Strategies
in NLP knowledge engineering

Responsible for the Report:
Annelise Bech, Marc Moens, Costanza Navarretta

Februar 9th, 1993

ET-12 Project
Methodologies for Constructing Knowledge Bases
for Natural Language Processing Systems
Report 1

Contents

Preface

This report is the first deliverable for the ET-12 project *Methodologies for Constructing Knowledge Bases for Natural Language Processing Systems.*

The project involves reasearchers from both the European Communities (CST - Denmark, HCRC LTG - Great Britain) and the United States (SRI International, California).

The report has been written by Marc Moens (LTG), Annelise Bech and Costanza Navarretta (CST). The authors wish to thank Bente Maegaard (CST) and Jerry Hobbs (SRI) for discussions and suggestions.

Having access to rich knowledge bases containing linguistic (syntactico-semantic) as well as extra-linguistic (world) knowledge is an essential pre-requisite for high-performance Natural Language Understanding Systems. The world knowledge comprises both general commonsense knowledge, i.e. knowledge that the reader of a text is presupposed to have, and domain specific knowledge, i.e. knowledge that is particular to the actual domain for an application.

The task of the knowledge engineering involved in building such knowledge bases for Natural Language Processing systems is the primary objective of this project. The goal of the project is to establish methodologies for constructing knowledge bases for Natural Language Processing systems by investigating which information is necessary and how it is selected.

Constructing knowledge bases from scratch is a very expensive and time consuming activity. Therefore there is a growing awareness both in the knowledge engineering community as well as in the Natural Language Processing community that for system design to be economically viable one should design components that can be reused in other systems or (re)use existing resources in one's own system. This project will contribute to the first of these types of reusability by designing a methodology that is multipurpose by which we mean generally applicable to different Natural Language Understanding systems. At the same time it will contribute to the second type of reusability by investigating the possibilities of (re)using existing resources such as text corpora, dictionaries and encyclopedia.

1 Introduction

Knowledge engineering can be defined as the activity of uncovering a body of knowledge and problem-solving strategies that domain experts use and converting it to a specific set of facts and rules stored in a knowledge base. Having access to rich knowledge bases containing linguistic as well as extra-linguistic knowledge is an essential prerequisite for high-performance natural language understanding systems. The knowledge engineering involved in building such knowledge bases for natural language processing systems is the topic of inquiry of this project.

It is often assumed that knowledge engineering for natural language processing systems differs from knowledge engineering for expert systems. The difference is argued to stem from the fact that expert systems are commonly perceived of as having more clearly defined tasks to perform than do natural language systems. The knowledge encoded in expert systems models task-dependent expertise in a particular domain; the knowledge encoded in knowledge bases for natural language systems also models task-independent commonsense theories that people use when talking about this domain (Hobbs *et al* 1987: 241). The former is often referred to as "expertise modeling"; the latter as "ontological engineering" (Alexander *et al* 1986). Because of the differences, it is commonly assumed that software engineering methods appropriate for the one are not easily transferable to the other but there are also some similarities.

In this project we are concerned with the development of methodologies and techniques that will facilitate the construction of knowledge bases that are designed specifically to be used in the processing of natural language texts. By extension, we will also examine methodologies for constructing knowledge bases from existing linguistic resources (e.g. corpora, dictionaries).

The aims of this report are to produce a state-of-the-art survey of the acquisition strategies for natural language processing and of the methods for collecting and/or (re)using existing material to support the acquisition and elicitation activities and to compare and evaluate them.

In the rest of this document, we will first give a general overview of some knowledge acquisition and knowledge elicitation strategies (section 2). This will help us to clearly situate the goals of this project and of knowledge engineering for Natural Language Processing with respect to general knowledge engineering practice.

We will then give an overview of recent acquisition and elicitation strategies for Natural Language Processing systems and evaluate them (section 3.1); in the conclusion to that section we will give a parameterized overview of the various strategies and methodologies followed (section 3.2).

We then move on to define in more detail the approach we will pursue in the project (section 4); we will discuss this both from a general knowledge engineering point of view as well as from a more strict computational linguistics and natural language processing system design point of view.

In appendix A we will survey and evaluate some methods for collecting and (re)using pre-existing material to support the knowledge acquisition and elicitation.

In appendix B we will shortly describe some of the most common techniques for knowledge elicitation in the field of knowledge engineering for expert systems.

2 Knowledge acquisition and knowledge elicitation

Since the area of knowledge engineering, knowledge acquisition and knowledge elicitation appears to be one with an ever burgeoning terminology, it is useful to clearly define the way in which we will be using some of these terms in the project.

Knowledge acquisition is usually divided in the following stages:

definition: deciding what knowledge is needed for a particular knowledge based system;

elicitation:

- **extracting:** getting the knowledge from what have been defined as relevant sources (e.g. experts); both this phase and the preceding one often involve the knowledge engineer scanning some background literature before starting interviews or other interactive acquisition protocols with experts;
- **analysis:** analysing and interpreting the extracted knowledge;
- **pre-encoding:** transforming the data into an operational representation.

encoding: writing down the knowledge in a knowledge representation format.

There may be loops in this scheme, representing the standard refine–and–debug cycle from software engineering, although they should only occur within the elicitation phase.

Handbooks about knowledge engineering for expert systems contain many techniques for eliciting knowledge from experts. Some of these techniques are interviews, focused discussion, construct elicitation, protocols, (a short description of these techniques can be found in appendix B).

In this project we will be concerned with all steps of the above knowledge acquisition process, to the exclusion of knowledge encoding proper. This means that we will be developing methodologies and possibly associated tools which are as much as possible system–independent.

There is a whole range of techniques for performing the elicitation task. Good overviews can be found in Boose (1989) and Gaines & Boose *eds*

(1988). It is generally acknowledged that most of these techniques are very labour–intensive, to the extent that knowledge elicitation is seen by many as a bottle neck in the development of expert systems and other knowledge based computational devices (e.g. Schweickert *et al* 1987). At the same time, it should be stressed that many tools have already been designed and implemented to make this process less labour–intensive. An important goal of many of these tools is to reduce the reliance on experts in the construction of the knowledge bases. Of interest for the project are also tools that involve natural language processing. For example, Moller (1988) describes the PROPOS/EPISTOS system which transforms text into a meaning representation and then performs an epistemological analysis using certain predefined pragmatic fields. And Silvestro's (1988) KBAM system analyses natural language explanations to assist in the construction of a domain–specific knowledge base.

One could see it as an ultimate goal of intelligent computer systems that they should be fully text–based— i.e. "that they acquire their knowledge by assimilating massive amounts of ordinary natural language text, rather than having to be spoon-fed rules handcrafted by knowledge engineers" (McDonald 1992: 83). Our project contributes to this goal in a tangential way. The primary goal of the project is to design methodologies and to some extent associated tools for developing knowledge bases for natural language processing applications. We will argue that, contrary to the claim reported on page 4, the knowledge engineering tasks involved in this are not all that different from the knowledge engineering that goes into constructing knowledge bases for other intelligent applications. However, since we are interested in building knowledge bases for text processing purposes, it follows that most of that knowledge will have to be extracted from text.

It follows that the goal of the project will be to develop methodologies for *defining* the needs of an NLP knowledge base, and for *extracting*, *analysing* and *pre-encoding* knowledge as is exhibited in textual material, such as corpora and dictionaries. The resulting methodologies and associated tools will therefore be of use to any knowledge engineering efforts which aim to automate some of the knowledge acquisition work by extracting knowledge from textual sources.

3 Survey of acquisition and elicitation strategies in NLP

In the last years a number of different knowledge–based NLP systems have been developed. At the same time research groups in different countries have been investigating strategies to build large machine readable lexicons containing both linguistic information and a certain degree of extra–linguistic knowledge in order to reduce the size of the knowledge bases to be built.

In this chapter some of the efforts made in these two fields are presented. The emphasis is put on the acquisition and elicitation strategies used in these efforts. The systems described have different size and scope, and the literature on them is quite heterogenous. We will in no way try to evaluate the systems themselves, but we will just concentrate on the knowledge acquisition and elicitation approaches.

The expressions **commonsense knowledge**,**world knowledge**, **encyclopedic knowledge** and **extra–linguistic knowledge** are often used intercheangeably in the described systems. In this chapter we will use these expressions as they are used in the literature about each system.

3.1 Overview of knowledge acquisition strategies in NLP

3.1.1 TACITUS

The TACITUS system (Hobbs 1986b) is a text understanding system which was originally built to handle casualty reports, using commonsense and technical knowledge. It is being developed at SRI, California.

The TACITUS parser produces a semantic translation of the natural language input in an ontologically promiscuous predicate calculus (Hobbs 1985a)—i.e. all predications are reified. These logical forms are then manipulated by the inference engine which uses abduction to choose the best explanation of a text. Atoms are given assumability costs which allows possible explanations to be ranked in some kind of order of likelihood or confidence.

Rich core theories of various domains (their basic ontologies and structures) are pre-defined and English words are characterized in terms of predicates provided by these core theories. A strategy combining explications of the core theories and characterizations of the words are used. The two processes

are repeated to check the adequacy of the definition of the core theories and of the characterizations of the words. In defining domain specific knowledge general solutions that can be used in many different applications are sought. A number of abstract systems are specified. The particular devices are concrete instantiations of these abstract systems, while the abstract systems can be used in other domains.

Strategy
To build the knowledge base Hobbs (Hobbs 1986a) developed the following three–stage working strategy:

1. Selection of the facts that should be in the knowledge base by determining what facts are linguistically presupposed by the actual text corpus.

2. Organization of these facts into independent domains (clusters) and within each domain. The aim of this stage is to discover gaps and dependencies in the knowledge base. The resulting classification is only theoretical and helps the elicitation process.

3. Formalization of the facts as predicate calculus axioms.

The group behind TACITUS determine the grain suitable for each specific domain. Linguistic and extra–linguistic knowledge are represented in the same way. Commonsense and domain knowledge are used to resolve so–called local pragmatic phenomena, such as metonymy, reference, compound nominals, lexical and syntactic ambiguities.

In defining commonsense knowledge a bottom–up strategy is combined with a top–down approach. The fact finding step implies looking at the content words in the text corpus, thus a linguistic anchoring is ensured. The universe is not seen as typed and knowledge is encoded with axioms. Thus the resulting knowledge description is independent of particular models and theories.

Here's a small sample of text that was analyzed with the three–step strategy (Bech 1989: 119):

> A bomb exploded at a Renault show room in Bilbao. A person claiming to represent the ETA–M had warned of the blast in a call to the police.

The strategy followed is one of introspection by the computational linguist who takes on the role of the "specialist" in terrorism. For a reasonable understanding of this text, the specialist decides, the system will need the knowledge that Renault is a French company manufacturing cars, that companies manufacture items in order to sell them, that a showroom is a place where one can display sellable items, that Bilbao is a city in Spain, that ETA is a so–called terrorist organisation and that it therefore has members, plans and goals, and that the methods it uses to achieve those goals would be labeled "violent" by some. These are just some of the relevant facts, which have to be encoded and added to the knowledge base or a proper subdomain of the knowledge base.

After this selection of the facts, they are organised into different domains, and occurrences of crucial words and expressions are checked in the whole text corpus. Then the knowledge about showrooms and bomb blasts is written down in TACITUS's knowledge representation scheme.

Evaluation
In defining commonsense knowledge a bottom–up strategy is combined with a top–down approach. The fact finding step implies looking at the content words in the text corpus, thus a linguistic anchoring is ensured. The universe is not seen as typed and knowledge is encoded with axioms. Thus the resulting knowledge description is independent of particular models and theories. The main problem with this strategy is that it is very introspective and very time consuming.

3.1.2 KBMT

KBMT–89 (Brown et al. 1989, Goodman and Nirenburg 1991) is the result of a two–year research project in knowledge–based machine translation at the Center for Machine Translation of Carnegie Mellon University in collaboration with IBM's Tokyo Research Laboratory.

The objective of the project was to develop a large prototype machine translation system from Japanese to English and vice versa, using an interlingua model. The translation domain is personal computer installation and maintenance manuals. KBMT–89 takes as input a sentence (in English or in Japanese) and produces a representation of its meaning in interlingua, ILT, which contains a text frame and a set of clause frames. When analysis produces ambiguous interlingua representations, these are resolved by the

automatic or interactive augmentor. From the ILT the generator produces the sentence in Japanese or in English. The analysis of source language text and the generation of target language text are based on knowledge bases including grammars, lexicons and mapping rules. The knowledge bases and augmentor use the domain model (also called ontology or concept lexicon). The knowledge representation system is FRAMEKIT (Goodman and Nirenburg 1991) which combines frames (properties and roles), with first–order predicate logic.

The ontology forms a densely interconnected network of the various types of concepts stored in frames. The system distinguishes between types and tokens. Types are the meaning of concepts while tokens are the meaning of propositions (actual events in the world). They correspond to semantic and episodic memory.

The concept lexicon is the domain ontology plus the lexical mapping rules that fill the lexical slots in frames encoding concepts. The extra–linguistic concepts are used to map the linguistic concepts into the interlingua representation to avoid mixing extra–linguistic concepts with ambiguous natural language words (yet they do not solve the problem: they must still map NL words to extra–linguistic concepts).

Strategy
At the basis of the ontology is an ontological model which defines a large set of generally applicable semantic categories for world description. The ontology is built up with a top–down strategy one domain at a time. The researchers behind KBMT–89 believe that some ontological distinctions can be found with a bottom–up approach, yet they think that top–down analysis should guide the bottom–up empirical research. In building the ontology an acquisition tool, ONTOS (Goodman and Nirenburg 1991), has been developed. It contains ontological postulates which specify a hierarchical network that helps knowledge engineers determine where domain concepts fit in the ontology. The first four postulates say that: 1) Each frame represents an ONTOS concept. 2) Concepts are subdivided into types of things that can be referred to, such as objects, events and their properties. 3) Properties of concepts are divided into relations and attributes (each slot corresponds to a property). 4) Relations map concepts into concepts, attributes map concepts into value sets.

The designers of the KBMT–89 system believe that the way in which they encode ontological distinctions provides a syntactic criterion which facilitates

consistency and type checking during the construction and extension of the knowledge base (Goodman and Nirenburg 1991). They also think that it provides a way for restricting the conceptual granularity of the ontology. The original ontological model has been refined and corrected during the process of building the personal computer concept lexicon.

The main interest in KBMT–89 is designing and building the knowledge representation language together with knowledge acquisition tools (Brown R. et al. 1989).

Evaluation
We think that the top–down approach adopted in KBMT is problematic because it does not ensure that all the facts necessary to process texts about a domain are covered. Moreover it is difficult to extend a base the structure of which is previously defined. Already in KBMT–89 the ontological model had to be modified when knowledge about texts from the specific domain on personal computers had to be encoded.

In the technical reports about KBMT–89 it is not explained which criteria guide the determination of conceptual categories, in case the choice was not straightforward, i.e. when a concept could belong to more than one category.

3.1.3 LILOG

The LILOG ("LInguistic and LOGical Methods for Text Understanding") project was organized by the IBM Scientific Center in Stuttgart in collaboration with research groups with different backgrounds from five German universities (Herzog and Rollinger 1991, Geurts 1990). The goal of the project was to develop methods for machine understanding of natural–language texts. Two prototypes were developed. The second one, LEU/2 (Linguistic Experimentation Enviroment), provides an enviroment for implementing these methods and includes a question/answer component for testing the text processing. The texts to be processed are non–technical. The actual implementation handles texts about a sightseeing tour in the center of Düsseldorf.

The parser gets lexical information from the Lexicon Manager, which delivers information about syntax, semantics and morphology of the items found in the input text. During semantic analysis LILOG differentiates between compositional semantics and analysis processes that can not be carried out

in compositional semantics.

L_{LILOG} is the representation language developed for representing the necessary domain knowledge and the knowledge extracted from the texts. L_{LILOG} is a formalism that combines sorts (features and roles) which are inherited down to subsorts, and first–order predicate calculus (typed logic). The inference engine constitutes the interpreter for L_{LILOG}.

Sorts are organized in a hierarchical structure. Functions and predicates define the functions and relations between them. The axioms state the logical properties of functions and predicates, i.e. they express which objects of which sort are related by the relations (functions and predicates) declared within the knowledge base.

The ontological model has two levels, a relatively domain– independent part, the *Upper Structure*, and a relatively domain–dependent part, the *Lower Structure*. In the Upper Structure of the ontology ENTITIES are discriminated from VIRTUAL CONCEPTS. This distinction isolates the ontological definitions supposed to be the goals of inferencing processes from abstract concepts used to define spacial environments for objects, spatio-temporal environments for events, temporal variability of single features of objects and other specifications (measures, units, names).

Strategy
The strategy used in LILOG to find what to put in the knowledge base is (Klose,G. and K. von Luke 1990):

- define a domain of discourse (in the actual implementation "sight–seeing in the center of Düsseldorf"),
- select some written information about the domain to find prototypical textual information, with a broad coverage of linguistic phenomena to be handled by the parsing and generating components,
- decide the scope and the depth of the model (granularity) considering the actual domain and system task.

Independent clusters of knowledge (space, time, objects, quality, quantity, measurability) are determined and defined. Some of the researchers in LILOG desire a more clear–cut discrimination between linguistic and extra–linguistic knowledge. In the existing implementation there is no well–defined distinction between linguistic and extra–linguistic sorts. The "linguistic"

group believes that the discrimination between sorts in the Upper Structure and sorts in the Lower Structure and the use of reification are artifacts that could be avoided if there were a clear distinction between the two different kinds of knowledge.

Evaluation
The strategy of eliciting knowledge about the specific domain is not precisely described, thus it is hard to evaluate it.

3.1.4 KT

KT (Kind Types) (Dahlgren and McDowell 1986, Dahlgren et al. 1989, Dahlgren 1992) is a system that uses commonsense knowledge to reason about natural language text. It is developed at the IBM Los Angeles Scientific Center. KT can take as input geography texts and newspaper articles and answers questions about them. The same team is at present developing the system, so that it can be used as a text selector (NewSelector).

The system is based on what the authors call naive semantics by which they mean a level of knowledge which is common to many speakers of a natural language. NS identifies words with concepts. The system distinguishes between nominal concepts that are categorizations of objects and verbal concepts that are naive theories of the implications of conceptualized events and states. Non–monotonic reasoning is used.

The system has a commonsense–knowledge base with two components, a commonsense ontology and databases of generic knowledge associated with lexical items. In the architecture of the system syntax, compositional semantics and naive semantics are separate components. The ontology has a structure similar to that of KL–ONE with features at the nodes, but descriptions in KT are probabilistic. The pattern of features relevant to each node is called a *Kind Type*. Kind types constrain the commonsense knowledge. The features of a node can be inherited by the nodes that are lower in the hierarchy.

The system distinguishes between complete and incomplete knowledge. If the current textual knowledge base conflicts with a generic inference, the knowledge contained in the textual database is chosen.

The text is read by the system and parsed. The parse is submitted to the module DISAMBIG that outputs a logical structure. Commonsense knowl-

edge is here used to determine the scope of quantifiers, the attachment of post–verbal adjuncts, and to select word senses. The logical structure is passed to a semantic translator whose output is DRS (discourse representation structures). As each sentence of a text is processed, it is added to the DRS built for the previous sentence. In this phase commonsense knowledge is used to determine definite noun phrase anaphora, sentence–external pronoun anaphora, temporal relations between the tense predicates, discourse relations and the rhetorical structure of the discourse. DRS are converted to first–order logic. In the text selection system under development a so called RELEVANCE module will determine the relevance of a text to a particular user. Naive theories include beliefs concerning the structure of the actual world and the significant relations about them.

Strategy
The basic hypothesis in KT is that people have the environment classified and that the classification scheme of a culture is reflected in its language. The ontology is a cognitive model and therefore in KT it was built empirically.

The ontological schema was constructed with the following strategy:

- the behaviour of hundreds of verbs in the geography text and newspaper corpus were studied and selectional restrictions, i.e. constraints reflecting the naive ontology embodied in English, were determined.
- The selectional restrictions were organized in a hierarchical schema from which inheritance of features could be computed.
- The hierarchy was subsequently modified on the basis of psychological studies of classification and philosophical studies of epistemology.

The process of adjusting the hierarchical schema was guided by the following constraints: a) the ontology was to be as compact as possible (pruning), b) nonexistent leaf nodes should be minimized, c) every node in the ontology was to dominate some subhierarchy, otherwise it was represented as a feature.

Verbs indicate relations between nominal classes. Some of the relational distinctions were based on the Vendler classification scheme for verb phrases (Vendler 1971). Other distinctions were based on psycholinguistic data.

Also, for defining the generic descriptions of nouns and verbs in the generic knowledge base psycholinguistic data were used. These data were also used

to define the grain of knowledge necessary to the specific implementation.

Evaluation
The strategy adopted in KT is bottom–up since the verbs and nouns in the text corpus are the basis for the construction of the ontological schema. The granularity problem and the problem of determining the categories in the ontology are claimed resolved using results from empirical research and studies. In the course of the KT project approximately one hundred persons were interviewed. It is not certain that this strategy can be used in connection with domains that are more technical than those analyzed in KT. The interesting thing about the KT approach is that it combines knowledge elicitation from texts and from experts. In the actual case the experts were "common" people.

3.1.5 Cyc

The Cyc project (Lenat et al. 1985, 1990, Lenat and Guha 1988) began in 1984 at MCC in Austin, Texas. The goal of the project is to build a very large commonsense knowledge base which can be used by many applications (both natural language processing systems and expert systems). The knowledge base should enable expert systems to handle unknown situations. A one–volume desk encyclopedia was chosen as knowledge acquisition source.

In 1984 the knowledge to be put in the Cyc knowledge base was collected manually. In the last phases of the project it should be primarily entered in an automatic way via natural language understanding (which itself requires a knowledge base for semantic disambiguation, for anaphora resolution etc.).

The representation language, CycL, is a frame–based language embedded in a predicate calculus framework along with features for representing defaults and for reification. In Cyc it is distinguished between an epistemological level (EL) and a heuristic level (HL) of the knowledge base (Lenat et al. 1990). The EL level uses a language that is first–order predicate calculus. The HL uses special purpose representations and procedures for speedy inference. A tool, TELL–ASK, for converting back and forth between the two levels has been created.

The assertion in the knowledge base are both monotonic and non- monotonic (default reasoning is allowed). The Cyc ontology is organized around the concept of categories, also called classes or collections. The collections are

organized in a generalization/specialization hierarchy. Predicates are all stongly typed and are themselves first–classes objects. Certain properties are intrinsic, others are exstrinsic.

In recent years an natural language processing system, called KBNL has been under development (Barnett et al. 1990). KBNL uses the general Cyc knowledge base. Linguistic knowledge and world knowledge are separated because the developers of KBNL believe that this separation results in a system which is more robust linguistically and more powerful as a problem solver. This separation does not mean that the knowledge base does not know about language, but that the linguistic knowledge and the domain knowledge are represented in different parts of the KB (e.g. the class *Turtle* and the word "turtle" must be represented independently). KBNL is intended to be a complete language processing system for typewritten English (the developers of the system believe that it can be extended to other languages with some modifications).

The processing components of KBNL are: Lucy, a knowledge–based English understanding system, Koko a knowledge- based English generation system, Luke a lexical acquisition tool that assists in building a lexicon. Lucy uses the knowledge base to do semantic interpretations. The overall approach to semantic processing is based on the theory of semantics presented by Montague. The use of semantic rules (for resolving metonymy, metaphors etc.) increases the number of interpretations for many expressions. To avoid combinatorial explosion Lucy assigns a level of effort to each rule.

Applications of the KBNL that are under development are navigation and query in Cyc (*Show*), text retrieval (*Scan*), and machine translation (prototype Spanish/English translation).

Strategy
The strategy to build the (ontology of) the knowledge base consists in alternating bottom–up growth with top–down design. The task is to identify, formalize and enter "microtheories" of various topics. The articles in the one–volume encyclopedia are divided into 400 distinct types (topics). One or two articles of each "type" are represented and then the last 99% of information are added using copy&edit (i.e. reuse pre–existing definitions, changing them where necessary). The strategy used is:

1. Take an article (a typical member of one of the four hundred classes of articles).

2. Represent in the representation language the knowledge which is explicitly stated in the actual article (disambiguating what the writer actually meant).

3. For each "fact" F move it up to the most general unit (frame) to which it is valid, i.e. move information to more general units. (Create a new unit if it is not already in the system).

4. For each fact write down the additional commonsense facts about the world that are needed to understand the actual fact, i.e. facts that the writer of the article presumed the reader already knew. Repeat steps 2 and 3 on this new set of facts.

5. For each adjacent pairs of sentences in the article extract and encode the intersentential knowledge.

6. Incrementally add to the representation language. At this point the representation language should "settle down" (it is refined during the previous steps). The KB should contain the most general concepts. To test whether a topic has been adequately covered, stories dealing with the topic are represented in the system. Then Cyc has to answer relevant questions about these stories.

7. Employ 2–4 dozens knowledge enterers to encode the final 99% of the knowledge base: take an article, find already represented similar articles (use copy&edit).

8. Continually test out the system by building within it various particular AI programs.

When all the knowledge extracted from the encyclopedia is encoded, knowledge from other kinds of texts (children's stories etc.) will be extracted.

Evaluation
The encyclopedia articles have been chosen in a random way. The definition of what different "types" of articles mean is not clear: "we are using an initial set of 400 mutually–distinct articles not primarily for the specific facts they contain, but rather for their "spanning the space" of knowledge" (Lenat et al. 1985). The granularity problem is not adressed because all the commonsense knowledge should be encoded. We think that it is problematic that there is no criteria for deciding when the right grain of knowledge is reached (steps

2, 3, and 4 in the method could be repeated thousands of times). The knowledge acquisition strategy in the first steps is very introspective. The Cyc base is intended to be "reusable", but we believe that it can only be used if one agrees with the Cyc ontology.

3.1.6 ACQUILEX

The ACQUILEX project (The Acquisition of lexical knowledge for Natural Language Processing Systems) was an ESPRIT project (ESPRIT–BRA 3030) that involved the Universities of Cambridge, Amsterdam, Barcelona, Pisa and Dublin, and Cambridge University Press (ACQUILEX 1992; Calzolari 1991).

The aim of the project was to represent syntactic and semantic information from machine readable dictionaries (MRDs) on a large scale and to build a Lexicon Knowledge Base (LKB). ACQUILEX had to develop techniques and methodologies for utilising and interpreting existing MRDs to extract lexical information. The main sources of this information were natural language definitions. The possibility of reusing existing lexicons for NLP systems was exploited. Several dictionaries (three monolingual English, two Italian, one Spanish and one Dutch, two bilingual Italian–English, and two bilingual Dutch- English) were used.

The representation language is a typed graph–based unification formalism with minimal default inheritance (typed feature structure language). The type system contains a type hierarchy and a constraint system. In ACQUILEX Pustejovsky's Qualia Structures (Pustejovsky 1989) are extended so that a word can be connected with more information than the four roles constitutive, formal, telic and agentive. Also the idea that lexical rules can be used to resolve some metonymic and metaphoric sense extensions is inherited from Pustejovsky. The generative rules to resolve metonymy and metaphor are, however, sometimes problematic, because they apply also to entities that are not covered by the metaphor.

The LKB is hierarchically organized and permits information to be inherited from more general words to more specific ones. To allow default inheritance they introduce the concept of **psort**, which is a feature structure from which another feature structure inherits information by default. In Pisa and Amsterdam the syntactic and the semantic interpretation are two separate processes. This is not the case for the system developed in Barcelona.

Strategy
Calzolari (1991) defines their method of extracting semantic information from the dictionaries as "heuristic and mainly inductive". It is based on the knowledge enterers' empirical observations and on some theoretical hypotheses. Their hypothesis, taken from Pustejovsky, is the existence of "meaning types" and that one can use templates (here feature–structures) for structuring semantic information.

Taxonomies are constructed from the lexical definitions in the different dictionaries. The construction of taxonomies is sometimes problematic because genus terms are not precisely defined. Feature structures are extracted from the "differentia", the properties discriminating the "definiendum" with respect to other members of the same class.

The construction of taxonomies in Barcelona and Cambridge were carried out using a top–down procedure, starting from an initial concept which acts as the head of the hierarchy. The criterion for choosing heads was that they had to appear frequently as a genus. In Pisa and in Amsterdam the taxonomies were built with a bottom–up procedure. The reaserchers behind AQUILEX believe though that the two strategies (top–down and bottom–up) should be combined (ACQUILEX 1992).

Also the extraction of feature structures from the "differentia" part of the definitions was performed with different strategies at the different sites. At all the sites the differentia are taken from a specific domain (food).

Evaluation
There are some problems with the ACQUILEX approach and these have been recognized in (ACQUILEX 1992). The fact that parts of the vocabulary have been isolated (e.g. food taxonomy) is problematic when other taxonomies must be encoded because polysemy and homonymy become relevant. The type system is too rigid, i.e. divisions between types are not always valid. The maintenance of consistency of the type structure, when many taxonomies are added, has not been adressed. The attribution of a word or a taxonomy to a type is not always straightforward and different dictionaries (in the same language or in different languages) define the same words in very different ways. Because they do not work with specific texts, the context can not help them in choosing the most suitable definition.

3.1.7 EDR

Japan Electronic Dictionary Research Institute in Tokyo is developing electronic dictionaries which they claim to be universal, i.e. not based on any specific linguistic theories or algorithms. The EDR electronic dictionaries (EDR 1990a) consist of a word dictionary, a concept dictionary, a co–occurrence dictionary and a bilingual dictionary.

The word dictionary comprises a general vocabulary dictionary and technical terminology dictionary based on lexical differences (Japanese and English versions). It includes grammatical information and a list of concepts represented by words.

The concept dictionary is divided into concept classifications and concept descriptions by type of information.

The co–occurence dictionary comes in a Japanese and an English version and is used to sentence generation.

The bilingual dictionaries comprise a Japanese–English and an English–Japanese dictionary (correspondence between Japanese headwords and English headwords).

First 170,000 vocabulary items are selected and the dictionary contents of each word are described. The word dictionary is created using the compiled dictionary data and a large set of sentences are analyzed to verify the contents written by humans. A large set of sentence examples is then collected (the EDR corpus containing 20 million sentences from newspapers, encyclopedias, textbooks and reference books). Data for the co–occurence dictionary and concept dictionary are extracted from the parsing trees and the concept relation representations in the EDR corpus.

The concept dictionary (EDR 1990c) has a network structure. It is expanded vertically and horizontally. The word dictionary and concept dictionary are linked together by headconcepts. Concept descriptions provide relations between concepts as seen in sentences. Concept classifications provide a hierarchy of concepts created to constrain the amount of knowledge described.

The concept dictionary is designed as an open–ended system and will be continually updated.

Strategy
Concept description data are described as “both top–down and bottom–up

descriptions which are refined by collating both ways..." (EDR 1990c). The bottom–up concept descriptions are made by decribing relationships between concepts in a large volume of text data and analyzing them grammatically with the word dictionary. The top–down concept descriptions are obtained by developing relationships between sub–concepts. Concept classification is developed along with the concept description using the following procedure:

1. Determine the concept categories to find super–concepts of the concept classification.
2. Classify headconcepts described in the word dictionary by the set concept categories. Adjust and modify the categories as required during classification.
3. Describe the relations between the concept categories which are considered typical concepts.
4. Determine super–concepts based on the relation descriptions between the concept categories.
5. Verify and modify the concept classification by collating the relationships between concepts deduced from the classification and the concept descriptions based on the text data with the results of concept classification.
6. Repeat the above procedures to reduce the volume of description data and obtain concept classifications consistent with the concept description.

To define the concept categories the following guidelines have been established:

- Select similar concepts (similarity is defined by all relationships of a concept with other concepts).
- Extract elements common to the similar concepts.
- Determine concept categories using the above elements as criteria.

Concept relations represented by words are extracted from semantic relationships between the words in the text data.

Evaluation
In the description of the acquisition strategy it is not explained how super–concepts are chosen and it is not clear how the huge amount of data can be organized so that it can be efficiently used and so that the knowledge bases remain consistent under expansion. The information contained in the different dictionaries is often redundant.

3.1.8 COGNITERM

The AI research group at the University of Ottawa is building a prototype bilingual (English, French) term bank, called COGNITERM (Meyer 1991). In the construction of COGNITERM they are using a generic knowledge engineering tool, CODE, which they have developed. It should allow both terminologists and users without a terminologist background to construct a knowledge base which describes concepts in frame–like units (concept descriptors). These frames are usually arranged in inheritance hierarchies. COGNITERM is a hybrid of a term bank and a knowledge base. At present the developers of COGNITERM investigate whether it is possible to distinguish between lexical–semantic and what they call encyclopedic information in the knowledge base. They believe that it is impossible to ignore encyclopedic knowledge in terminology because many applications need it.

Strategy
Meyer (1991) describes the strategy used in COGNITERM, a strategy which she finds is generally useful to define terms for a specific project. The knowledge sources for a project (domain) must be selected. Terminologists very often use texts as knowledge sources (they can also use other sources, though). First some general knowledge about the field must be acquired by doing introductory readings of different relevant materials. On the basis of these readings the boundaries of the field and subdivisions and areas of overlap with other fields must be determined. At this point terminologists can often sketch out the general knowledge structures of the field in the form of a concept network. The most relevant concepts are found. These preliminary activities help the terminologists to delimit the range of the documentation, to select the documentary corpus (and to understand experts) and to divide the corpus into subfields. The corpus can then be carefully read, i.e. relevant terms are extracted together with their contexts. At this point the terminologists can refine the conceptual network they outlined in the preliminary phase. A systematic analysis of terms in context can then begin

(both linguistic characteristics and meaning of the terms). Quality control can be achieved by revision (by other terminologists or by domain experts) and updating.

Evaluation
The strategy described by Meyer is in many aspects similar to the methods described by knowledge engineers for expert systems and to Hobbs' three step strategy.

3.1.9 The corpus lexicography approach

Fillmore and Atkins (1992) have investigated whether the use of large electronic corpora can help the lexicographers to give more accurate and more complete account of the meanings and of the use of a word than dictionaries do. They have examined the definitions of the word *risk* in ten one–volume monolingual dictionaries. Then they have analyzed the use of *risk* in 2,200 citations in order to cover all the facets of the word and encode them in frame semantics.

Comparing the entries for *risk* in the ten dictionaries they have identified the following problems in the analysis of the word:

1. sense differentiation in the verb and noun.
2. distinction between "run a risk" and "take a risk".
3. patterns of verb complementation.

During the analysis of the citations from the text corpus, Fillmore and Atkins began distinguishing three situation types where the critical differences in the meaning of the word **risk** are the presence or absence of a decision on the part of the person centrally involved in a *risk* scenario and the presence or absence of the decision–maker's awareness of the possible consequences of his decision. With frame semantics they resolve the dictionaries' difficulties in describing the word *risk* and they reduce the polysemy of the word to the cases where the uses of it instantiate different schemas. Frame semantics also helps in discovering metonymy relationships. The resulting frame structure for the word *risk* is complex but Fillmore and Atkins believe that it contains elements common to other words.

Though their approach is not economically viable in common commercial lexicography they believe that it would be very useful as electronic multi–dimensional dictionaries are developed.

Evaluation
The strategy followed by Fillmore and Atkins is interesting but we are not sure that it is suitable to use so large a text corpus as they do. Furthermore it is not certain whether it is possible to determine a text corpus that covers all the meanings of all words, and it is clerly not feasible to follow such a time–consuming approach as theirs for encoding more than a few words.

3.2 Conclusion

The problem of building a knowledge base for natural language purposes is a familiar one for many language technology groups. Faced with the problem of building an automated processor for a particular type of text, it is necessary to build a knowledge base that is suitable for this purpose. The survey of recent developments in the design of knowledge bases for natural language processing applications shows that a number of different strategies can be used for this task. They can be paramaterized along the following dimensions:

top-down vs bottom-up. The *top-down approach* was followed, e.g., in the design of the knowledge base for the KBMT system. The strategy behind CYC was originally also intended as a top-down one. The top-down strategy involves starting off from a pre-defined, non-linguistic characterization of knowledge structures. The goal is to then relate the rest of human knowledge (or in our case: knowledge needed for a particular natural language processing application) to these top layers. The task of designing such an overall ontology may seem like an awe-inspiring one. However, the reason such an approach is feasible at all is that for most applications ontologies need only be *locally* consistent and perspicuous (Lenat & Guha 1989: 23). For example, few people know what the overall ontology is in which Peter Roget embedded his Thesaurus, yet this does not have repercussions for its usefulness. The advantage of the top-down approach is that it is easy to maintain consistency, since the pre-defined grid acts as a structuring device. The disadvantage is that it can also act as a straitjacket,

leading to unwanted side-effects such as some knowledge not being accommodatable in the framework.

The *bottom-up approach* was used to a large extent in KT. It involves building the knowledge base by going from linguistic expressions and representing their meaning in a constantly evolving model. Sources for the bottom-up approach can differ. In the KT approach, the naive semantics was arrived at using results from psycholinguistic studies on category formation and conceptual organisation, results which themselves were obtained through various well-known knowledge elicitations techniques such as interview, card sorting, etc. Other systems use a corpus of text to start the bottom-up work. Again different approaches are possible here. In Fillmore & Atkins (1992) a large corpus of dictionary definitions is used as well as as comprehensive as possible a collection of the occurences of a particular word; the dictionary definitions are of course themselves already an abstraction of the meaning of a particular word that has been examined in a large corpus by lexicographers.

Few systems use one strategy at the exclusion of the other. For example, although CYC started off with a top-down strategy, it was supplemented with a bottom-up procedure. And TACITUS, which looks bottom-up, also uses a particular higher–level definition of some deep knowledge in terms of which system developers try to write knowledge base rules and definitions. Such a *combined approach* basically involves alternating bottom-up growth with top-down design.

knowledge based systems vs lexicalist systems. Efforts in the design of knowledge based natural language processing systems seem to fall into two groups. On the one hand, there are systems, like the TACITUS system, which have a relatively underspecified lexicon, with a rich knowledge base with general and domain specific rules, and a powerful inference engine. However, more recently, a lot of research has been done into the creation of lexicons for natural language processing systems which contain a lot of the commonsense knowledge one typically finds in dictionaries (e.g. Pustejovsky 1992). It could be argued, however, that this is not really a methodological distinction, but a historical one. The work on the creation of large-scale and knowledge-rich lexicons is relatively new; when development work started on systems like TACITUS, no such lexicons were available. Now that these rich

lexicons are becoming more readily available, we will have to assess to what extent methodologies like the TACITUS methodology for constructing knowledge bases can be ameliorated by taking into account this recent development.

corpus supported approach vs expert based approach. The corpus based approach to knowledge base construction relies mostly on existing text sources relevant to a particular application domain. Very few systems have applied this method. Some take a small sample of texts and look at a few example sentences. Few have developed tools for going through large bodies of text in the construction of the knowledge base. We contrast this approach with the expert based approach, where information sources from a very different nature are used. For example, in the LILOG project travel guides were scanned for relevant information. In a sense, the computational linguists acted as the "experts", in the sense that they were the kind of people who might be using the kind of tourist information service the LILOG project was prototyping. And similarly in some of the TACITUS applications (e.g. the one on terrorism) the computational linguists acted as the experts in the field.

There is a sort of intermediate approach, which involves the use of dictionaries and term banks. These are constructed on the basis of occurrences of words and terms in corpora. Using dictionaries when filling up knowledge bases could thus be construed as using knowledge derived from a corpus.

As we saw in section 2, knowledge acquisition is a labour-intensive task, which could be made more manageable if the reliance on human experts could somehow be reduced. The corpus-supported and intermediate approaches offer clues as to how that could be achieved.

4 The project approach

We think that the strategy that is best defined and that is most promising is the three-step strategy due to Hobbs. It combines bottom–up analysis with limited top–down guidance. This ensures that the knowledge encoded is still "linguistically anchored" (i.e. relevant to text processing) while at the same time maintaining some order and organisation in the process. We also prefer this strategy because its aim is not of giving guidelines for defining words, because one cannot always expect to find necessary and sufficent conditions for a word. Instead it tries to guide how to characterize words which means to find a great number of the necessary and sufficient conditions for them but not necessarily all [Hobbs 1986a].

In recent collaborative work with Jerry Hobbs carried out as part of the first phase of this project, we have extended the methodology as follows:

1. Look for all occurences of a word in the text corpus.
2. Reduce the found citations to their predicate argument relations.
3. Divide these predicate-argument relations into heaps, according to a first intuition about which predicates should go together.
4. Give an abstract characterization of the facts about the word that justify each of the heaps.
5. Find the core theory where these abstract characterizations are defined or where equivalent words are defined.
6. Generalize the concept as much as possible.
7. Determine whether the concept is appropriate at that level by examining other instance core theories.
8. Look at other predicates in the chosen core theory in order to define the word.
9. Write the definitions in predicate calculus.

This 9-step methodology should be followed for all words in the text.

4.1 Short-term goal: adaptation of Hobbs methodology

Although we think that the Hobbs methodology is the most promising, it also has at least two drawbacks. One is that it is clearly a labour-intensive technique. The other is that for domains other than the most simple ones, the full-time envolvement of both a computational linguist and a domain expert is needed. This may not be immediately obvious in the case of newspaper articles about terrorism, because both the linguist and the expert could be combined in one person.

But this may not always hold in different application domains. An added level of complexity may arise when the message understanding task is to be performed in a domain that is not understood by the computational linguist. For example, one of the partners in this project is involved in the development of a natural language understanding system for patient discharge summaries in the domain of percutaneous coronary angioplasties. Here's a representative quote:

> He has got disease in the left anterior descending artery branch with some impairment of left ventricular function.
>
> You remember that he has sustained an anterior myocardial infarction and modified Bruce protocol treadmill ECQ in July 1990 showed exercise tolerance of 9 minutes without angina and with 1mm ST segment depression.

It is not possible for computational linguists to develop knowledge axioms about things like "descending artery branches". Such knowledge has to be produced separately, by experts in the domain; we have to assume that some or all of this "expertise modeling" has taken place before the computational linguist starts her work. But experts who do the expertise modeling tend to produce knowledge bases describing their particular domain of expertise, rather than the knowledge needed for processing that particular type of text, i.e. the ontological engineering. For example, many things that are important in understanding these texts will be obvious to the experts when looking at these texts and will not be made explicit. It is up to the builder of the natural language front end to detect these gaps and add them to the knowledge base.

At the same time, we do know that a lot of the expert knowledge in most domains also exists in texts. If we were able to tap into these, we could

overcome the time-consuming hurdle of needing an expert involved in all stages of the knowledge elicitation process.

It is one of the main aims of the project to make the Hobbs revised method less labour-intensive.

An other aim of the project is to investigate whether it is possible to reuse pre–existing sources in order to make the Hobbs' strategy less introspective and less time consuming. The resulting methodology should be as general as possible, i.e. it should guide the knowledge acquisition process necessary to build knowledge bases for different domains and for different NL understanding systems. It should also give guidelines about which knowledge can be reused in different domains and/or NLU systems.

We also aim to start on the computational realisation of some aspects of this methodology through the design and later possibly prototype implementation of special knowledge acquisition tools for computational linguists, and to situate these tools within a general knowledge engineering environment.

4.2 Longer term: tool development

Tools for the computational linguist

In developing these tools, we can make a number of idealisations about the computational linguist's work bench. We assume that a computational linguist who wants to build a system for processing newspaper articles or patient discharge summaries will start from a large-coverage grammar, possibly with training facilities to fine-tune the grammar to the chosen domain, and that this grammar produces some kind of semantic representation, for example in the form of quasi-logical forms. This is not an unrealistic assumption, given the availability of large-coverage grammars for English, like the Alvey grammar (Grover *et al* 1992) or the Core Language Engine (Alshawi 1992), and the kinds of training facilities developed, e.g., by Briscoe (Briscoe & Carroll 1992) for the Alvey grammar. We will also assume that the system builder starts from a large coverage but, more importantly, knowledge-rich lexicon. Techniques for representing information efficiently in such large lexica is an active research topic (see, e.g., Pustejovsky 1992), as is the study of techniques for deriving such lexica from machine-readable dictionaries (cf. ACQUILEX) or from corpora (cf. the DELIS project).

The assumed presence of these knowledge sources does not change the general knowledge acquisition cycle described in section 2, it merely places it in a different computational context. Building a natural language understanding system for terrorist newspaper articles, or porting one to the domain of angioplasties, will involve a cycle in which portions of the text are automatically analysed by these language components, resulting in a semantic analysis the depth of which will depend on the depth of knowledge available in the lexicon or available in the general purpose knowledge component provided by the expert. The suitability of that semantic representation for the language understanding task at hand will have to be evaluated, possibly—although not usually—against some standard, and changes made (amongst other things) to the lexical knowledge base and to the general knowledge base. At that point, the natural language system developer may have to take corrective action, in the form of providing (further) background axioms that encode knowledge needed for the processing of the text.

Tools and standardisation in knowledge representation

One of the tools one could develop to make the Hobbs methodology less time-consuming is a tool for the writing of background axioms within the TACITUS system. Obviously, we should try to generalise such a tool to other representation formalisms. But whether this is possible is an open question. Although there is a substantial body of work in knowledge engineering which attempts to improve the reusability of knowledge components (e.g. Steels 1990), the current state of the field suggests that it is impossible to assume that a single language could be developed in which all knowledge representation schemes can be expressed (*pace* Neches *et al.* 1991), other than in the trivial sense guaranteed by the universality of a specific representation scheme like first-order logic.

Nevertheless, we can examine the possibility of providing a tool which allows the knowledge axioms to be stated in a language close to natural language and which maps these into some minimalistic language, probably nearly equivalent to first-order logic, but with specific notational conventions which act as pointers to operators and methods that are peculiar to particular knowledge representation schemes.

The main properties of this minimalistic language should be

- that it provides something of use in any knowledge representation

scheme;

- that it can be extended, either to follow global trends in knowledge representation theory, or to adapt it to a local knowledge representation scheme.

The aim of this aspect of the work would not be to develop a standard for the expression of background or other knowledge axioms. Standards are appropriate only where a consensus is emerging, and knowledge representation (like natural language semantics) is not an area with an emerging consensus (cf. Ginsberg 1991).

There is a growing awareness both in the knowledge engineering community as well as in the natural language processing community that for system design to be an economically viable activity one should move away from always building systems from scratch and instead design components that can be reused in other systems or reuse existing components in one's own system.

This project will contribute to the first of these types of reusability by designing a methodology for the construction of knowledge bases not restricted to a particular application, system or representation formalism. This should ensure that the methodology can be reused in the design, extension or tailoring of many different natural language understanding systems.

At the same time, we will not be starting this work from scratch but reuse various existing computational linguistics tools and techniques, such as taggers, morphological analysers, parsers, etc., the development of which lies outside the scope of this project.

5 References

ACQUILEX [1992] Final Evaluation of LDB/LKB System, Amsterdam, Barcelona, Cambridge, Dublin, Pisa. ACQUILEX, ESPRIT BRA 3030.

Alexander, J., M.F. Freiling, S.J. Shulman [1986] *Knowledge level engineering: ontological analysis.* AAAI-86, Philadelphia, PA, pp963–968.

Alshawi, H. (ed.) [1992] *The Core Language Engine.* Cambridge, MA: MIT Press.

Angele, J., D. Fensel, D. Landes, S. Neubert, R. Studer [1991] Knowledge Engineering in the Context of Related Fields of Research. In Herzog and Rollinger (eds.), pp490–500.

Barnett, J., K. Knight, I. Mani and E. Rich [1990] Knowledge and Natural Language Processing. *Communications of the ACM*, Vol.33, no.8, August, pp50–71.

Bech, A. [1990] "Hvem var ansvarlig for bombeangrebet?" Om opbygning og anvendelse af en vidensbase i et tekstforstaaelsessystem. In E. Engberg-Pedersen *et al* (eds.) *Anvendt Sprogvidenskab.* Köbenhavn: Museum Tusculanums Forlag, pp107–131.

Bech, A. [1989] The Design and Application of a Domain Specific Knowledgebase in the TACITUS Text Understanding System. In J. Pind and E. Rögnvaldsson (eds.) *Papers from the Seventh Scandinavian Conference of Computational Linguistics.* Reykjavik, pp114–126.

Bech, A. and B. Maegaard [1991] Videnrepraesentation i maskinoversaettelse. In Arnt Lykke Jakobsen (ed.), *Oversaettelse af fagsproglige tekster.* ARK 65, Köbenhavn, pp101–116.

Bell, J. and R.J. Hardiman [1989] The third role—the naturalistic knowledge engineer. In D. Diaper (ed.), *Knowledge Elicitation - principles, techniques and applications.* Ellis Horwood, pp49–85.

Bledsoe, W.W. [1986] A Man-Machine Procedure For Building a Medium Sized Knowledge Base by Analogy and Learning (Preliminary Report). MCC Technical Report Number AI-159-86, Austin Texas.

Boose, J. H. [1989] A survey of knowledge acquisition techniques and tools. In *Knowledge Acquisition*, Vol 1, pp3–37.

Bosch, P. [1991] The Bermuda Triangle: Natural Language Semantics

between Linguistics, Knowledge Representation and Knowledge Processing. In Herzog and Rollinger (eds.), pp243—258.

Braasch, A.D. [1991] Delprojekt 3: Oversaettelsesteori i Maskinoversaettelse Valg af Tekstsort – Korpus – Undersögelsesaspekter. In Arnt Lykke Jakobsen (ed.), *Oversaettelse af fagsproglige tekster.* ARK 65, Köbenhavn, pp117-131.

Brachman, R.J. & H.J. Levesque [1985] *Readings in Knowledge Representation.* Morgan Kaufmann.

Briscoe, T. [1991] Lexical Issues in Natural Language Processing. In A. Sanfilippo (ed.), *The (other) Cambridge* ACQUILEX *Papers.* Technical Report No.253. University of Cambridge, Computer Laboratory, pp2–23.

Briscoe, T. & J. Carrol [1992] Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars. To appear in *Computational Linguistics.*

Briscoe, T. and A. Copestake [1991] Sense extensions as Lexical Rules. University of Cambridge, Computer Laboratory, ESPRIT BRA 3030 ACQUILEX WP No.022.

Brown, R. *et al* **[1989]** KBMT-89 Project Report. Center for Machine Translation, Carnegie Mellon University.

Bylander, T. and B. Chandrasekaran [1988] Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. In Gaines and Boose (eds.), Vol.1, pp65–77.

Börkel, M., P. Gerstl [1991] A Knowledge Engineering Environment for LILOG. In Herzog and Rollinger (eds.), pp482–489.

Calzolari, N. [1991] Acquiring and Representing Semantic Information in a Lexical Knowledge Base. In Pustejovsky and Bergler (eds.).

Calzolari, N., A. Zampolli [1991] Methods and Tools for Lexical Acquisition. In M. Filgueiras, L. Damas, N. Moreira, A.P. Toms (eds.) *Natural Language Processing, EAIA'90 Proceedings.* Berlin Springer-Verlag, pp4–24.

Calzolari, N., T. Marti, P. Vossen [1991] Taxonomies and Feature Structures. **acquilex, esprit bra** 3030, Pisa, ILC-ACQ-91.

Copestake, A., T. Briscoe [1991] Lexical Operations in a Unification-based Framework. University of Cambridge Computer Laboratory, ESPRIT

Cordingley, E.S. [1989] Knowledge Elicitation Techniques for Knowledge-Based Systems. In D. Diaper (ed.), *Knowledge Elicitation - principles, techniques and applications*. Ellis Horwood, pp87–176.

Dahlgren, K. and J. McDowell [1986] Kind Types in Knowledge Representation. *Proceedings of COLING*, Bonn, Germany, pp216–221.

Dahlgren, K., J. McDowell, E.P. Stabler Jr. [1989] Knowledge Representation for Commonsense Reasoning with Text. *Computational Linguistics*, vol.15, no.3, pp149–170.

Dahlgren, K. [1991] The Autonomy of Shallow Lexical Knowledge. In Pustejovsky and Bergler (eds.), pp255–267.

Diaper, D. [1989] Designing Expert Systems - From Dan to Beersheba. In D. Diaper (ed.): *Knowledge Elicitation - principles, techniques and applications*. Ellis Horwood, pp15–46.

EDR [1990a] An Overview of the EDR Electronic Dictionaries. Technical Report-024, Japan Electronic Dictionary Research Institut Ltd.

EDR [1990b] English Word Dictionary. Technical Report-026, Japan Electronic Dictionary Research Institut Ltd.

EDR [1990c] Concept Dictionary. Technical Report-027, Japan Electronic Dictionary Research Institut Ltd.

Edwards, J.S. [1991] *Building Knowledge-Based Systems—towards a methodology*. London: Pitman.

Fillmore, J.C. and B.T.S. Atkins [1992] Starting where the dictionaries stop: the challenge of corpus lexicography. In B.T.S. Atkins & A. Zampolli (eds.): *Computational Approaches to the Lexicon*, Oxford University Press.

Gaines, B.R. and J.H. Boose (eds.) [1988] *Knowledge Acquisition for Knowledge-Based Systems*. Vol 1 and 2. Academic Press.

Gaines, B.R. [1988] An Overview of Knowledge-Acquisition and Transfer. In Gaines and Boose (eds.), Vol 1, pp3–22.

Geurts, B. (ed.) [1990] Natural-Language Understanding in LILOG—An Intermediate Overview. IWBS Report 137.

Ginsberg, M.L. [1991] Knowledge Interchange Format: the KIF of death. *AI Magazine*, Vol 12, 3, pp57–68.

Goodman K. and S. Nirenburg (eds.) [1991] *The* KBMT *Project: A Case Study in Knowledge-Based Machine Translation.* Morgan Kaufmann.

Grover, C., J. Carrol & T. Briscoe [1992] The Alvey natural language tools grammar. (4th release) Technical Report. University of Cambridge: Computer Laboratory.

Gust, H. [1991] Representing Word Meanings. In Herzog and Rollinger (eds.), pp127–142.

Hart, A. [1986] *Knowledge acquisition for expert systems.* London: Kogan Page.

Hayes, P. [1985] The second naive physics manifesto. In Hobbs and Moore (eds.), pp1–36.

Herzog, O. and C.-R. Rollinger (eds.) [1991] *Text Understanding in* LILOG. BerlIn: Springer-Verlag.

Hobbs, J.R. [1984] Sublanguage and Knowledge. Technical Note 329. SRI, California.

Hobbs, J.R. & R.C. Moore [1985] *Formal Theories of the Commonsense World.* Ablex.

Hobbs, J.R. [1985a] Ontological Promiscuity. In *Proceedings of ACL-85*, pp61-69. University of Chicago, Illinois.

Hobbs, J.R. [1985b] Granularity. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence. Vol.1*, pp432–435.

Hobbs, J.R., W. Croft, T. Davies, D. Edwards, K. Laws [1986a] Commonsense Metaphysics and Lexical Semantics. Technical Note 392. SRI, California.

Hobbs, J.R. [1986b] Overview of the TACITUS Project. *Computational Linguistics*, Vol 12, No.3.

Hobbs, J.R. [1987] World Knowledge and Word Meaning. In *Proceedings of TINLAP-3*, Las Cruces, New Mexico.

Hobbs, J.R. and P. Martin [1987] Local Pragmatics. Technical Note 429. SRI, California.

Hobbs, J.R., M. Kameyama [1990] Translation by Abduction. *Proceedings of COLING-90*, Helsinki, vol.3, pp155–161.

Hobbs, J.R., M. Stickel, D. Appelt and P. Martin [1990] Interpretation as Abduction. Technical Note 499. SRI, California.

Kelly, G.A. [1970] A brief introduction to personal construct theory. In D. Bannister (ed.): *Perspectives in personal construct theory.* Academic Press, London 1970.

Klose, G. and K. von Luck [1990] The Representation of Knowledge in LILOG. In H.Czap, W.Nedobity (eds.) *TKE'90: Terminology and Knowledge Engineering*, Vol.1, Indeks Verlag, Frankfurt/M, pp263–275.

Klose, G. and K. von Luck [1991] The Background Knowledge of the LILOG System. In Herzog and Rollinger (eds.), pp455–463.

LaFrance, M. [1988] The Knowledge Acquisition Grid: a method for training knowledge engineers. In Gaines and Boose (eds.), Vol.1, pp81–104.

Lang, E. [1991] The LILOG Ontology from a Linguistic Point of View. In Herzog and Rollinger (eds.), pp464–481.

Lenat, D.B., E.A. Feigenbaum [1987] On the Thresholds of Knowledge. MCC Technical Report Number AI-126-87, Austin Texas.

Lenat, D.B., R.V. Guha [1988] The World According to CYC. MCC Technical Report Number ACA-AI-300-88, Austin Texas.

Lenat, D.B., R.V. Guha, K. Pittman, D. Pratt and M. Shepherd [1990] Cyc: Toward Programs with Common Sense. In *Communications of the ACM*, Vol.33, no.8, pp30–49.

Lenat, D.B., M. Prakash, M. Shepherd [1985] Cyc: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. MCC Technical Report Number AI-055-85, Austin Texas.

McDonald, D. D. [1992] Robust partial parsing through incremental multi-algorithm processing. In P. S. Jacobs (ed.), *Text-Based Intelligent Systems*, pp83–100. Hillsdale: Erlbaum.

Maegaard, B. and H. Ruus [1987] The Composition and Use of a Text Corpus. *Linguistica Computazionale.* Vol.4, no.5, pp103–122.

Matsukawa T. and E. Yokota [1991] Development of the Concept Dictionary—Implementation of Lexical Knowledge. In Pustejovsky and Bergler (eds.), pp305–319.

Meijs, W. and P. Vossen [1991] In so Many Words: Knowledge as a

Lexical Phenomenon. In Pustejovsky and Bergler (eds.), pp137–153.

Meyer, I. [**1991**] Knowledge Management for Terminology-Intensive Applications: Needs and Tools. In Pustejovsky and Bergler (eds.), pp21–38.

Moller, J.U. [**1988**] Knowledge acquisition from texts. *Proceedings of the second European knowledge acquisition workshop,* Bonn, June, pp25.1–16.

MUC–3 [**1991**] *Third Message Understanding Conference.* Proceedings of a Conference held in San Diego, California, May 21–23. Kaufmann.

MUC–4 [**1992**] *Fourth Message Understanding Conference.* Proceedings of a Conference held in McLean, Virginia, June 16–18. Kaufmann.

Neches, R., *et al* [**1991**] Enabling Technology for Knowledge Sharing. In *AI magazine*, Vol.12, no.3, pp36–56.

Nirenburg, S. *et al* [**1988**] Acquisition of Very Large Knowledge Bases: Methodology, Tools and Applications. CMU-CMT-88-108, Carnegie Mellon University .

Nirenburg, S. and L. Levin [**1991**] Syntax-Driven and Ontology-Driven Lexical Semantics. In Pustejovsky and Bergler (eds.), pp5–20.

Pustejovsky, J. [**1989**] Current Issues in Computational Lexical Semantics. In *ACL Proceedings, Fourth European Conference*, Manchester, England, ppxvii–xxv.

Pustejovsky, J. and S. Bergler (eds.) [**1991**] *Lexical Semantics and Knowledge Representation. First SIGLEX Workshop, Berkeley, Ca.* Berlin Springer-Verlag

Renouf, A. [**1984**] Corpus Development at Birmingham University. In J. Aarts and W. Meijs (eds.) *CORPUS LINGUISTICS – Recent Developments in the Use of Computer Corpora in English Language Research.* Rodopi, Amsterdam, pp3–39.

Silvestro, K. [**1988**] Using explanations for knowledge base acquisition. *International Journal of Man–Machine Studies*, **29**, pp159–170.

Sowa, J.F. [**1984**] *Conceptual Structures—Information Processing in Mind and Machine.* Addison-Wesley, USA .

Steels, L. [**1990**] Components of Expertise. *AI Magazine*, 11, 2, pp28–49.

Velardi, P., M.T. Pazienza[**1989**] Computer Aided Interpretation of Lexical Cooccurrences. In: *ACL Proceedings, 27th Conference*, Vancouver,

Canada, pp185–192.

Velardi, P., M.T. Pazienza, and M. Fasolo [1991] How to Encode Semantic Knowledge: A Method for Meaning Representation and Computer-Aided Acquisition. In: *Computational Linguistics*, Vo.17, No.2, pp153– 170.

Veronis, J., N.M. Ide [1991] An Assessment of Semantic Information Automatically Extracted from Machine Readable Dictionaries. In: *ACL Proceedings, Fifth European Conference*, Berlin, Germany, pp227–232.

Vendler, Z. [1971] Singular Terms. In D.D. Steinberg, L.A. Jakobovits (eds.), *Semantics*. Cambridge University Press, Cambridge England, pp115–133.

Vossen, P. [1991] Comparing Noun-Taxonomies Cross-Linguistically. University of Amsterdam, , ESPRIT BRA 3030 ACQUILEX-WP No.014.

Woodward, J.B., M.L.G. Shaw and B.R. Gaines [1992] The Cognitive Basis of Knowledge Engineering. In F. Schmalhofer, G. Strube and Th. Wetter (eds.), *Contemporary Knowledge Engineering and Cognition*. Berlin: Springer-Verlag.

Zweigenbaum, P., M. Cavazza [1991] Extracting implicit information from free text technical reports. In: *RIAO, Conference Proceedings, Intelligent Text and Image Handling*, Barcelona, Spain, Vol.2, pp695–705.

A Reusability efforts

At present NLP systems (and likewise knowledge–based systems) are built from scratch. That is for each new NLP system (and knowledge–based system) a new knowledge base is constructed. Building knowledge bases is, though, a very expensive and time consuming process. Therefore researchers have become more and more interested in investigating the possibility of sharing and reusing existing resources. In this chapter we will give a survey of some of the efforts made to reuse and/or to share resources as they are described in literature. First the meanings of the word reusable are briefly discussed then the different approaches to the reusability issue are presented.

A.1 Meanings of the word reusable

Nicoletta Calzolari [Calzolari 1991] distinguishes between two main senses of the word reusable in the field of computational lexicography:

reusable_1 to reuse implicit and explicit lexical information in pre–existing lexical resources (machine readable dictionaries, terminological databases, textual corpora etc.) to facilitate the construction of reusable (in the sense of reusable_2) computational lexicons.

reusable_2 to construct large computational lexicons so that different NLP systems with appropriate interfaces can extract lexical information.

Thus reusable can both mean to reuse pre–existing resources and be (re)used by many systems.

A.2 ACQUILEX

The aim of the ACQUILEX project (cf. section 3.1.6) was to exploit methods of building very large dictionaries, reusing pre-existing lexical resources. In ACQUILEX information was extracted from eight monolingual and bilingual dictionaries and the goal was to build a lexical knowledge base prototype. The large lexical knowledge base should not only contain the lexical information which was implicit and explicit in the considered dictionaries but also some semantic information. In the project they (re)used (reusable_1) the definitions of the eight dictionaries to build taxonomies. This process

was not straightforward because different dictionaries contain different genus terms for the same word. The problem of choosing the "correct" reading of a word is still open: "The attribution of a word or of a taxonomy to a type seems sometimes more of an ontological or philosophical nature" [, 1992]. For the same reason strategies to automatically extract knowledge from the existing dictionaries have not yet been found.

There is at present a project sponsored by the EC (Semantic analysis, using a natural language dictionary) that is investigating the possibility of extracting semantic information using the COBUILD dictionary .

A.3 EDR

The EDR dictionaries under development at the Japanese Dictionary Research Institute (cf. section 3.1.5) are intended to be reusable (reusable_2) in different systems and applications. They include word and concept dictionaries in both Japanese and English. Their construction was based on existing dictionaries and on an extensive text corpus.

The dictionaries are still under development, but it is not clear how the consistency of the large amount of data collected is mantained during the continuous expansions. It is not clear either how efficient the navigation through these very large dictionaries is.

It is believed that in order to be reusable the dictionaries must not depend on particular linguistic theories, and it is claimed that a neutral representation of concepts is given in the EDR dictionaries. It is, however, not explained in which way the actual representation of concepts and words is "neutral".

A.4 Cyc

The approach to reusability in the Cyc system (cf. section3.1.5) is different from that of the lexical approaches described above. The aim of the Cyc project is to build a huge knowledge base containing both commonsense and some domain specific knowledge. The Cyc knowledge base is built not only to be reused by different NLP systems (reusable_2), but also by different expert system applications. To test the reusability of the Cyc knowledge base, different applications that rely on it are continuously added to the system, e.g. NLP systems.

To make the knowledge base reusable the researchers at Cyc have encoded it in declarative semantics. The heuristics necessary to enable the inference engine to work efficiently are totally separated from the declarative definition of the knowledge base.

The effort made in Cyc is very interesting because in practice it results in the construction of a big knowledge base which many applications can use. To reuse the Cyc knowledge base it is, however, necessary to accept the world model which is behind its ontology. This can be problematic if the knowledge contained in the base is not sufficient for a particular application. It would then be necessary to add the extra knowledge in a way that is consistent with the existing Cyc model.

A.5 The Shared Effort

The Knowledge–Sharing Effort is a project sponsored by DARPA, by the Air Force Office of Scientific Research, by the National Science Foundation and the Corporation for National Research Initiatives to develop the technical infrastructure to support the sharing of knowledge among systems.

The motivation for the project is the vision of being able to build knowledge–based systems by assembling reusable (reusable_2) components [Neches et al. 1991]. Then system developers would only need to worry about creating the specialized knowledge and reasoners new to the specific task of their system.

At present it is technically problematic to realize this vision because there is no consensus on the appropriate form or content of the shared ontologies. [Neches et al. 1991] think that one should build a few shared knowledge bases, extract generalizations from the set of systems that emerge, and capture these generalizations in a standard format.

The people involved in the Shared Effort believe that application systems contain many different kinds of knowledge. They claim that at the top level are ontologies that represent topic independent (time, causality....) or topic dependent knowledge. Together with more application–specific models these ontologies define how the application describes the world. They believe that at the bottom level assertions using the vocabulary of these models capture the current state of the knowledge of the system. Knowledge at the higher levels is easier to share and reuse, because it is less specialized. Knowledge

at the lower levels can only be shared if the other systems accept the models in the levels above.

The philosophy behind the Shared Effort is that knowledge- based systems should be assembled by components that include a framework for local system software in which one or more local knowledge bases are tied to a shared ontology (libraries of reusable ontologies). Remote knowledge bases can be accessed and are understood by the local system by virtue of being tied to the ontology.

The Knowledge–Sharing Effort is organized into four working groups:

1. The Interlingua Working Group, which is developing an approach to translate between knowledge representation languages.
2. The Knowledge Representation System Specification Working Group that is seeking to remove arbitrary differences among knowledge representation languages within the same paradigm.
3. The External Interfaces Working Group is developing a set of protocols and conventions for interaction that would allow a knowledge–based system to obtain knowledge from another knowledge–based system by posting a query to this system and receiving a response.
4. The Shared Reusable Knowledge Bases Working Group is working on overcoming the barriers to sharing that arise from lack of consensus across knowledge bases on vocabulary and semantic interpretations in domain models.

The Shared Effort is a new interesting initiative to remove some of the formal obstacles to the sharing of knowledge bases.

A.6 Comparison of the reusability approaches

The main problem with reusing (reusable_1) existing dictionaries instead of text corpora is that the definitions contained in dictionaries do not always cover the meaning of a word which is interesting in a particular application. The quality of dictionaries is not always high and lexicographers already have picked and chosen some meanings after criteria that are not necessarily valid for all systems and applications. It would, however, be a big help in

the building of a knowledge base to have the possibility of getting some of the meanings of a word from pre–existing resourses. If definitions from dictionaries were compared with the meanings of a word needed to a specific application it would be possible to cope with some of the difficulties met by the ACQUILEX researchers who have no context to rely on when they choose among the different definitions in different dictionaries.

The approach taken in Cyc is interesting, but it is not guaranteed that other systems would be able to use the Cyc knowledge base if the world model behind it is not compatible with their models.

The researcher involved in the Shared Effort try to enable different systems to share knowledge bases, seen as different modules. The goals of the project are centred on establishing standards (representation languages, protocols etc.) that will enable the reuse (reusable_2) and/or the exchange of data bases among different systems. The Shared Effort is based on a vision, and its results will not be seen in the nearest future.

B Overview of some of the most common elicitation techniques in knowledge engineering for expert systems

Many techniques for knowledge elicitation in the field of knowledge engineering for expert systems have been described in literature [Hart 1986] [Cordingley 1989]. Some of these techniques are inherited from other fields (generic software engineering, psychology etc.) and have been modified to suit the construction of expert systems. Some methods are formal, others are quite informal. Most of them deal with knowledge elicitation from human expert(s). In this appendix some of the most common of these techniques are shortly described.

Interviewing and focused discussion
The most natural way of eliciting knowledge from an expert is by interviewing him/her. There are different strategies for interviewing people, but the most general distinction is between *unstructured* and *structured interviews*. The first kind of interview is useful in the initial stages of the elicitation activity when the knowledge engineer does not have much knowledge about the specific domain. It consists of asking questions to the expert in an undefined order. In the ensuing phases of the elicitation process it is advisable [Cordingley 1989] to use *structured interviews*. Here the knowledge engineer must work with specific questions in the same order for each interview.

Focused discussion is similar to interviewing, though it is more informal and centered around the element in focus. The focus of a discussion can be cases the expert is working with, artefacts or concepts of the domain, lists of relevant objects, tasks etc.

Construct elicitation
The term *construct* comes from the **Personal Construct Theory** developed by George Kelly in the 50's and 60's [Kelly 1970]. Originally the theory was developed in the field of clinical psycology. Since then it has often been used in knowledge engineering. The theory is based on the postulate that a person uses a mental tool, a *construct*, to discriminate between elements of his world. Each construct is a bipolar discrimination which the person uses for understanding the world. Each element for which a construct is relevant can best be characterised by a pole of the construct. The pole that is named first is called *emergent* one, the other is referred to as the *implicit*

one. There may or may not be intermediate positions between the two extremes. People make sense of the world by anticipating events on the basis of their personal construct system. Each person's system is always under development: when some of the expectations to the world are not fulfilled, the person has to modify the system to handle the new,surprising events. The theory of construct is called *personal* because each person has his individual construct system. According to the theory people with similar backgrounds have many common constructs, though.

There are different ways of eliciting constructs. The most commonly used are the techniques of *triads* and *dyads*. In the first technique the knowledge provider is presented with three elements and he has to say which two are alike in some way and different from the third one (these two elements will identify the emergent poles). In the technique of dyads the knowledge provider is asked to consider two elements and he has to say whether they are similar or different and then he has to explain his affermations. This technique is preferred to that of the triads when the elements of the domain are too complex to be considered in groups of three.

The *repertory grid* is a two-way classification of the elements relevant to the domain against the constructs. There are different grids in use, but there is no one correct format for grids and it is possible to use them in different ways. An example of repertory grid is *LaFrances' grid* which is a matrix of five *forms of knowledge* by six *Question types* [LaFrance 1988]. The forms of knowledge are:

- **Layouts** which are the expert's descriptions of tasks.
- **Stories** which are accounts of previous experiences (case studies etc.)
- **Scripts** which give the expert's sequential and procedural knowledge of the domain.
- **Metaphors** which provide the expert's alternative images of the task.
- **Rules of thumb** provide tactics and heuristics (tacit knowledge).

The six question types are:

- **Grand tour questions** whose aim is to provide an overview of the domain.

- **Cataloguing the categories** which should provide a taxonomy of the expert's terms and concepts.
- **Ascertaining the attributes** whose aim is to discover the distinguishing features of the expert's concepts.
- **Determining the interconnections** which should provide the relations among the concepts in the domain.
- **Seeking advice** which should reveal the expert's strategies.
- **Cross checking questions** which are used to validate the information got with the other kinds of questions.

This grid provides a framework for knowledge elicitation. *Laddering* is another technique based on the theory of personal constructs. It is used to organize concepts in hierarchies. To get superordinate concepts the knowledge engineer asks the knowledge provider "why...?" of constructs. To get subordinate concepts he asks "how...?" and to get concepts at the same level in the hierarchy he asks for "alternative examples of...".

Twenty questions
Twenty questions has originally been used by ethnographers as a research technique to investigate the habits of people from different cultures. Recently the technique has been used to elicit knowledge from experts. The knowledge engineer chooses an element from a set of situations, diagnoses, problems, states etc. which are relevant to the actual domain. The knowledge provider must then guess the chosen element by asking questions that can be answered with "yes" or "no". The questions give the knowledge engineer a lot of information about the domain and about the way the knowledge provider thinks.

Role play
In *role play* the knowledge provider gets a "realistic" role and must accomplish a task where his expertise is necessary. The method is useful when the knowledge engineer has to extract knowledge about work routines and procedures.

Simulation
Simulation is a method in which the knowledge provider is put into a situation which is made to seem as real as possible. His reaction to the situation provides information to the knowledge elicitator. The method is often used

to develop user interfaces and in prototyping. A famous example is the "Wizard of Oz" method developed by Diaper. In this method the knowledge engineer simulates an intelligent, full NLP interface for expert systems, so that the expert believes that he is communicating with a real expert system. The analysis of these simulations is useful to extract both the linguistic and extra– linguistic knowledge that the final system should contain.