance, Defaults, and the Lexicon*, Studies in Natural Language Processing, Cambridge University Press, 1993.

[Cater, 1995] A. Cater, "Lexical Knowledge Required for Natural Language Processing", in C. Guo, editor, *Machine Tractable Dictionaries: Design and Construction*, pages 31–54, Ablex Publishing Corp., 1995.

[Copestake *et al.*, 1993] A. Copestake, A. Sanefillipo, T. Briscoe, and V. De Paiva, "The ACQUILEX LKB: An Introduction", in Briscoe et al. [1993].

[Copestake, 1990] A. Copestake, "An approach to building the hierarchical element of a lexical knowledge base from a machine readable dictionary", in *Proceedings of the First International Workshop on Inheritance in NLP*, pages 19–29, June 1990, ESPRIT BRA-3030 ACQUILEX WP No. 008.

[Copestake, 1993a] A. Copestake, "Defaults in Lexical Representation", in Briscoe et al. [1993].

[Copestake, 1993b] A. Copestake, *The Representation of Lexical Semantic Information*, PhD thesis, University of Sussex, 1993.

[Copesteak, 1993] A. Copesteak, "The Compleat LKB", Technical Report 316, University of Cambridge Computer Laboratory, 1993, ACQUILEX-II Deliverable 3.1.

[Costantino *et al.*, 1996] M. Costantino, R. J. Collingham, and R. G. Morgan, "Natural Language Processing in Finance", *The Magazine of Artificial Intelligence in Finance*, 2(4), 1996.

[DAR, 1995] DARPA, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Morgan Kaufmann Publishers, November 1995.

[DAR, 1998] DARPA, *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Morgan Kaufmann Publishers, 1998.

[DeJong, 1979] G. DeJong, "Prediction and Substantiation: A New Approach to Natural Language Processing", *Cognitive Science*, 3:251–273, 1979.

[Downing, 1977] P. Downing, "On The Creation And Use Of English Compound Nouns", *Language*, 1977.

[Fellbaum, 1998a] C. Fellbaum, "Introduction", in Fellbaum [1998b], pages 1–19.

[Fellbaum, 1998b] C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[Fernandez, 1995] M. A. Fernandez, "Spanish Generation in the NLP System 'LOLITA"', Master's thesis, Durham University, Department of Computer Science, 1995.

[Fischer, 1997] D. H. Fischer, "Formal Redundancy and Consistency Checking Rules for the Lexical Database WordNet 1.5", in P. Vossen, N. Calzolari, G. Adriaens, A. Sanfilippo, and Y. Wilks, editors, *Proceedings of the ACL/EACL-97 workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 22–31, Madrid, July 1997.

[Franz, 1996] A. Franz, *Automatic Ambiguity Resolution in Natural Language Processing: An Empirical Approach*, LNAI 1171, Springer-Verlag, 1996.

[Garigliano *et al.*, 1993] R. Garigliano, R. Morgan, and M. H. Smith, "The LOLITA System as a Contents Scaning Tool", in *Proceedings of the 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language Processing*, Avignon, France, May 1993.

[Garigliano *et al.*, 1996] R. Garigliano, R. Morgan, and D. Nelson, "Rolls Royce Wizard of Oz Experiment", Technical Report 6, Department of Computer Science, University of Durham, 1996.

[Guha and Lenat, 1990] R. V. Guha and D. B. Lenat, "Cyc: A Midterm Report", *A.I. Magazine*, 1990.

[Guha *et al.*, 1990] R. V. Guha, D. B. Lenat, K. Pittamn, D. Pratt, and M. Sheppard, "Cyc: toward programs with commonsense", *Communications of the ACM*, 33(8), 1990.

[Guthrie *et al.*, 1996] L. Guthrie, J. Pustejovsky, Y. Wilks, and B. M. Slater, "The Role of Lexicons in Natural Language Processing", *Communications of the ACM*, 39(1), 1996, Special Issue on NLP.

[Hawkins and Nettleton, forthcoming] P. Hawkins and D. J. Nettleton, "Large Scale WSD Using Supervised Learning applied to SENSEVAL", *Journal of Computers and the Humanities. Special Issue on SENSEVAL*, forthcoming.

[Hirst, 1987] G. Hirst, *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press, 1987.

[Hobbs, 1986] J. R. Hobbs, "Resolving Pronoun References", in B. J. Grosz, K. S. Jones, and B. L. Webber, editors, *Readings in Natural Language Processing*, pages 339–352, Morgan Kaufmann, 1986.

[Jones, 1993] C. Jones, "Dialogue Analysis and Generation: A Theory for Modelling Natural English Dialogue", in *Proceedings of EUROSPEECH'93, the 3rd European Conference on Speech Communication and Technology*, September 1993.

[Jones, 1994] C. Jones, *Dialogue Structure Models: An Engineering Approach to the Analysis and Generation of Natural English Dialogues*, PhD thesis, Department of Computer Science, University of Durham, 1994.

[Kautz, 1987] H. A. Kautz, *A Formal Theory of Plan Recognition*, PhD thesis, University of Rochester, Department of Computer Science, May 1987, Also availiable as TR 215.

[Kilgarriff, 1998] A. Kilgarriff, "Why lexicographic quality matters", *LE Journal*, 1998.

[Landes *et al.*, 1998] S. Landes, C. Leacock, and R. I. Tengi, "Building Semantic Concordances", in Fellbaum [1998b], pages 199–216.

[Lenat *et al.*, 1986] D. Lenat, M. Prakash, and M. Shepard, "Cyc: Using Commonsense Knoweledge to Overcome Brittleness and Knowledge Acqusition Bottlenecks", *AI Magazine*, 6, No. 4:65–85, 1986.

[Lenat *et al.*, 1995] D. Lenat, G. Miller, and T. Yokoi, "CYC, WordNet and EDR: Critiques and Responses", *Communications of the ACM*, 38(11):45–48, 1995.

[Long and Garigliano, 1988] D. Long and R. Garigliano, "Inheritance Hierarchies", Technical Report 4/88, Department of Computer Science, University of Durham, 1988.

[Long and Garigliano, 1994] D. Long and R. Garigliano, *Reasoning by Analogy and Causality: A model and application*, Artificial Intelligence, Ellis Horwood, 1994.

[Miller *et al.*, 1993] G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker, "A Semantic Concordance", in *Proceedings of the ARPA Human Language Technology Workshop*, pages 303–308, Morgan Kaufmann, 1993.

[Miller, 1990] G. A. Miller, "Introduction to WordNet: An On-line Lexical Database", *International Journal of Lexicography*, 3:235–244, 1990.

[Miller, 1995] G. A. Miller, "WordNet: A Lexical Database for English", *Communications of the ACM*, 38(11), 1995.

[Miller, 1998] G. A. Miller, "Forward", in Fellbaum [1998b].

[Morgan *et al.*, 1994] R. G. Morgan, M. H. Smith, and S. Short, "Translation by Meaning and Style in LOLITA", in *Machine Translation: Ten years on*, Cranfield University and the British Computer Society, November 1994.

[Nakamura and Nagao, 1988] J. Nakamura and M. Nagao, "Extraction of Semantic Information from an Ordinary English Dictionary and its Evaluation", in *Proceedings of the 12th International Conference in Computational Linguistics*, pages 459–464, 1988.

[NCITS.T2 Committee on Information Interchange and Interpretation, 1998] NCITS.T2 Committee on Information Interchange and Interpretation, "Conceptual Graphs, draft proposed American National Standard (dpANS)", 1998.

[Nettleton and Garigliano, 1995] D. J. Nettleton and R. Garigliano, "Evolutionary Algorithms and Dialogue", in L. Chambers, editor, *Practical Handbook of Genetic Algorithms: New Frontiers*, volume 2, CRC Press, 1995.

[Parker, 1994] B. Parker, "Spelling Correction in the NLP system LOLITA: Dictionary Organisation and Search Algorithms", Master's thesis, Department of Computer Science, University of Durham, 1994.

[Poria and Garigliano, 1997] S. Poria and R. Garigliano, "Granularity for Explanation", in E. Costa and A. Cardoso, editors, *LNAI 1323: Progress in Artificial Intelligence EPIA '97*, Springer, 1997.

[Poria and Garigliano, 1998] S. Poria and R. Garigliano, "Factors in Causal Explanation", in C. Ortiz, editor, *Working Notes on Progress for a Commonsense Theory of Causation*, AAAI, 1998.

[Procter, 1995] P. Procter, editor, *Cambridge International Dictionary of English*, Cambridge University Press, 1995.

[Quirk *et al.*, 1985] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik, *A Comprehensive Grammar of the English Language*, Longman, 1985.

[Schank and Abelson, 1977] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding*, Erlbaum, 1977.

[Slator and Wilks, 1990] B. M. Slator and Y. A. Wilks, "Toward Semantic Structures from Dictionary Entries", in U. Schmitz, R. Schutz, and A. Kunz, editors, *Linguistic Approaches to Artificial Intelligence*, pages 419–460, Peter Lang, 1990.

[Slator *et al.*, 1990a] B. M. Slator, S. Amirsoleymani, S. Andersen, et al., "A Methodology for Empirically Deriving Semantic Classes", Technical Report NDSU-CS-TR-90-7-32, Department of Computer Science and Operations Research, North Dakota State University, 1990.

[Slator *et al.*, 1990b] B. M. Slator, S. Amirsoleymani, S. Andersen, et al., "Towards Empirically Derived Semantic Classes", in *Proceedings of the Fifth An-*

nual Rocky Mountain Conference on Artificial Intelligence, pages 257–262, Las Cruces, N.M., 1990.

[Smith, 1996] M. H. Smith, *Natural Language Generation in the LOLITA system: An Engineering Approach*, PhD thesis, Department of Computer Science, University of Durham, 1996.

[Sowa, 1984] J. F. Sowa, *Conceptual Structures - Information Processing for Mind and Machine*, Addison Wesley, 1984.

[Urbanowicz, 1999] A. Urbanowicz, *Interpretation of Anaphoric Expressions in the LOLITA system*, PhD thesis, Durham University, 1999.

[Vossen, 1990] P. Vossen, "The end of the chain: Where does decomposition of lexical knowledge lead us eventually?", in *Proceedings of the Fourth conference of Functional Grammar*, pages 11–39, June 1990, [Also ESPRIT BRA-3030 ACQUILEX WP NO.010].

[Vossen, 1997] P. Vossen, "EuroWordNet: a multilingual database for information retrieval", in *Proceedings of the DELOS workshop on Cross-language Information Retrieval*, March 1997.

[Vossen, 1998] P. Vossen, "EuroWordNet: Building a Multilingual Database with Wordnets for European Languages", in K. Choukri, D. Fry, and M. Nilsson, editors, *The ELRA Newsletter*, ELRA, 1998.

[Wang and Garigliano, 1992] Y. Wang and R. Garigliano, "Detection and Correction of Transfer by CAL", in *Second International Conference on Intelligent Tutoring Systems (ITS-92)*, Springer-Verlag, June 1992.

[Wang, 1994] Y. Wang, *An Intelligent Computer-based Tutoring Approach for the Management of Negative Transfer*, PhD thesis, Department of Computer Science, University of Durham, 1994.

[Whitten, 1994] D. Whitten, "Cyc: Frequently Asked Questions (FAQ)", Technical report, MCC and Cycorp, 1994.

[Wilks *et al.*, 1996] Y. Wilks, B. M. Slator, and L. M. Guthrie, *Electric Words*, MIT Press, 1996.

[Wilks, 1975a] Y. Wilks, "Preference semantics", in E. L. Keenan, editor, *Formal Semantics of Natural Language*, pages 329–348, Cambridge University Press, Cambridge, 1975.

[Wilks, 1975b] Y. Wilks, "A Preferential, Pattern-Seeking Semantics for Natural Language Inference", *Artificial Intelligence*, 6:53–74, 1975.

[Wilks, 1996] Y. Wilks, "Natural Language Processing – Introduction", *Communications of the ACM*, 39(1), 1996, Special Issue on NLP.

Appendix A

Glossary

A.1 Terms

Anaphor	a lexical item which refers to a previously introduced object in some text.
Antecedent	also called referent, co-referent is the object to which an anaphor refers to.
Cyc	a large common sense knowledge base project.
Differentia	the part of a dictionary definition following the genus.
Genus	the term within a dictionary definition which is more general than the headword.
Headword	the word in a dictionary entry which is being defined.
IS-A	a semantic relationship used to indicate set inclusion. between two concepts.
Utterance	a string of words produced by a speaker or writer on a given occasion and in some context.
Referent	the object to which an anaphor refers.
Word Sense	a particular view of a word.

A.2 Acronyms

ACQUILEX	Acquisition of Lexical Knowledge for Natural Language Processing
COBUILD	Collins COBUILD English Language Dictionary
CIDE	Cambridge International Dictionary of English
GUI	Graphical User Interface
KR	Knowledge Representation
KRL	Knowledge Representation Language
LDOCE	Longman Dictionary of Contemporary English
LOLITA	Large-Scale Object-Based Linguistic Interactor, Translator and Analyser
MRD	Machine Readable Dictionary
NL	Natural Language
NLE	Natural Language Engineering
NLU	Natural Language Understanding
NP	Noun Phrase
POS	Part of Speech
PP	Prepositional Phrase
WSA	Word Sense Ambiguity
WSD	Word Sense Disambiguation

Appendix B

CIDE Definitions Used for the Evaluation

B.1 Noun Definitions

abattoir — a place where animals are killed for their meat

adder — a poisonous snake

ball — a large formal occasion where people dance

cake — a sweet type of food made with a mixture of flour, eggs, fat and sugar

carp — to complain continually about unimportant matters

carrot — a long pointed orange root eaten as a vegetable

chaplain — a Christian official who is responsible for the religious needs of an organisation

crack — a pure and powerful form of the drug cocaine

dot — a very small round mark

drive — a planned effort to achieve something

drop — a very small amount of liquid

eagle — a very large strong bird with a curved beak which eats meat and can see very well

fold — a small area of a field surrounded by a fence where sheep can be put for shelter for the night

hangar — a large building in which aircraft are kept

hangman — a person whose job is to operate the device which kills criminals by hanging them from a rope by their necks

hangover — a feeling of illness after drinking too much alcohol

hide — the strong thick skin of an animal which is used for making leather

hideaway — a place where someone goes when they want to relax and get away from their usual surroundings

hostage — someone who is taken prisoner by an enemy in order to force the other people involved to do what the enemy wants

hunch — an idea which is based on feeling and for which there is no proof

hydrogen — the lightest gas with no colour, taste or smell, that combines with oxygen to form water

imprint — the name of a publisher as it appears on a particular set of books

industry — the people and activities involved in one type of business

ingredient — a food that is used with other foods in the preparation of a particular dish

initiative — the ability to use your judgement to make decisions and do things without needing to be told what to do

jack — a piece of equipment which can be opened slowly to allow heavy weights to be raised

line — a row of words that form part of a text

paint — a coloured liquid that is put on a surface such as a wall to decorate it

vertigo — a feeling of spinning round and being unable to balance caused by looking down from a height

B.2 Verb Definitions

administer — to cause someone to receive something

adore — to worship as sent from god

arrange — to put something in a particular order

avert — to prevent something bad from happening

bail — to remove water from a boat using a container

bake — to cook inside a cooker without using added liquid or fat

banish — to send someone away from their country and forbid them to come back

bend — to cause to curve

bite — to use your teeth to cut into something

blather — to talk in a foolish way

bob — to move something up and down quickly and gently

check — to leave something at a particular place in the care of someone, so that it can be sent by aircraft to another place

chew — to crush food into smaller, softer pieces with the teeth so that it is easier to swallow

coerce — to persuade someone forcefully to do something which they are unwilling to do

crush — to defeat someone completely

file — to walk as a long line of people, one behind another

putt — to hit a golf ball gently across an area of short and even grass towards or into a hole

untangle — to remove the knots from an untidy mass of string, wire etc. and separate the different threads

vomit — to empty the contents of the stomach through the mouth

walk — to move along by putting one foot in front of the other, allowing each foot to touch the ground before lifting the next

weld — to join two pieces of metal together permanently by melting the parts that touch

Appendix C

USER MANUAL

CIDE User Manual

Introduction

The proliferation of electronic information in recent years had led to a demand for computer systems which are able automatically process text in order to carry out a predefined task, *eg*, find all documents to do with a particular topic, extract certain key information from within a document, summarise a piece of text, *etc*. These tasks require an ability to 'understand' the meaning of the various sentences and phrases which make up a document. In order for a computer system to do this it requires various levels of knowledge:

grammatical word information

knowledge about the structure of a word.

semantic word information

knowledge about meaning of a word, *eg:*
- "*soccer*" is a game played on a pitch by two teams.
- "*sell*" involves a transfer of money from a buyer to a seller.

world knowledge

knowledge such as:
- what particular objects are used for,
- why particular events happen.

It is widely recognised that knowledge about words and their meanings (the first two types) is already available in conventional dictionaries. However, dictionaries are aimed at human readers and need to be processed to extract and represent the knowledge within them in a form that can be utilised by computer systems. This project represents an effort to process that dictionary knowledge for computer use.

Task Description

This document provides information that will be required to carry out the data input process from CIDE (Cambridge International Dictionary of English). Before starting to read about the input process it is important that the user is familiar with the layout of definitions contained in CIDE (click here).

The system will analyse as much of the definition as is possible. However, it will need help in resolving various problems and ambiguities that occur in the original definition. This help will be given by the user in the form of a question-answering session. The questions fall into a number of different categories which are listed below. The information in each of these sections present important information and tutorials to enable the successful completion of the analysis process.

- choosing grammatical categories
- picking word meanings
- entering word information
- solving structural ambiguities
- finding referents for mentioned words
- finding referents for implicit objects
- making objects more specific
- naming events and entities
- naming relationships
- confirming the analysis

The categories of questions in the list above also represents a rough estimate as to the order of the question - answering process (earlier questions such as picking word meanings may also occur in a different context later in the process).

In practice the interaction with the underlying natural language system is via a GUI. This greatly simplifies the interaction process. It is anticipated that these pages of documentation will be used in close conjunction with the GUI.

Walk Through Examples

Two commented examples illustrate the various stages of the question-answering process. A discussion as to why various answers were selected throughout the process is also provided. The two examples are:

- the noun, hide
- the verb, banish

CIDE Layout

CIDE is laid out so that the *main definition* is in bold font and indented out of the main text. Consider the following example (from page 29 of CIDE):

air *(obj)* [BROADCAST] /............../ *V Am* to broadcast (something) on radio or television **o** The interview with the President will air tomorrow morning. [I] **o** The game will be aired **(live) on** CBS at 7.00 tonight. [T]

air /................./ *n* [U] **o** If a person or a programme is **on/off** (the) air, they are/are not broadcasting on radio or television: *The radio station is on air from 6.00 a.m.* **o** *As soon as the war started, any broadcasts with a military theme were* **taken** *off* the air.

This defines the word '**air**'. Each main definition can be thought of as consisting of the following parts:

the word being defined	this is called the *headword* of the definition, in this case it is '**air**'.
a field of *grammatical information* about the headword	for the example above this is "*(obj)* [BROADCAST] /............../ *v Am*" and often includes extra information about the meaning of the headword.
the *definition* of the headword	*i.e.,* "to broadcast (something) on television or radio".
a number of bulleted *examples* showing the usage of the headword	these are separated by the **o**'s above.
a number of indented *subsidiary definitions*	these define words which share a similar form or meaning to the main headword being defined. Often they are different parts of speech of the headword (*i.e.*, the noun meaning of 'air').

Only the main definitions are of interest to us for this task.

[back to top]

Choosing Grammatical Categories

The first question that a user will be asked when analysing a definition is to choose the correct grammatical category for the word. The user must choose between:

1. Noun representing an event
2. Noun representing an entity
3. Verb

The distinction between a noun and a verb is given in the grammatical field of the dictionary (the label '*n*' identifies nouns and '*v*' verbs). In addition, most verb definitions begin with 'to ...', *eg,* the definition of 'putt' begins 'to hit ...'. If the word is a noun then a more difficult distinction needs to be made, *i.e.*, whether it is an event or an entity. The remainder of this section discusses means by which the user can determine whether a noun represents an event or an entity.

What is an 'event'?

Nouns in English can be categorised as 1) commonplace objects (called entities) examples of which include, *a table*, *a car*, or *a shoe*, and 2) events, such as, *a scandal*, *a hurricane*, or a *murder*. The difference between the two types of nouns is that, while entities simply exist, events represents processes that happen, *i.e.*, they have a time-span and may themselves involve other events. A *murder* takes place at a certain time, and may involve f*inding a weapon*, *waiting for the target* and so on.

Distinguishing between 'event' and 'entity'.

Given a noun (call it **nn**) the user is expected to determine if it corresponds to an event, or to an normal entity. In order to determine this the user should ask themselves whether it makes sense to say that "an **nn** occurred", or that, "an **nn** happened". If it does make sense to say this then the entity probably represents an event.

For example, consider the definition,

> **alarm** *n* (a loud noise, flashing light etc., that gives) a warning of danger

then this sense of "*alarm*" is clearly an event because it makes sense to say "an *alarm* happened", or, "a *loud noise* happened". However, if the user is unsure, they should choose the entity meaning because most nouns in English are of this type.

[back to the main page]

Picking Word Meanings

At various stages in the question-answering process the user will be asked to select between various meanings of a word in order to determine which meaning is relevant to the particular word or word definition that is being considered. Various pieces of information about the alternative meanings will be output to aid in the selection of the appropriate meaning.

For example, consider the noun 'bag' (an entity). Amongst the different meanings may be the following three possibilities:

```
1)
Bags. (=> Baggages
      = Suitcases , Grips
      <= Weekenders , Gripsacks , Overnighters , Carpetbags , Portmanteaus)
 family: inanimate manmade
 emotional value: indifferent
 level of language: common level

a portable rectangular traveling bag for carrying clothes

2)
Bags. (=> Containerfuls
      = Bagfuls)
 family: quantity
 emotional value: indifferent
 level of language: common level

3)
Bags. (=> Mammas
         = Dugs , Udders)
 family: inanimate organic
 emotional value: indifferent
 level of language: common level

mammary gland of e.g. cows and sheep
```

Associated with each meaning are up to three pieces of extra information which are of use to the user when attempting to select the appropriate meaning. This extra information consists of the following (in order of informativeness):

An English Description

For many of the word meanings an English description of the word is provided. This is perhaps the most informative piece of information and if available is shown at the end of a meaning's description. An example of the meaning's use is also included if available. In the above example, English descriptions are available for meanings 1 and 3, an example use is available for meaning 3, but meaning 2 does not have any of this information.

Associated Words

The second most useful piece of information for distinguishing meanings is given by a list of words that is usually associated with each of the particular meanings. Some of these other words may be familiar and will therefore help in identifying the intended meaning. In this case the intended sense of 'bag' in (2) is clear because it is associated with known words such as a '*containerful*' or '*bagful*'. An example of this meaning would be that of 'bag' in "he had a *bag* of sweets".

Other Information

The remaining information that is included in the description of a word's meanings is an assortment of information about particular aspects of the word. This is presented as a list of categories and their associated value. So from 2) above this information is the section:

family: quantity
emotional value: indifferent
level of language: common level

The most important of these from a meaning identification point of view is the *family* of the word which is a broad categorisation as to the type of the object, *e.g.*, *manmade*, *organic*, *food*, *human*, *inanimate, quantity, etc.*

The categories in the list (*e.g.*, family, type, emotional value, *etc.*) are known as control values and are discussed further in the section on entering word information.

[back to main menu]

Entering Word Information

After selecting the appropriate meaning for a word it is important to ensure that a number of values that are associated with that word are correct. There are three broad categories of information that the user is expected to provide:

1. usage information about the word
   - applicable to both nouns and verbs
2. grammatical information explicit in the dictionary
   - nouns
   - verbs
3. grammatical information implicit in the dictionary
   - nouns
   - verbs

This information will often be referred to as control information. Each of the types above are dealt with in turn below.

The actual entry/alteration of existing control values is facilitated by the GUI. This displays the appropriate controls that can be set for the word that has been selected at the "picking word meaning" stage. Typically various lists of control values are displayed. The user then selects that those values that are most appropriate. The labels that are displayed above each of the lists in the GUI are shown below in red.

Word Usage Information

This provides information about the context in which different words are often used. This information is required for all types of words (*i.e.*, both nouns and verbs). There are two categories that need to be considered:

Level of Language

Describes the general context in which words are often associated. The following table shows the possible values together with a description of the value's meaning and some examples.

Level of Language	Description	Examples
specialist	language of a specialised domain	medical terms, plant names
social jargon	slang language generally used by youngsters	bod, cool, hip, chunder
archaic	words from the past which are not really used much anymore	valour
figurative	words not used in the same sense as their commonly understood counterparts	'*drink*' in "my car *drinks* petrol"
taboo	words that are avoided in everyday speech	swear words, death
formal	words that are not used in normal everyday speech - tend to be the 'proper' name for common terms	automobile, offspring
informal	words which sound odd in polite conversation	kid, motor
common	words used in everyday language (the default selection)	anything that doesn't fall into the above.

Emotional Value

Describes the grade of emotion often associated with a word:

Emotional Value	Description	Examples
negative	words that generally invoke negative emotions	war, fight, cheat, bomb, starve
positive	words that generally invoke positive emotions	love, relax, sunbathe, ice-cream
context dependent	words which have a strong emotional value in the contexts in which they are of personal relevance	prison, flood, beat
controversial	words that nearly always have a strong emotional value associated with them	hang, hunt, religion, genocide
indifferent	words which carry no emotional value in themselves (the default selection)	household objects

Explicit Grammatical Knowledge

This refers to the entering of knowledge which is explicitly encoded in the dictionary entry, either before the main definition, or, within the examples. It simply requires the user to detect certain patterns in the text.

This section is divided into the knowledge required for nouns and verbs.

Nouns

uncountable

> nouns that cannot be counted, i.e., it makes no sense to ask, how many? They are indicated by the pattern "[U]" in the grammatical field of the entry, or, following any example

irregular plural

> nouns whose plurals are not formed by adding a "es/ies/s" ending, *eg*, the plural of knife (knives) is an irregular plural. They are indicated by the pattern "*pl* <pln>" in the grammatical field of the entry. The user will be expected to enter the string that corresponds to <pln>.

Verbs

There are three main classes of verb. These can be distinguished between by examining the relevant dictionary entry.

Relation Type

This will typically be automatically selected by the system (from the information attached to the word meaning). The user should correct this if it does not correspond to the dictionary definition.

Relation Type	Description	How to extract value from dictionary definition
transitive	a verb which always take an object	indicated by the pattern "*obj*" in the grammatical field of the entry
transitive implicit	a verb which can occur with or without an object	indicated by the pattern "*(obj)*" in the grammatical field of the entry
intransitive	a verb which does not take an object	if neither of the above situations occur

Irregulars

For many verbs their past and past participle forms can be determined by following a set of rules. There are however a number of cases where this is not the case. Such examples are said to have irregular forms. The dictionary definition contains the information on these irregulars and the user is expected to enter this. To enter the information the user should click on (for example) "irregular past", this then allows the irregular form to be typed into the box to the right.

irregular past

verbs whose past tenses are not formed by adding an "ed" ending, *eg*, get, keep. They are indicated by the pattern "*past* <pst>", or, "*past simple* <pst>" in the grammatical field of the entry. The user is expected to enter the word corresponding to <pst>.

irregular participle

verbs whose part participle is not formed by adding an "ed" ending, *eg*, seen, fallen. They are indicated by the pattern "*past* <prt>", or, "*past part* <prt>" in the grammatical field of the entry. The user is expected to enter the word corresponding to <prt>.

Information appropriate to Transitive Verbs only

Transitive	Description	Examples	How to extract value from dictionary definition
dative	verbs in which the destination of the action can be expressed as an object before the normal one	"I gave *you* the book", really means "I gave the book *to you"*	indicated by an occurrence of the label "[+ two objects]" following any example sentences

Information appropriate to Transitive and Transitive Implicit Verbs

Transitive and TransitiveImplicit	Description	Examples	How to extract value from dictionary definition
copula	verbs which link the properties of something to that thing	John became angry. He felt embarrassed.	indicated by an occurrence of the label "[L]" following any example sentences
infinitive	See below.		

Infinitive

Depending on the relation type that is selected a number of infinitive possibilities will be available. The GUI displays only those that are appropriate for the relation type selected. In the case of Intransitive the elements of the list of possibilities can be selected immediately. For the remaining two types if the user wishes to select a value in the Infinitive list then it must be activated by selecting infinitive in the "Transitive" or "Transitive Implicit" list.

NB multiple infinitive values can be selected.

Infinitive	Description	Examples	How to extract value from dictionary definition
infinitive	A verb which can have a "to" phrase following it (*i.e.*, "to" followed by a verb).	He likes *to dance*, She wants *to dance*	indicated by the pattern "[+ *to* infinitive]" in the grammatical field of the entry or, following any example sentences
inf+obj	A verb which can have a "to" phrase following the object of the verb.	She wants *him to dance*	indicated by the pattern "[+ obj + *to* infinitive]" in the grammatical field of the entry, or, following any example sentences
inf-to	A verb similar to the the infinitive case above, but does not need to occur with the "to".	I heard *him shout*	indicated by the pattern "[+ infinitive without *to*]" in the grammatical field of the entry, or, following any example sentences
inf+obj-to	A verb similar to the inf+obj case above, but do not need to occur with the "to".	They acted quickly *to repair* the damage	indicated by the pattern "[+ obj + infinitive without *to*]" in the grammatical field of the entry, or, following any example sentences

Implicit Grammatical Knowledge

A certain amount of grammatical knowledge which is to be extracted is not encoded explicitly within the grammatical field of the major definition of a word. The descriptions below provide a list of these items together with an indication of how the knowledge should be extracted. The section is divided into the knowledge required for nouns and for verbs.

Nouns

gender

The biological gender in nouns.

gender	Description/Example
male	*eg*, man, boy, *etc.*
female	*eg*, woman, girl, *etc.*
sexed	*eg*, arbitrary individual persons or animals
neutral	*eg*, inanimate/abstract objects and notions
unknown	objects which are hard to sex, *eg*, certain plants, organisms

Verbs

Information appropriate to Transitive Verbs only

Transitive	Description	Examples
communication	These are verbs whose objects are linguistic in nature and not observable events in themselves.	"John told me *he crashed the car*", "He wrote that *he was sad*".
fully reflexive	A class of verbs which always have the same subject and object.	"she prides *herself*" "he perjured *himself*"

Information appropriate to Transitive Implicit Verbs only

Transitive Implicit	Description	Examples
symmetric	Verbs which have the same meaning, if their subject and object are reversed.	"United played City" is the same as "City played United".
semi reflexive	In the cases where the object is implicit, then it can be assumed that the verb applies to the subject.	"I shave" means "I shave *myself*", "He dressed" means "He dressed *himself*"

Information appropriate to both Transitive and Transitive Implicit Verbs

Transitive and Transitive Implicit	Description
personal	Denoting private states which can only be subjectively verified, *i.e.*, they hold only in the mind of the speaker. The various kinds of states include: ● intellectual (*eg*, know, believe, think) ● emotion/attitude (*eg*, intend, wish, want, like) ● perception (*eg*, hear, feel, smell, taste) ● bodily perception (*eg*, hurt, ache, tickle) Note the user does not have to select the kind of state.
sentential	See below.

Sentential

Applicable only to "Transitive" or "Transitive Implicit" verbs. To activate this list of values the sentential value in the "Transitive" or "Transitive Implicit" list should be selected.

sentential	Description	Examples
open	prefers the verb form of the event instead of noun form	"I know *John was murdered*" instead of "I know *John's murder*"
closed	prefers noun form of the event instead of the verb form	"I understand *John's murder*"
indifferent	no preference	"I saw *John's murder*" and "I saw that *John was murdered*"

[back to main page]

Guidelines for Resolving Structural Ambiguities

Introduction

When analysing sentences, the system must make decisions about the structure of the sentences involved. For example, given the sentence "*John hit the man with a stick*", the system must resolve the ambiguity of whether *John used a stick to hit the man*, or if, *the man he hit had a stick*. Such decisions can require considerable knowledge about the world and the context of the sentence, and are consequently difficult for a computer to make. To be sure of producing accurate results, the system will often require assistance in cases of such ambiguity. This assistance will be in the form of a question answering process. Each question will present a number of alternative readings of a sentence and you will be asked to reject those which violate your intuitive understanding of the text. The system uses brackets to indicate the different readings, with each bracket showing the parts of the sentence that form a coherent unit. For the example above you may be presented with the following:

1. *John (hit the man) with a stick*
2. *John hit (the man with a stick)*

You should answer these questions by checking that the units formed with brackets correspond to your understanding of what is being said. For example, if you understand that *John used a stick to hit the man* then the unit *(the man with a stick)* does not match with your understanding (because in your understanding the man who was hit does not have a stick). In this case you should reject that particular form of brackets, *i.e.*, reject interpretation 2.

In many cases, you will find several answers which seem to be incorrect. In that case you should reject them all. Supposing we are analysing the sentence, "*John hit the man with a stick on his head*", and we understand that *John used a stick to hit the man on his head.* We might be offered the following choice of structures:

1. *John (hit the man) with a stick on his head*
2. *John hit (the man with a stick on his head)*
3. *John (hit the man with a stick) on his head*

Given our understanding of what is happening, we can reject 2 immediately because this builds a unit *(the man with a stick on his head)* and our interpretation does not involve such a man. However, the remaining units *(hit the man)* and *(hit the man with a stick)* are perfectly consistent with our interpretation (despite the fact that the second also contains the same structural ambiguity that occurs in our first example). The system will then look for any other structural distinctions between the remaining sentences (1 and 3), producing a further questions if any can be found:

1. *John hit (the man) with a stick on his head*
2. *John hit (the man with a stick) on his head*

Given the understanding discussed at the start of the example, we can reject option 2 because this understanding does not involve a *man with a stick.* We are now left with only one option which has the structure which corresponds to our desired understanding.

It is also interesting to consider what would happen if we had understood that *the man had a stick* (rather than *John used the stick to hit him*). Given this interpretation of the simple sentence "*John hit the man with a stick*", we would be asked to select from the following structures:

1. *John (hit the man) with a stick*
2. *John hit (the man with a stick)*

With the new interpretation, we can quickly see that the structure of 1 can be rejected. This is because phrases such as *'with a stick'* can only modify the main item in a bracketed unit, not a subordinate item. In this case, the main item in (*hit the man*) is the *hitting* of the man. *The man* is a subordinate item (it is involved in the *hitting*). It is therefore possible for the phrase '*with a stick'* to modify the *hitting* but it cannot modify *the man.*

As noted above, phrases such as '*with a stick'*, '*on his head'*, '*at the house'*, '*of the people'*, *'to the shops'* which occur directly after a bracketed unit, can only modify the main item in a bracketed unit. In addition, only certain types of main items can be modified by such phrases; namely nouns (eg, '*the man'*) and verbs (eg, the '*hit'*). For example, the phrase following the bracketed units below, cannot modify any item in the bracket:

1. *John hit the man (with a stick) on his head*
2. *His ambition was (to play well) at the tournament*

because '*with'* and '*to'* (the main items in the brackets) are neither nouns or verbs. In these cases the phrases following the bracketed segments are assumed to modify some earlier unspecified part of the sentence.

Procedure for Resolving Ambiguities

After deciding on the correct reading of the sentence you should ask yourself the following questions when making a choice on which interpretation(s) to reject:

1. Does the entity in the brackets form a unit which does not exist in your understanding of the sentence? If so then reject the interpretation.
2. Does the phrase following a bracketed segment modify an entity which is a subordinate item in the bracketed segment? If so then reject this interpretation.
3. If the phrase following the bracketed segment can modify the main item in the bracketed unit, then is the resulting modification consistent with your understanding of the sentence? If not then reject the interpretation.

Now please work through the examples given below. In the cases where you answer wrongly please ensure you understand the reason for this before returning to the question.

1. banish
2. hangman
3. jack

If you feel that you need to know more about the modification of the terms which occur in bracketed segments, you can read more here.

[back to the main page]

Prepositions and Prepositional Attachment

Many relationships in text are implicitly represented by the use of prepositions. Prepositions relate two entities: the prepositional subject and the prepositional complement. Examples of a **preposition** and its *complement* (collectively known as a prepositional phrase) are:

in	*the house*
on	*the table*
by	*signing a treaty*
for	*a hundred pounds*

the complement is characteristically a noun phrase or -ing clause and always occurs directly after the preposition itself. However, the prepositional subject (which can only be a noun or verb) need not directly precede the preposition. Consider the examples

1. *John punched Mary in the stomach*
2. *He laughed at the joke*
3. *The money appeared quickly on the table*
4. *They ate the fish with a fork*
5. *He saw the man with the hat looking suspicious*

which have the prepositional subjects and phrases underlined. The prepositional phrase is said to modify or be attached to the prepositional subject, hence the phrase '*in the stomach*' modifies the *punching* (i.e., *punched in the stomach* and not *Mary in the Stomach*) and '*with a fork*' is attached to the 'eat' (i.e., *eat with a fork* and not *fish with the fork*)

(1), (3) and (4) show that the prepositional subject need not directly precede the prepositional phrase. Examples (4) and (5) show that a preposition can be attached to different grammatical objects depending upon the context; the verb '*eat*' and the noun '*the man*' respectively.

The result of this means that coherent text is often written so that the prepositional subject is easy for the human reader to identify. Confusion can arise when this subject is ambiguous as in the sentence:

6. *I saw the man looking through a telescope*

which has the two interpretations

6a. *I saw the man   looking through a telescope*
  b. *I saw   the man looking through a telescope*

(6a) corresponds to the interpretation in which *I saw a man while I was looking through a telescope*, whereas the attachment in (6b) specifies that *I saw a man and he was looking through the telescope.*

A list of the English prepositions which occur in CIDE dictionary definitions are: against, at, before, between, beyond, by, down, for, from, in, into, like, of, off, on, over, through, to, under, with.

Finding Referents for Mentioned Words

These questions involve finding objects for words such as '*he*', '*she*', '*it*', 'their', '*they*', *etc.* which occur in the original definition. The particular object to which they refer is called its **referent**. An example of this type of question occurs in the processing of the definition:

> **abattoir** n [C] a place where animals are killed for their meat

During the processing, one of the questions that will be asked is:

```
Choose referent for #their# in
"each abattoir is a place where animals are killed for #their# meat":

0 Places, abattoirs, in which something kills animals
1 Animals
```

Notice the form of the question. The word in the original sentence which requires its object to be chosen is enclosed in #'s, with the relevant fragment of the definition shown underneath. In this case the whole definition is shown. The correct answer to this question is 1, as it is the animal's meat and not the abattoir's meat that is being referred to.

In some cases it is not necessary for the whole of the definition to be shown. For example in processing the definition:

> **hideaway** *n* [C] *infml* a place where someone goes when they want to relax and get away from their usual surroundings

the following question will be asked:

```
Choose referent for #their# in
"#their# usual surroundings":

0 Something that wants to relax and that somethings
1 Places, hideaways, that something go when they wants to relax
```

This shows only a fragment of the original definition. In this case the user may need to examine the full definition in order to make a decision as to the correct referent. Although the descriptions of entities in the list of possible referents may be a little convoluted (*eg*, case 0 above) the appropriate referent should be chosen if it can be identified. In the above example meaning 0 refers to the subject of the verb 'want' (although the description is convoluted) and so this is the correct answer. Selection of alternative 1 would have meant that it was the hideaways that wanted to get away from their usual surroundings. If the correct referent cannot be identified, then the analysis of the definition should be abandoned.

[back to main page]

Finding Referents for Implicit Objects

As well as finding referents for words which are explicit in the text the user will also be asked to find referents for implicit objects in the definition. An example of a definition with an implicit object is:

> **hangman** *n* [C] *pl*-**men** a person whose job is to operate the device which kills criminal by hanging them from a rope by their necks.

In this example there are a number of objects that are important to the full understanding of the definition that are not explicitly mentioned in the definition, these are the implicit objects. For this example the implicit objects are:

1. an "*entity which is operating the device*",
2. an "*entity which kills criminals*",
3. an "*entity which hangs them*".

For each of these the user will be asked a question to find the appropriate referent. For example the question for the first of these would be:

```
Choose referent for "Something that operates a device":

0 Persons, hangmans
1 Persons's job that is something operates a device. A device kills criminals
2 some other entity
```

The user should simply select the appropriate entity, if it exists in the list. In this case the correct answer is the hangman and so option 0 should be selected. However, in some situations the most appropriate entity is not explicitly given in the list of possibilities. The following example contains such a situation:

> **abattoir** *n* [C] a place where animals are killed for their meat

the entity that is doing the killing (*i.e.*, the subject of the killing) is an implicit one because it is not mentioned. The user will therefore be asked the question:

```
Choose referent for "Something that kills animals in places, abattoirs":

0 Places, abattoirs, in which something kills animals
1 Animals
2 Animals's meat
3 some other entity
```

If the entity which is being asked about is not present in the list of referents, then the user should select the last option "*some other entity*". In the above example it is clear that the killing of the animals is not being carried out by the answers 0, 1 or 2 and so option 3 should be selected. Further questions will be asked at a later stage of the definition's analysis in order to determine more information about the 'some other entity'.

[back to main page]

Making Objects Specific

The notion of an implicit object was described in the section on implicit objects. The referents for some implicit objects are not present in the original definition. An example of this occurs in the following definition:

> **carrot** *n* [C] a long pointed orange root eaten as a vegetable

In this case the object that actually eats the carrot is not explicitly mentioned, *i.e.*, it is implicit. In addition the referent is "some other entity" which is not mentioned in the definition (in this case the computer is able to decide that the referent is "some other entity" without the need to ask the user a question).

In cases such as this the user will be asked additional questions to further specify the implicit object:

Choose meaning for "Things that eat something":

0 human	any human or group of humans
1 organisation	human organisations
2 human_or_organisation	any human or group of humans, including agents who could be either humans or human organisations
3 animal	all kinds of animals, except humans
4 animate	all animates, including humans and non human creatures
5 inanimate	all inanimate entities, both organic and inorganic
6 entity	generic label for all objects
7 event	generic label for situations, states, phenomenons etc.
8 other	enter more specific entity

When presented with such a question the user should select the most specific entity from the list of referents given.

In some cases the entity which is being sought has a specific name (this is discussed in detail in the following section on naming entities). However, if there is no one-word which describes the referent, and the correct answer falls in-between two of the referents above, the user should select the most specific referent which includes the correct group of objects. In the example above it is probably only humans and animals that eat carrots, but there is no such referent. However the closest referent which includes humans and animals is animate things (which may also include other things like organisms, etc.), and so this should be selected. The case in which there is a word which describes the particular set of things (where option 8 is chosen) is described in the following section on naming entities.

[back to main page]

Naming Entities

An important part of the user's interaction with the system is providing knowledge on the implicit objects that are contained within a definition. Previous sections have described the notion of an implicit object and a mechanism that allows them to be broadly classified (*e.g.*, is the implicit object an organisation or a human). Although a broad classification is useful it is often possible to give a very specific name to the implicit entity that is under consideration. This section describes the mechanism that the system employs to allow the user to give specific names to an implicit entity.

Let us consider the analysis of the definition of the transitive verb 'murder':

> **murder** to commit the crime of intentionally killing a person

amongst other questions the user will be asked to choose the type of the implicit entities corresponding to:

1. the thing that murders things
2. the thing that is murdered

from a list of possibilities that include:

- humans
- animals
- animates
- humans or organisations
- organisations
- entities
- events
- other more specific entity

At first glance the answer to (1) is that '*humans*' commit murder. Although this answer is correct it is a quite general answer, *i.e.*, not all humans commit murder. Often there exist a more specific set of items characterised by an English word. In this case the correct answer is that '*murderers*' commit murder and so the final option should be chosen and the appropriate name should then be entered (NB: the *root* singular form of the noun should actually be entered, *i.e.*, '*murderer*'.)

After entering the name and selecting the appropriate meaning (if more than one exists, the mechanism for picking word meanings will be used), you will be asked whether:

> [a]ll murderers or [s]ome murderers that murder?

In this case it is '*all murderers*' that commit the act of '*murder*'. This is the case because the verb *defines* the *set of murderers* in the sense that there cannot exist a particular *murderer* who does not *murder*.

A similar process is followed for the implicit entity (2) - the thing that is murdered.

Again the entity has a more specific name than one of those offered in the list above. The '*thing*' that is murdered is called a '*victim*'. However in this case it is not '*all victims*' that are murdered, *i.e.*, a person can be a victim without being murdered (*e.g.*, a *victim* of robbery or illness). In this case the verb '*murder*' does not define the set of '*victims*'. The user should select 'some' as the correct answer.

The dictionary entry of the main definition (in the case of verbs) often contains the names of defined entities because the meaning of these names are directly connected with the word being defined, *e.g.*, "*sellers sell* goods", "*cooks cook* food", "*drummers drum*", "people *possess possessions*", *etc.*

Naming Events

When a verb is being defined one of the questions that will be asked is to name the event being defined. In the example above the question would be stated as:

> Is there a name for the process: "*murderers murder some victims*"

All verbs define events, some of which will have names. These names are nouns which are generally given in CIDE as an indented minor definition to the main verb being defined. In this case the noun is also called '*murder*'. In general the name of the defined event will be some minor variation of the verb, *e.g.*, the verb '*to shop*' defines an event '*shopping*', and similarly '*to entrap*' defines an event '*entrapment*'.

[back to main menu]

Naming Relationships

Definitions contain many relationships between entities which are contained within it. Often these relationships are implicit. This section describes the types of questions the user will be asked to try and make those relationships explicit. The various types of implicit relationships are illustrated in the definition:

abattoir *n* [U] a place where animals are killed for their meat

The relationships are of two types:

1. those specified by words like "*to*", "*from*", "*by*", "*for*", "*of*", "*at*", *etc.* In the example above there is a relationship specified by the word "*for*" between "*killing*" and the "*animal's meat*".
2. those that exist between a compound noun (*i.e.*, two adjacent nouns such as "*car door*"). In the example above there exists a relationship between "*animals*" and "*meat*".

The user will be expected to use their world knowledge to make these relationships explicit.

Relationships specified by words

The following example question requires the user to decide on the relationship between two entities where the relationship is specified by the word 'for':

Choose the meaning of the word "for" between, "killing" and, "meat" :

0 showing the length of time	eg, I'm going to sleep for an hour
1 showing amount of distance	eg, He drove for 10 miles
2 towards; in the direction of	eg, They followed signs for the town centre
3 intended to be given to	eg, Roberto bought a toy for the baby
4 for the purpose of	eg, The neighbours invited us for dinner
5 in order to obtain	eg, He sent off for the details
6 in order to go into and travel in	eg, John ran for the bus; Roberto applied for a job
7 in order to achieve	eg, Kevin was trying for a first in his exams
8 because of; as a result of	eg, Bob was better for his week's holiday
9 compared to other similar things	eg, Jane is very mature for her age
10 getting in exchange	eg, Roberto paid $100 for the glasses
11 representing (a company, country, etc.)	eg, John works for a charity. Fred swims for England.
12 in trouble	eg, Gavin was in for it after that display
13 NONE	(leave structure and go on to next)

There are clearly a large number of different interpretations of the word "for". In order that the user can more easily distinguish between them, an example of the use of each is included. These examples are extremely useful in helping clarify the intended meaning of a particular relationship and should be studied carefully.

For this example, the correct relationship between *the killing* and *the meat* is (5), *i.e.*, *killing for the purpose of obtaining meat.* At first glance, answer (4) also seems plausible, *i.e., killing for the purpose of meat.* However, this option applies to events (*eg*, dinner) and not to entities (*i.e.*, the meat) as shown in the example. One way of deciding on competing relations is to try and think of a sentence in which the relevant entities can be inserted, eg, "*He kills animals for the purpose of their meat*", sounds less natural then saying, "*He kills animals in order to obtain their meat*".

If the user is unsure about which particular relationship to pick, or, cannot find one in the list that matches what they think it should be, then simply selecting the last option is the safest course of action.

Relationships between nouns

The second category of questions on relationships is for those between compound nouns. An example of this is:

Choose the meaning for the relation between "animal" and "meat"

0 uses	eg, Rick's machine
1 possess	eg, the boy's books
2 owns	eg, John's house
3 has a part	eg, Sarah's teeth
4 lives in	eg, my aunt's place
5 works in	eg, the butcher's shop
6 OTHER	more specific relation
7 NONE	leave structure and go on to next

Again the user should select the appropriate option if one exists. In many cases the exact relationship is not specified. In these cases the user should select one that approximately characterises the relationship if one exists. For example the relationship between "*animal*" and "*meat*" above is not as simple as one might think: "*meat is the flesh of an animal which is often eaten when the animal is dead*" (an intuitive definition of meat). However, ignoring the temporal details the relationship can be approximated to "*has a part*".

[back to main page]

Confirming the Analysis

At the end of the question/answering process the system will have completed the analysis of the definition. In order to determine whether the computer's interpretation of the analysis is correct the user will be asked for a final confirmation of whether to accept or reject the system's understanding. For example, following the analysis of the definition:

> **abattoir** *n* [C] a place where animals are killed for their meat

the final confirmation question may be similar to:

> **Accept definition as:**
> "Abattoirs are buildings in which some humans kill animals that are made of meat for meat."

As the computer system is using a natural language generation program to generate the system's understanding of the definition some of the sentences that are generated can be quite convoluted. However, a definition should only be rejected if the description is incoherent, or, inconsistent with the user's understanding of the definition. Often the description like the one above will omit certain details (*i.e.*, that the meat in "for meat" is the animal's meat). However this is no reason to reject it because the description is still consistent with the definition (although incomplete). The above interpretation should therefore be accepted.

Another example is with the definition:

> **bob** v to move something up and down, quickly and gently

the final confirmation question may be similar to:

> **Accept definition as:**
> "Some humans bobbing animate things are something moving up down quickly gently them"

which is not inconsistent, but is incomplete. The agent doing the "*moving*" is described as "*something*" when it could possibly be more specific. Again such an understanding should be accepted.

[back to main page]

Appendix D

GUI MANUAL

CIDE Analyser User Manual

This document provides an overview of how to use the interface to the CIDE Analyser. Details of the question answering process are described elsewhere.

Before using the analyser the user must set an environment variable. The following should be typed at the command prompt:

```
setenv LOLA_HOST default
```

The CIDE Analyser is then invoked by typing:

```
cide
```

at the command line.

The screen shots below show the GUI through which the user will interact with the underlying natural language processing system (LOLITA). Each of the interface windows shown can be viewed at full size by clicking on it.

The main types of interaction component are:

Start Screen
List Selection (single select)
Control Editor
List Selection (none, one or more than one)
Entering Text to make an Entity or a Relationship Explicit

These are described in detail below along with the following:

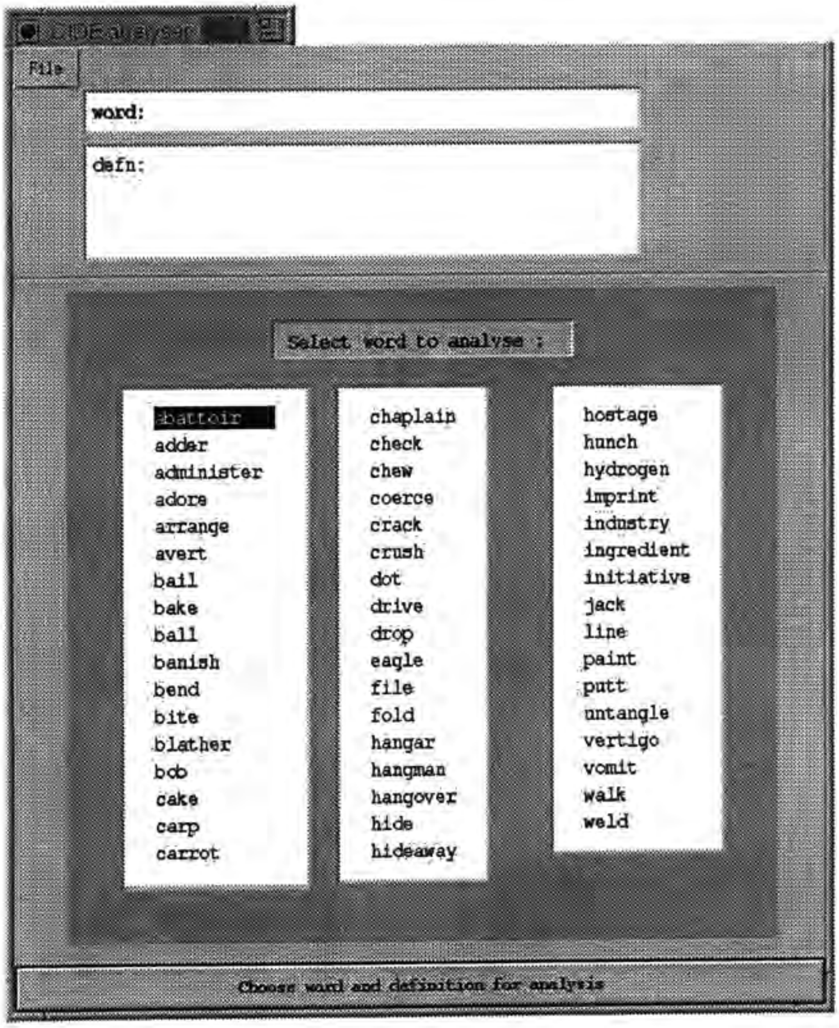
Menu Bar
Status Bar
Accepting/Rejecting Analysis
Abandoning Analysis
Analysis Summary

Start Screen

The screen shot to the left of the two below is the start-up screen. This is composed of a number of components:

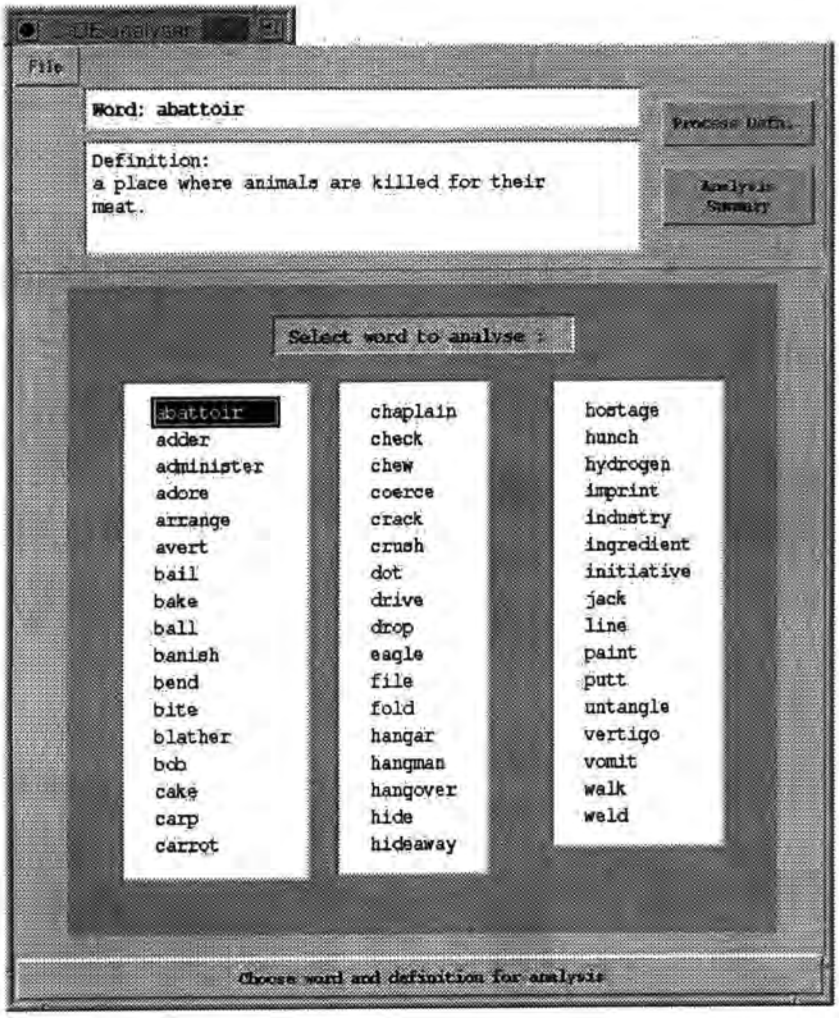
Menu bar	A pull down menu bar.
Text window containing "word:"	Displays the word which has currently been selected.
Text window containing "defn:"	Displays the definition that corresponds to the word that is selected.
Three lists	Lists containing the 50 words that are part of this exercise.

To examine a word's definition the user should use a single click of the left mouse button on the word. So clicking on the word 'abattoir' brings up the corresponding CIDE definition in the "defn:" text window. This window is shown on the right of the two below.



abattoir
adder
administer
adore
arrange
avert
bail
bake
ball
banish
bend
bite
blather
bob
cake
carp
carrot
chaplain
check
chew
coerce
crack
crush
dot
drive
drop
eagle
file
fold
hangar
hangman
hangover
hide
hideaway
hostage
hunch
hydrogen
imprint
industry
ingredient
initiative
jack
line
paint
putt
untangle
vertigo
vomit
walk
weld

Choose word and definition for analysis



Word: abattoir

Definition:
a place where animals are killed for their meat.

abattoir
adder
administer
adore
arrange
avert
bail
bake
ball
banish
bend
bite
blather
bob
cake
carp
carrot

chaplain
check
chew
coerce
crack
crush
dot
drive
drop
eagle
file
fold
hangar
hangman
hangover
hide
hideaway

hostage
hunch
hydrogen
imprint
industry
ingredient
initiative
jack
line
paint
putt
untangle
vertigo
vomit
walk
weld

Choose word and definition for analysis

Selecting a word also brings up two buttons:

Process Defn..	Left clicking on this starts the analysis of the definition.
Analysis Summary	Left clicking on this pops up a window which contains a summary log of any previous analysis for this definition. see

To start the analysis left click the "Process Defn.." button. An analysis can also be started by double clicking on the word of interest.

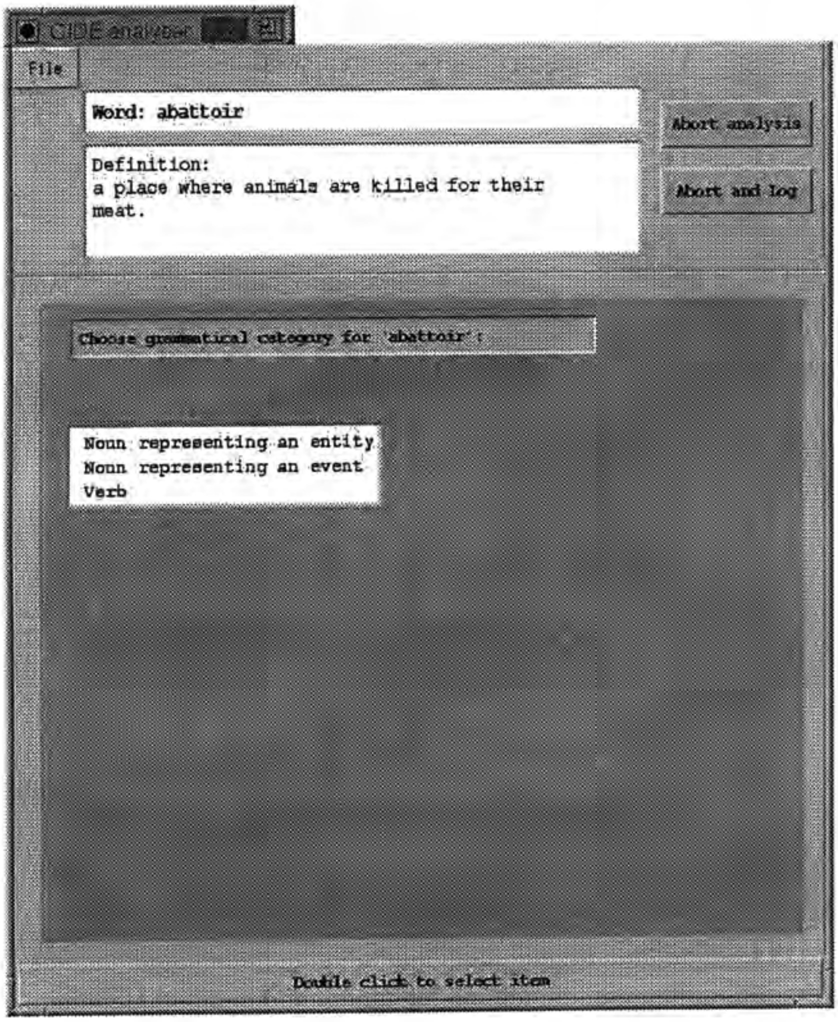
Status Bar

Throughout the interaction process there is a text status bar at the bottom of the interface which provides the user with useful information.

List Selection (single select)

The most common type of interaction that the user will have with the system is the selection of the most appropriate item from a list.

Example:
For each of the definitions the first question that the analysis asks is for the grammatical category for the word. This is an example of a list selection interaction. The following screen-shot shows the window that appears for abattoir:



Word: abattoir

Definition:
a place where animals are killed for their meat.

Choose grammatical category for 'abattoir':

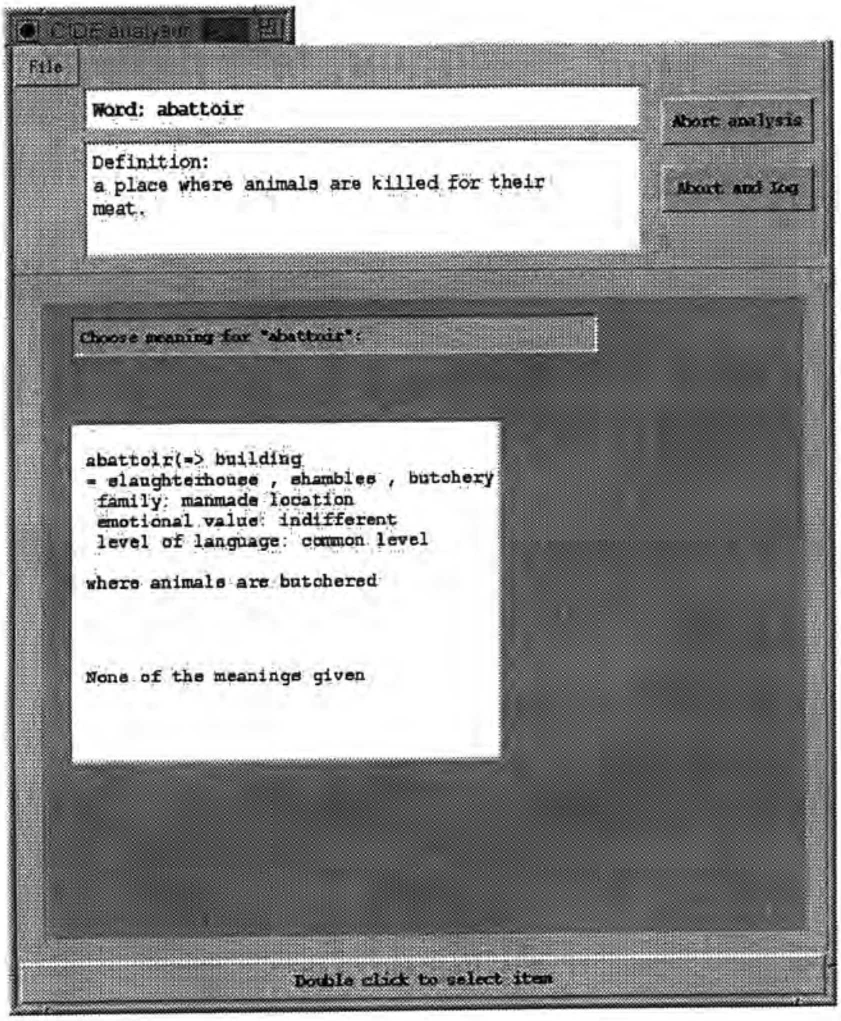
Noun representing an entity
Noun representing an event
Verb

Double click to select item

Note the two light blue buttons have now been replaced with some buttons which allow the user to abandon the analysis. These are described in their own section here.

A question is shown in the grey text box and below that are a list of possible answers that the system has generated.

The user can single click with the left mouse button to select any item in the list of possibilities. A single click just highlights the selection and the user can use single clicks to alter their selection. At any point a double click can be used to send the value for analysis. Double clicking on "Noun representing an entity" would be the correct selection in this case. This would then bring up the window on the left of the two shown below:

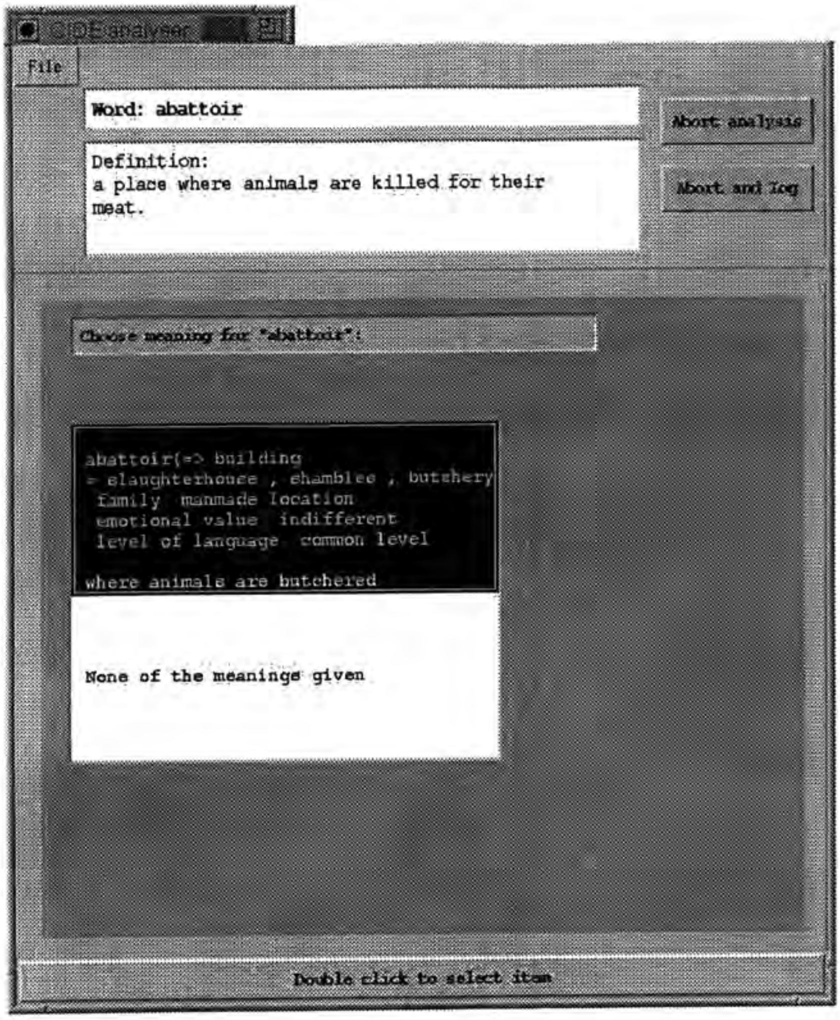


Word: abattoir

Definition:
a place where animals are killed for their meat.

Choose meaning for "abattoir":

```
abattoir(=> building
= slaughterhouse , shambles , butchery
 family: manmade location
 emotional value: indifferent
 level of language: common level

where animals are butchered



None of the meanings given
```



Word: abattoir

Definition:
a place where animals are killed for their
meat.

Choose meaning for "abattoir":

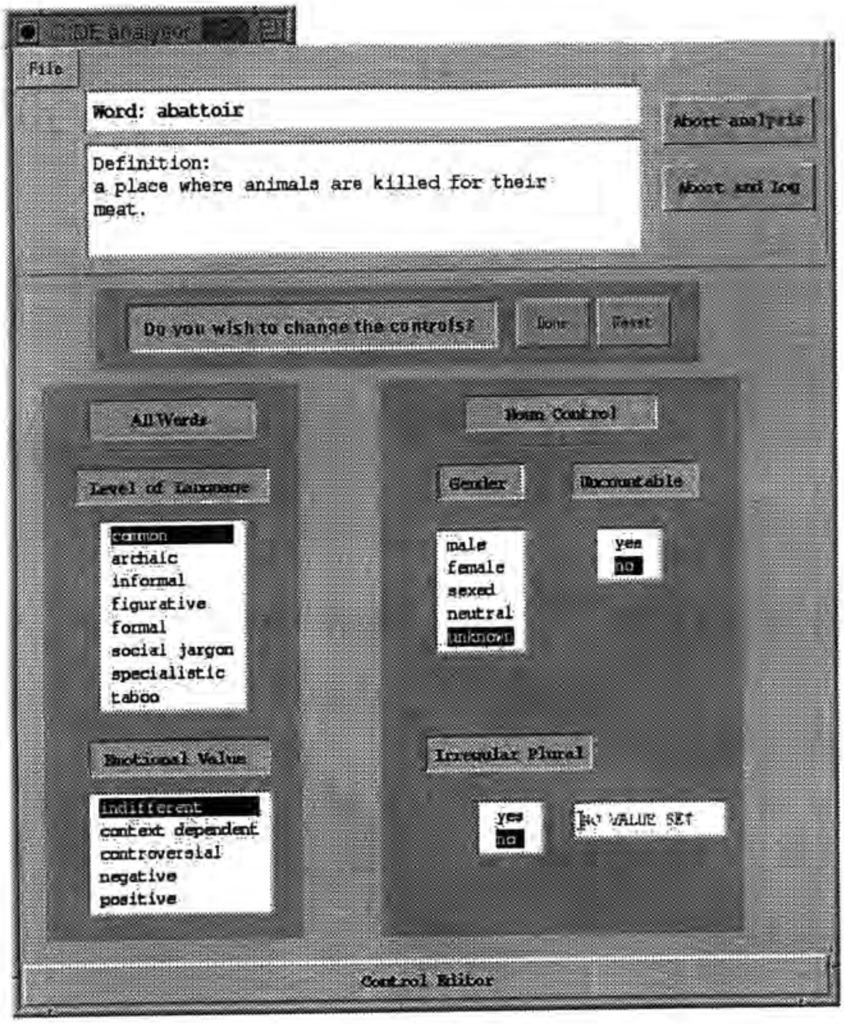
```
abattoir(=> building
= slaughterhouse , shambles , butchery
 family  manmade location
 emotional value  indifferent
 level of language  common level

where animals are butchered
```

None of the meanings given

Again a list selection interaction is shown. A single left mouse click on the top item in the list would produce the screen on the right of the two shown above. Double clicking this item leads the user to the control selection interface.

Control Editor

When the user has selected a word meaning there are some values associated with it. These are called control values and are automatically picked up by the interface. In most cases the values shown will be correct (although there may be some errors), but often some other values can be added from the information available in the dictionary. This editor allows the user to add this information.



Word: abattoir

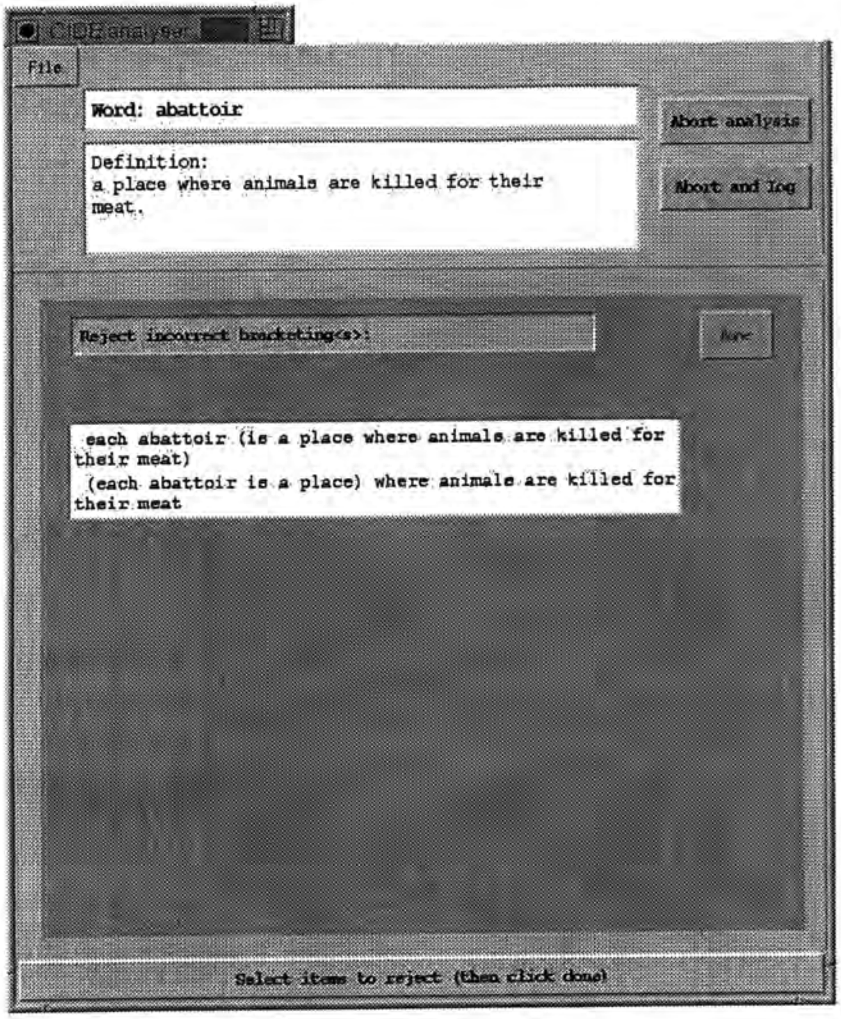
Definition:
a place where animals are killed for their meat.

The user should use the interface to select the appropriate control values. Once they are happy with their selection they should left click the "Done" button to continue (this sends the values to the underlying analyser). Should a user think they have made an error at this stage and wish to return to the values that were set on entry to the Control Editor they should left click on "Reset".

List Selection (none, one or more than one)

A section above described how a user may select and send to the analyser a single answer to a question. This is appropriate when the analyser ***expects*** a single answer to a question. There are, however, some situations where none, one or more than one answer may be appropriate. In such cases a different form of interaction component is required.

The screen shot on the left of those below shows an example of the interaction component that allows for this form of question answering.



Word: abattoir

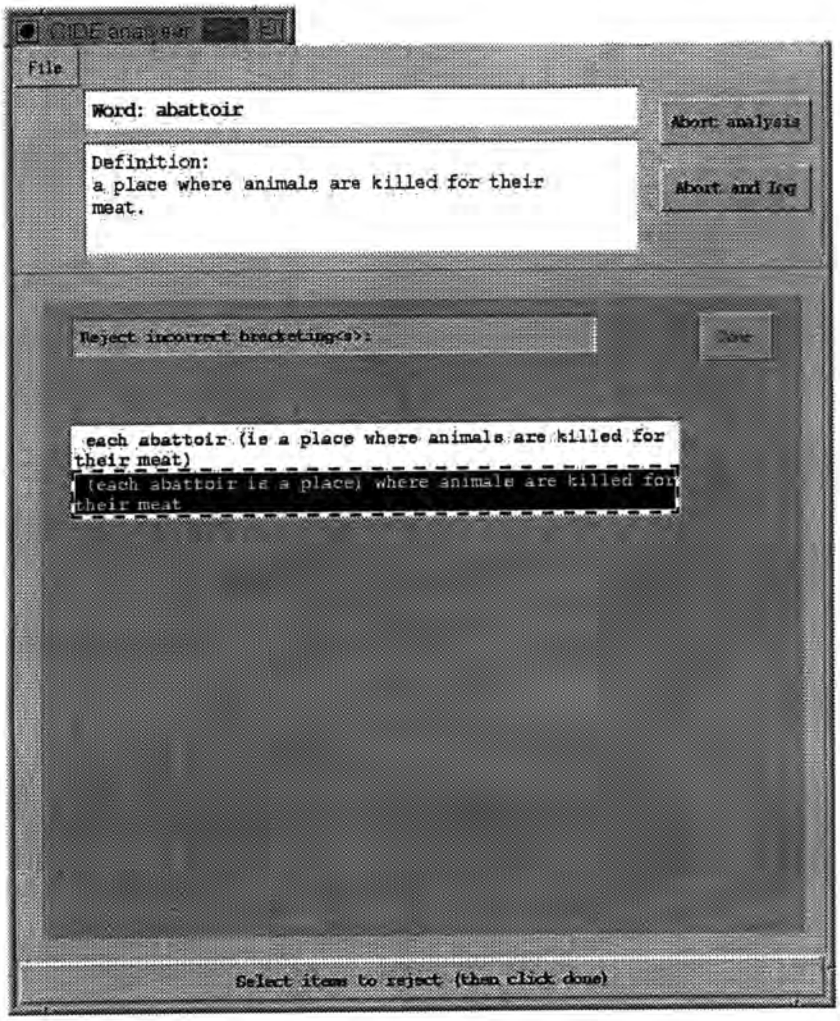
Definition:
a place where animals are killed for their meat.

Reject incorrect bracketing(s):

each abattoir (is a place where animals are killed for their meat)

(each abattoir is a place) where animals are killed for their meat

Select items to reject (then click done)



Word: abattoir

Definition:
a place where animals are killed for their meat.

Reject incorrect bracketing<s>:

each abattoir (is a place where animals are killed for their meat)

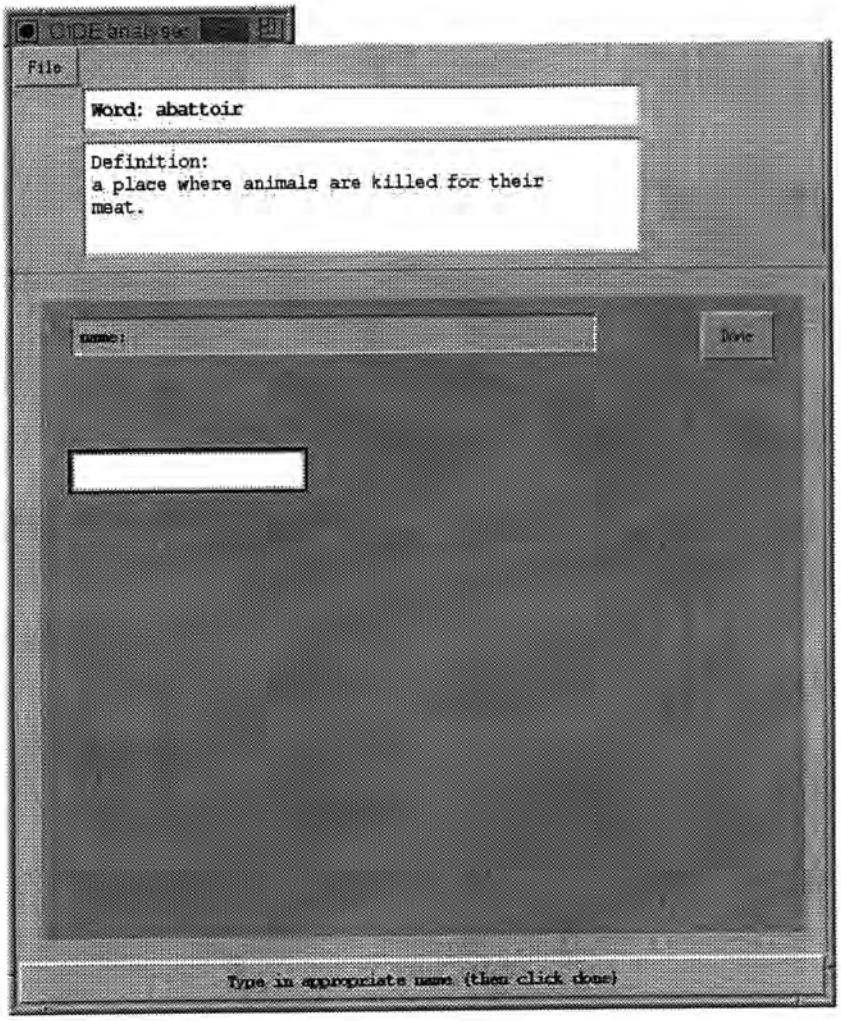
(each abattoir is a place) where animals are killed for their meat

Select items to reject (then click done)

The main difference between this type of interaction component and that for single answer selection is that there is an additional button. The "Done" button should be pressed (left click) when the user is happy with their selection of items from the list. In this case however the user is able to select as many (or as few) items as they wish. An item is selected/deselected by a click with the left mouse button. The screen shot to the right of the two above shows a single item having been selected. To send this value to the analyser the user should left click on "Done".

Entering Text to make an Entity or a Relationship Explicit

In some cases during the interaction process the user is able to use their world knowledge to give an explicit name to an entity or relationship. To be able to do this they must be allowed to type text into the GUI. The interface component below allows for the entry of text in such cases.



Word: abattoir

Definition:
a place where animals are killed for their meat.

Type in appropriate name (then click done)

To enter text the user should left click on the text box, type the text and then left click on "Done".

Accepting/Rejecting Analysis

At the end of the analysis process the user is presented with a final question as to whether to accept or reject the analysis of the system. The interaction component is simply a single selection list with yes and no being the possible answers. It is however making a special note of this case as the answer to this question is saved in the summary log.

Menu Bar

The menu bar contains two items:

Analysis Summary	This has the same functionality as the Analysis Summary button available at the word/definition selection stage.
Exit	Exits the CIDE Analyser.

Abandoning an Analysis

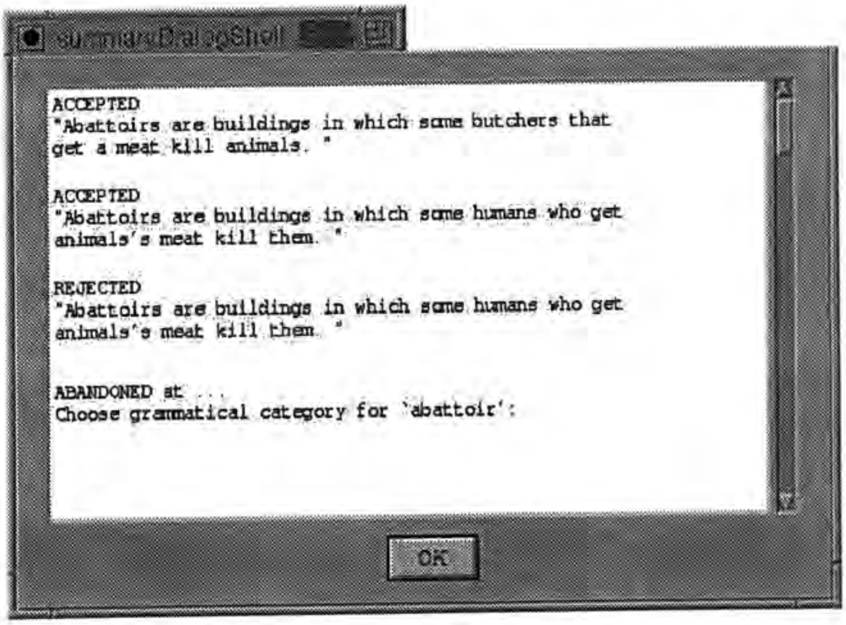
Throughout the analysis of a definition the user is at most points allowed to abandon the analysis. There are two buttons that allow this: "Abort Analysis" and "Abort and Log". Both of these return the user to the word selection screen.

The "Abort Analysis" button allows the user to experiment with the system, but does not log occasions that the user decides to abandon.

The "Abort and Log" button logs the question that was being asked when the user decided to abandon the analysis. This is useful when the user has been asked a non-sensical question by the system and wishes to record this fact.

Analysis Summary

An example of the window which appears when clicking "Analysis Summary" is:



```
ACCEPTED
"Abattoirs are buildings in which some butchers that
get a meat kill animals. "

ACCEPTED
"Abattoirs are buildings in which some humans who get
animals's meat kill them. "

REJECTED
"Abattoirs are buildings in which some humans who get
animals's meat kill them. "


ABANDONED at ...
Choose grammatical category for `abattoir':
```

This contains a brief log of where past interactions for this word/definition ended. There are four possible categories:

ACCEPTED
REJECTED
ABANDONED
FATAL ERROR

ACCEPTED and **REJECTED**

The final question of an interaction involves the user determining whether or not they are satisfied with the system's understanding of the definition's analysis. The response to this question is always logged.

ABANDONED

On some occasions during the analysis of a definition the user will be asked a question that may make no sense. The reason for this is often due to the natural language generator (that is used to ask the questions) being unable to generate an appropriate piece of English. In such situations the user will wish to abandon the analysis in such a way that the question is recorded. The "Abort and Log" button on the interface allows the user to accomplish this.

FATAL ERROR

There are situations in which the underlying natural language processor is unable to continue the analysis any further. On such occasions the system will automatically log the FATAL ERROR message and take the system back to the word selection screen.